Leiden University

ICT in Business and the Public Sector

Bimodal Architecture: The first Reference Architecture for setting-up a Bimodal IT platform

Name: Rajesh Lale

Student no: S2090996

Date: 07-05-2021

1st supervisor: Drs. J.B. Kruiswijk

2nd supervisor: Dr. C.J. Stettina

Master Thesis

Leiden Institute of Advanced Computer Science (LIACS)

Leiden University

Niels Bohrweg 1

2333 CA Leiden

The Netherlands

Preface

This thesis, Bimodal Architecture: The first Reference Architecture for setting-up a Bimodal IT platform, is written to fulfill the graduation program and requirements for the degree Master of Science in ICT in Business at Leiden University. The thesis is developed under the supervision of Drs. Bas Kruiswijk and Dr. Christoph Stettina. The topic and insight to write this thesis are based on my own experience as a Senior Integration and Solution Architect at Accenture. It is very hard to visualize and set-up a proper architecture within this turbulent digitalization and transformation era. Also, the thesis is written during the COVID-19 pandemic, causing a lot of distractions while working from home and keeping the necessary focus on writing the thesis. Despite the demanding situations and challenges, I believe that the thesis is an appropriate conclusion of the ICT in Business masters' program and can hopefully assist within the transformation journey of organizations.

In addition, I would like to take this opportunity to express my deepest gratitude to everyone who has helped me throughout the process. Foremost, my first supervisor, Bas Kruiswijk, who has helped me from the very beginning until the end. His time, effort, and guidance helped me to shape an idea into a research topic and eventually to create a practical architecture that can be used in a real-time context. I would also like to thank Christoph Stettina, my second supervisor, for his effort, guidance, and for challenging me to think out-of-the-box. His feedback helped me to take the thesis to a higher level, as well from an academically perspective as from a content perspective.

My gratitude also goes to my nani Satwatie Chauthi, adjie Sosiel Bakas, and especially to my parents Boyke and Lien, for their endless love, support, guidance, blessings, care, and tolerance. They are my inspiration and driving force to achieve and accomplish anything in life! These appreciations also apply to my aunts Asha mausi, Roma mami, Sharda phoewa, Sherita kaki, Helda mausi, and tante Annelies. And, last but not least, my uncles Radjin mama, Johnny kaka, and oom Rob.

Finally, I would like to thank my best friends, also referred to as "my brothers and sister," Maninder Nandra, Leroy Thode, Brain Badal, Arno ter Horst, Vincent Jacobs, Krystyna Cichon, Dominic Jahangier, Malik Samnani, and Irfan Yuta. They helped me stay focused throughout the entire masters' program and supported me, especially Dominic and Arno, in finishing this thesis. Without them, it would not be possible to do this!

Rajesh
Lelystad, May 7th, 2021

Abstract

Background. In the last decade, customer behavior has changed drastically. The demand for intuitive and radical user-oriented products and new market entrants with digital business models have transformed how organizations, especially established organizations, should compete and conduct business. This change also impacts the IT organization on how they should support the business to fulfill the different needs of their stakeholders. To help deal with the existing (legacy) IT systems and to adopt modular IT systems on the other side, concepts, such as Bimodal IT, have emerged in the last several years to assist within the transformation journey.

Aim. Since the concept of Bimodal IT is mainly focused on the organizational and managerial aspects, the architectural aspects are not highlighted. While the architectural aspects are equally or even more important because, without a proper architecture, an organization will not be able to realize and facilitate agility.

Therefore, the aim of the research is to provide more insight into how a Bimodal Architecture could be arranged or realized on an enterprise architecture level or be implemented into an IT landscape. It describes the design patterns of a Bimodal IT platform and provides the core components of the platform through a Reference Architecture.

Method. In this qualitative research, a combination of a literature review and a multi-case study is used to identify and gather the core elements that are required to set-up the foundation of a Bimodal IT platform. In total, 15 semi-structured in-depth interviews were conducted to obtain the information with candidates that already or still have experience with the transformation towards Bimodal IT on an enterprise-level, such as Enterprise Architects, and on a project/solution implementation/development level, such as Integration Architects, Solution Architects, Integration and Technical Leads, and developers.

Results. The literature and the interviews disclose that the fundamental design patterns of a Bimodal IT platform are a Stable Layer, Explorative Layer, and a Communication Layer. These layers are supported by two tiers, namely, a SDLC tier to deliver the tools used to build and deliver the various IT systems and a Deployment tier to host the IT systems.

These design patterns also close the gap in the current body of knowledge regarding the architectural aspects within the concept of Bimodal IT.

Conclusion. Based on the identified core elements, a Reference Architecture has been developed. The Reference Architecture can assist in the visualization, realization, implementation, and adoption of a Bimodal IT platform within an organization.

Universiteit Leiden

Table of contents

List of figures

List of tables

1. Introduction

1.1. Background

In the last decade, changing customer behaviors, such as demand for intuitive and radically user-oriented products and new market entrants with digital business models, have transformed how organizations are competing and conducting business (Bughin et al., 2013). With the rise of digital offerings, services, and enhanced customer experience, established organizations need to focus on IT agility and IT exploration to enable digital transformation (Haffke et al., 2017a). However, the problem is that many established organizations are struggling with the implications of this digital transformation due to complex and rigid IT infrastructures and inflexible hierarchical organizational structures between business, IT, and traditional business models (Ghareeb & Berchten, 2011). Therefore, established organizations need to find a balance in which they can improve and enhance the customer experience and create new revenue streams as well as "keeping the lights on" within their IT organization(s) and department(s) (Haffke et al., 2017a; Horlach et al., 2017).

1.2. Research topic

To cope with the rapidly changing environment, customer demands and preserve the high IT investments of the various organizations, Gartner came up with the concept of "Bimodal IT," also referred to as "Two-Speed IT" within the literature (Bossert et al., 2014; Haffke et al., 2017b; Horlach et al., 2016). The concept of Bimodal IT is described as "the practice of managing two separate, but coherent, styles of IT delivery. The focus is on one side on exploitation and predictability (Mode 1), while on the other side, it tries to deal with exploration and innovation (Mode 2)" (Gartner, 2013).

In contrast, from an IT architectural point of view, Mode 1 and Mode 2 could be delineated as two separate IT platforms with their own set of characteristics (Bossert et al., 2014). Because the attributes differ for the two IT platforms, each platform could use another approach and method to implement an IT system within its domain.

Therefore, Bimodal IT implies more than only a change within the organizational culture and how established organizations deliver their IT projects. From an IT architectural point of view, the challenge is to create an enterprise application architecture that is based on a component-based distributed architecture via the use of a Microservice Architecture (MSA) or a Service-oriented Architecture (SOA) (Yale et al., 2016). These application architectures facilitate the transition of silo-based systems towards a service-oriented one, in which independent services provide specific functionalities and are reusable in multiple applications (Namiot & Sneps-Sneppe, 2014; Serrano et al., 2014).

Through the integration of the two IT platforms via middleware, such as an Enterprise Service Bus (ESB) (Kruiswijk, 2017b), established organizations can take advantage of the emerging tools and platforms for the development and deployment of customer-facing front-end applications, as well keep, support and leverage their existing mission-critical back-end IT systems (Horlach et al., 2016). This Bimodal IT platform will help established organizations in their digital transformation journey and to react on the fast-changing customer and user demands.

1.3. Significance and research gap

Bimodal IT is an actual subject due to the current digital age with increased digitalization, increased customer expectations, increased usage of customer-facing applications, and the large number of integration projects performed by organizations (Haffke et al., 2017a).

In addition, the concept of Bimodal IT, and related concepts such as Multi-Modal IT and Unimodal, is receiving significant attention from practitioners (Haffke et al.,2017b; Horlach et al., 2017). The reason is that the concept itself can support established organizations within their digital transformation journey by creating a capability and a culture that allows them to explore and experiment to manage uncertainties (Mesaglio & Mingay, 2015).

By considering a Bimodal Architecture implementation as a business transformation (Mesaglio & Mingay, 2015), only the approaches on how to deliver an IT project at different speeds are discussed. The architectural side, namely how to change or arrange the IT landscape and which related functional and technical aspects are needed to support the Bimodal IT transformation, are not discussed and described by the practitioners. The same applies from an academic perspective, whereas there is almost no research available on Bimodal Architecture implementations in practice (Horlach et al., 2017). Therefore, there is still a gap regarding concrete architectural and implementation details of the transition towards a Bimodal IT platform within the current literature.

To close the gap within the existing literature and to provide more insight into how a Bimodal Architecture could be realized on an enterprise architecture level or be implemented into an IT landscape, this research will describe the design patterns of a Bimodal IT platform. Concrete, this research will investigate the essential foundation of a Bimodal IT platform from a functional and technical perspective. It will describe the set-up of the platform through the identification of the elements that are part of the platform and the related aspects and elements that support the platform, such as the related delivery tools and deployment models that are essential to realize and facilitate a Bimodal IT platform within an organization.

Subsequently, based on the identified components, a Reference Architecture (RA) can be developed. In this study, a RA will be described as a generic architecture that can be applied or used by multiple organizations to shape, align, structure, verify, or complete their own architectural views and models. The developed RA can be used as a reference to comply or even to realize a Bimodal IT environment within their own IT landscape(s). Since this study will provide the first RA within the Bimodal IT field, the RA itself will not be based on proven or existing implementations. The first version of the RA will provide a high-level overview and describe the foundation of a Bimodal IT platform. It can be used as a guideline to help organizations to structure, prepare and manage future Bimodal IT initiatives. Other versions of the RA can be developed when there are new insights or when more detailed models and views are needed to support a specific organization.

In short, this study will answer how the Bimodal IT pattern could be achieved, or the related platform could be realized from an enterprise architecture point of view and deployed from a practical perspective within an organization to support their digital transformation journey.

1.4. Research question

Currently, the implementation of a Bimodal Architecture is considered as a business transformation (Mesaglio & Mingay, 2015), which can be achieved via various archetypes (Haffke et al., 2017a; Horlach et al., 2017). Bimodal IT can facilitate an environment in which organizations can experiment to manage uncertainties (Mesaglio & Mingay, 2015). However, the architectural part and the technical transformation on how the Bimodal IT concept can support the business needs is not highlighted.

As mentioned, from an architectural point of view, a Bimodal Architecture can be delineated as two separate IT platforms (Bossert et al., 2014) that are tightly integrated to meet the increased demand in digital models (Bossert et al., 2015).

Since there is almost no research available from an architectural or technical point of view (Horlach et al., 2017), this study will try to provide an answer to the following research question (RQ) regarding the establishment of a Bimodal IT platform:

What are the fundamental design patterns of a Bimodal Architecture?

The RQ zooms in on the foundation and set-up of a Bimodal Architecture. This study will investigate the components, which are minimally required, to establish the foundation and realize and facilitate a Bimodal IT platform.

The following guiding questions (GQ) will help to answer the main question within this study:

1: Which components, services, functionalities, and tools are perceived as an integral part of a Bimodal IT platform?
2: How do organizations currently implement these components to establish the foundation of a Bimodal Architecture?

With the support of the GQ's, more could be said about the implementation details of a Bimodal Architecture. Consequently, with the gathered knowledge, a RA will be created to support the implementation of a Bimodal IT platform.

1.5. Research scope

By investigating the functional and technical aspects of a Bimodal Architecture and combining this with the business aspects, the characteristics, patterns, and components could be determined, which are needed to comply or set-up and implement a Bimodal IT platform. The identified elements will also help to create a RA, which can facilitate the implementation of a Bimodal IT platform within an organization.

Since the investigation on the functional and technical aspects can be very broad, the actual scope of the research can only be defined after the literature review. However, in a broader context, the research scope will focus on the design patterns of a Bimodal IT platform. Especially on the core components that are minimal required to set-up and establish the foundation of a Bimodal IT platform. These elements can be, for example, an ESB that acts as an integration mechanism for both the IT platforms, standardization of data models via services, the introduction of agility and associated tools to speed up the development and delivery of increments (systems, applications, technologies, solutions, functionalities, features, IT products, software, etc.), and other relevant components that are required for a successful implementation of a Bimodal IT platform within an organization.

Also, the terms Bimodal IT, Bimodal IT platform, and Bimodal Architecture will be used interchangeably within this study because the terms refer to the same phenomenon. The terms indicate the coexistence of two modes or IT platforms within an organization. Consequently, the platforms themselves comprise their own groups of IT systems, in which the various IT systems are realized and implemented through a particular IT delivery style (Gartner, 2013).

The term IT system is derived from the IT definition provided by Gartner (Gartner, 2019). It will be used in this study to describe the electronic processing and distribution of data throughout an organization, including software, hardware, communications technologies, and related services. This term will cover the entire spectrum of programs, systems, applications, IT solutions, IT products, IT technologies, Information Systems (IS), and software.

1.6. Research objective and approach

This research aims to provide the elements, which are part of the design patterns needed to set-up the foundation of a Bimodal IT platform. To answer the research question, multiple research approaches will be used. These approaches and the aim of each question are stated in table 1-1.

Research Question	Objective	Approach
GQ1: Which components, services, functionalities, and tools are perceived as an integral part of a Bimodal IT platform?	To understand and identify all the relevant core components for the establishment of a Bimodal Architecture foundation	Systematic Literature Review Holistic multi-case study through interviews
GQ2: How do organizations currently implement these components to establish the foundation of a Bimodal Architecture?	To understand and define the interlinkages between the core components	Systematic Literature Review Holistic multi-case study through interviews
RQ: What are the fundamental design patterns of a Bimodal Architecture?		Based on the results and findings of GQ1 and GQ2.

Table 1-1 Research approach and objective

Figure 1-1 shows the steps that are associated with the execution of this research.

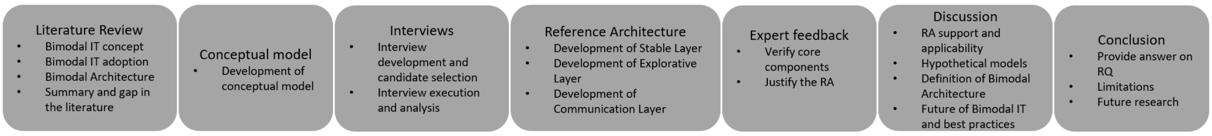

Figure 1-1 Research steps

1.6.1. Literature review

The objective of the literature review is to give an insight into the concept, characteristics, and implementation approaches of Bimodal IT. Simultaneously, the aim is to outline the patterns and components that are associated with the implementation of a Bimodal IT platform.

The literature review will be conducted via a systematic approach. The executed literature review will provide the first answers to the questions GQ1 and GQ2. In this context, the literature review will give an insight into the concept, characteristics, and current implementation methodologies. It will also expose most of the core components that are essential for the establishment of a Bimodal IT platform.

In addition, the literature review will also highlight the gaps in the existing literature. These gaps can be used to develop a set of interview questions to discover the missing information.

1.6.2. Conceptual model

The objective of the conceptual model is to provide a draft high-level design of the RA. The model will be based on the conducted literature review and can be used to develop and guide the interviews.

The conceptual model can also serve as a baseline for the development and finalization of the RA.

1.6.3. Interviews

The objective of the interview is to collect the unidentified design patterns and to determine the core components regarding the foundation of a Bimodal IT platform from the candidates who were, and still are, involved in the transformation towards Bimodal IT.

After the interviews, the GQ's can be answered.

1.6.4. Development of a Reference Architecture

In this step, the RA will be developed based on the literature review findings, the conceptual model, and the interview results.

The RA will provide a generally applicable blueprint for the set-up of a Bimodal Architecture. It will determine the essential IT platforms that are an integral part of the Bimodal IT platform and disclose its related elements. The IT platforms themselves will be referred to as "layers" within this study to distinguish the elements from the Bimodal IT platform.

In addition, the RA will also provide an insight into the relationships between the different layers, present the used software development tools to support the layers, and reveal the associated hosting platforms.

1.6.5. Expert feedback

The objective of the expert feedback is to verify and justify the identified core components and the developed RA.

The process will be performed partly together with the examined organizations that were part of the multi-case study and with organizations that did not participate in the multi-case study. The feedback will provide an answer and prove whether the identified core components can be used as a baseline for the set-up of a Bimodal IT platform and if the developed RA can be used during the implementation of a Bimodal Architecture within an organization.

1.6.6. Discussion

The discussion section will first discuss the RA and justify the research. Subsequently, it will discuss the definition of Bimodal IT and provide a new definition that can help to understand the concept of Bimodal IT from an IT architectural point of view. In addition, this section will also provide some implication practices and discuss the future of Bimodal IT.

1.6.7. Conclusion

The conclusion will provide an answer on the RQ by delivering the core components of a Bimodal IT platform through a RA. Also, the limitations and future research will be discussed in this chapter.

1.7. Literature review methodology

Through a literature review, it is possible to locate and understand the current body of knowledge regarding Bimodal IT and provide context to answer the research question.

The literature review will follow the process of a systematic approach. The reason for choosing this method is that it comprises a comprehensive pre-planned search strategy (Saunders et al., 2015). This allows to perform a detailed analysis of the existing body of knowledge on Bimodal IT. Additionally, a systematic review can also include published as unpublished literature (Tranfield et al., 2003). Since the concept of Bimodal IT is developed and widely used by practitioners, much of the existing literature on this topic is never published, such as consulting firm studies, whitepapers, and reports.

The first step is to locate and gather the existing literature regarding the topic of Bimodal IT. To achieve this, an RQ will be developed to conduct the literature review.

The second step of the literature review involves the determination of keywords to find the relevant literature. In total, 18 keywords will be defined in English. The existing literature can be located and identified within the various databases by combining the keywords into search terms. The keywords that are determined to facilitate the literature search are:

- Bimodal IT;
- Two-Speed IT;
- Agile IT;
- Digital IT;
- Bimodal Architecture;
- Bimodal implementation;
- Bimodal design patterns;
- Bimodal foundation;
- MSA;

- SOA;
- Services/Web services;
- Containerization;
- Continuous Delivery;
- Agility (to accelerate digital transformation);
- Multi-Modal IT (as an alternative for Bimodal IT);
- Software development models;
- Software Delivery Life Cycle traditional and agile model;
- Deployment and hosting models.

The third step will focus on duplications. Some keywords and search terms can locate the same literature. Therefore, these duplicates must be removed from the total search result.

The fourth step incorporates the document analysis. In this step, all the abstracts and conclusions will be screened. Only the full-text articles and publications that are addressing or have a specific focus on the concept of Bimodal IT or corresponding concepts, such as Two-Speed IT, and other related aspects, such as the used methodologies for the delivery of IT projects, drawbacks, acceleration of the digital transformation, deployment models, and the underlying components of a Bimodal IT platform will be selected for this research.

As a final step, all the selected publications will be stored and managed in the Reference Manager Tool Mendeley.

1.7.1. Literature review execution

To gather the information and literature on the concept of Bimodal IT and its underlying platform, the following RQ has been developed to conduct the literature review:

"How do current Bimodal IT models look like, which patterns and components are an integral part of the architecture, and what are their added value?"

To answer the question, a search has been conducted in the databases of the University Leiden Library, IEEE, Elsevier, Forbes, and Google Scholar. Additionally, some practitioners' websites were advised to find relevant white papers and reports, such as Gartner, IBM, Accenture, Boston Consulting Group (BCG), Microsoft, and McKinsey. To guarantee the completeness of the literature, a search on Google was performed to identify missing academic publications, whitepapers, and other relevant articles. Also, no date filter was set to ensure that all related studies, papers, articles, whitepapers, etc., could be located and identified during the search.

The total search result, based on the keywords and the used search terms, exceeded 10.000 hits. Therefore, only the first 100 hits per keyword or search term have been reviewed. It was important to exclude some hits during the review because some publications addressed Bimodal in another scientific area rather than the Business-IT domain or were focusing on a specific industry. Also, duplicated titles were removed from the total search result to ensure that all publications only occurred once.

After the search result was narrowed down, each document was screened by reading the abstract and conclusion. Only the publications that were addressing or had a specific focus on the specified criteria were selected in this process. Subsequently, the selected articles were read thoroughly to understand the content. Once this process was performed, the final selection took place. Only articles that could answer the GQ's or could contribute to the research by providing more context on the concept, characteristics, implementation approaches, deployment approaches, alternatives, continuous delivery, and components of the Bimodal IT platform, such as the layers and their associated integration patterns were selected and stored in Mendeley.

2. Literature review

This chapter presents the existing body of knowledge on the concept of Bimodal IT. The first section discusses the concept and characteristics of Bimodal IT. Subsequently, the adoption and the underlying platform of the Bimodal IT concept are discussed. And the last section will summarize the literature review and discuss the identified gap within the literature.

2.1. Bimodal IT Concept

2.1.1. What is Bimodal IT?

The term Bimodal IT was coined in 2013 by Gartner. According to Gartner, the concept of Bimodal IT can help organizations to focus on predictability and stability, while on the other side to deal with exploration and innovation (Gartner, 2013). The concept is described as "the practice of managing two separate, but coherent, styles of IT delivery. The focus is on one side on exploitation and predictability, while on the other side, it tries to deal with exploration and innovation." To underpin this concept, Gartner introduced Mode 1 and Mode 2. Mode 1 focusses on the exploitation and renovation of the legacy environment, also referred to as "systems of records" by Horlach et al. (2016), so it can fit within the digital world. Mode 2 focusses on innovation and the "new." Horlach et al. (2016) define these customer-facing applications as "systems of engagement" (Horlach et al., 2016). Therefore, this mode is explorative and needs an agile way of IT delivery to meet customer demands and expectations (Gartner, 2013). The concept of Bimodal IT supports organizations to exploit its existing legacy environment(s) and, on the other hand, to deal with the adoption and adaptation of customer-facing applications. By applying this concept, established organizations will become more flexible and able to respond quickly to changing market needs.

2.1.2. What is Two-Speed IT?

Within the literature, Bimodal IT is also referred to as "Two-Speed IT" (Bossert et al., 2014; Haffke et al., 2017b; Horlach et al., 2016). According to McKinsey, a Two-speed Architecture implies "a fast-speed, customer-centric front-end running alongside a slow-speed, transaction-focused legacy back end" (Bossert et al., 2014). From a software deployment and release cycle point of view, the fast-speed architecture should be modular to enable quick implementation of new software, avoiding time-consuming integration work. In contrast, the slow-speed architecture deals with transactional core IT systems (systems of records), which will lead to a longer release cycle because these IT systems are designed for stability and high-quality data management (Bossert et al., 2014).

2.1.3. Traditional IT and Digital IT

Bils (2014) argues that the concepts of Bimodal IT and Two-Speed IT should be embedded into two distinctive IT units instead of having a single IT organization (Bils, 2014). Through the creation of two IT units, where Mode 1 is referred to as "Traditional IT" and Mode 2 as

"Agile/Digital IT" (Bils, 2014; Horlach et al., 2016), the Traditional IT unit can focus on the business-critical IT systems and ensure a stable and reliable IT environment. This IT environment will also incorporate predictability, scalability, operational excellence, cost savings, and the automation of processes and services (Bils, 2014; Gartner, 2013; Horlach et al., 2016; Rivera, 2014). In contrast, the Digital IT unit can emphasize on agility and speed (Gartner, 2013; Horlach et al., 2016).

The priority of the Digital IT unit should center around value creation. This means that the unit must consider all the needs and demands of the business units, customers, and (external) partners and drive innovation (Horlach et al., 2016; Kirschner & Kenney, 2014; Rivera, 2014). To achieve this, operations need to support short release cycles, be non-linear, and make use of iterative and agile principles (Bils, 2014).

2.1.4. Comparison of Bimodal IT, Traditional IT, Digital IT, and Two-Speed IT

By comparing the concepts of Bimodal IT, Two-Speed IT, and the segregation of the IT organization into two distinctive IT units, it becomes clear that all three concepts emphasize the same phenomenon.

The concepts are describing a contradiction between stability and reliability versus speed and innovation within the IT domain. To stay competitive and to cope with the rapidly changing environment, the established organizations need to focus on the existing IT systems, while they also need to introduce exploration and experimentation within the IT domain (Bils, 2014; Bossert et al., 2014; Gartner, 2013).

By dividing the IT systems into two groups or IT platforms, business-critical versus market-facing IT systems and services (Bils, 2014), the concepts can be used interchangeably because they serve the same purpose.

To prevent confusion, and because the initial concept is developed by a leading research and advisory company, namely Gartner, the concept of Bimodal IT will be used throughout the research.

2.1.5. Bimodal IT characteristics

As mentioned in the previous section, the concept of Bimodal IT distinguishes two modes. Because each mode fulfills a different purpose within the organization, it has its own set of attributes. The characteristics of each mode are summarized in table 2-1.

Mode 1 (Traditional IT)		**Mode 2 (Digital IT)**
Stability	*Goal*	Agility & Speed
IT-centric	*Culture*	Business-centric

Remote for customer	*Customer proximity*	Close to customer
Performance and security improvement	*Trigger*	Short term market trends
Performance of services	*Value*	Business moments, customer branding
Security & reliability	*Focus of services*	Innovation
Waterfall development	*Approach*	Iterative, agile development
Systems of records	*Applications*	Systems of engagement
Slow	*Speed of service delivery*	Fast

Table 2-1 Characteristics of Mode 1 and Mode 2 (Horlach et al., 2016)

2.2. Adoption of Bimodal IT

2.2.1. Reasons to adopt Bimodal IT

The traditional way of working and the need for change

Established organizations that work in a traditional way use a detailed plan to deliver an IT system. Therefore, any changes during the project need to go through a strict change control management and prioritization process (Stoica et al., 2013).

In this study, the explanation of Stoica et al. (2013) will be used to define the traditional way of working. The traditional way of working can be defined as a predictive approach for delivering IT systems that depend entirely on the provided requirements and careful planning at the beginning of an IT project. During the execution of the project, no new requirements are taken into account until the IT system is released into production unless there are requirements (changes) that pass the change control management and prioritization process.

With the rise of digitalization and increased use of digital channels, established organizations are compelled to transform their business and operating models to stay competitive and engaged with their environment (Ghareeb & Berchten, 2011; Haffke et al., 2017a). By developing and adopting a digital strategy, organizations can become more responsive to internal and external pressures from customers, suppliers, partners, and employees (Ghareeb & Berchten, 2011). This change, within the organizational operations, also affects the internal IT organization. In most cases, the internal IT organization needs to become more flexible and adaptable to keep up with the internal and external demands and keep the business-critical IT systems up and running.

The necessity to become agile

Organizations use the concept of agile to cope with unexpected changes, survive unprecedented threats, and to take advantage when opportunities arise (Zhang & Sharifi, 2000). Because the concept has proven its importance and disruptiveness, agility has become a necessary phenomenon within today's business (Denning, 2020).

But what is agility, and what does it mean? The concept of agility originated in the 1960s and 1970s and started with the set-up of self-managing teams (Denning, 2020). Dennis (2016) explains that agility is about "embracing a different mindset" (Denning, 2016). It involves "a different way of thinking as much as a different way of doing" (Denning, 2020). Other researches define agility as the "ability to sense and respond quickly to opportunities in rapidly changing environments" (Mathiassen & Pries-Heje, 2006; Oosterhout et al., 2006; Overby et al., 2006; Roberts & Grover, 2012).

This study will make use of the definition of Conboy (2009). Conboy (2009) has reconstructed the concept and applied it within IS development. According to Conboy, Agility is "a method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" (Conboy, 2009).

Agility has become a key driver to accelerate digital transformation (Newmann, 2017). The introduction of agility within an IT organization can help to respond quicker to the digital initiatives of an organization (Bossert et al., 2015). Via two parallel IT tracks, the organization can deal with new projects and incrementally build solutions in short release cycles using an agile framework. While also support waterfall-driven projects to leverage their legacy environment(s) (Haffke et al., 2017a).

The importance of IT exploration and experimentation

Digital transformation requires the embracement of innovation (Ghareeb & Berchten, 2011) and continuous focus on identifying and exploiting new opportunities (Denning, 2016). However, this model conflicts with the traditional way of working, where the primary focus of the IT organization is on the exploitation of a cost-efficient and reliable IT environment (Henthorn-Iwane, 2015; Horlach et al., 2016).

By adopting an agile mode, next to the traditional mode, the IT organization can enable quick delivery of high-quality products via a new way of working that includes agile development, a culture that accepts failures, and approaches that encourage exploration and experimentation (Andersson & Tuddenham, 2014; Haffke et al., 2017a).

The need for Business Alignment and Business Support

Several kinds of research indicate that IT can be an enabler for organization agility (Lu & Ramamurthy, 2020; Oosterhout et al., 2006; Overby et al., 2006).

However, the reality is that established organizations experience a gap between the IT function and the other organizational disciplines (Mathiassen & Pries-Heje, 2006). This phenomenon is also referred to as the "IT paradox" (Morgan, 2004). This concept describes that the higher management recognizes the value of having an effective IT unit, but they lack to understand how the IT function enhances and contributes to the business value (Mathiassen & Pries-Heje, 2006; Morgan, 2004). To bridge this gap, an IT strategy is required that is aligned with the business goals, diffuses on IT innovations, and support active collaboration between the IT function and the various disciplines (Desmet et al., 2016; Ketterer et al., 2016; Mathiassen & Pries-Heje, 2006).

Bimodal IT can help to close this gap. The concept provides a strategy that maintains the principles of Traditional IT through the realization of a stable IT infrastructure and operations. And at the same time, it enables organizational agility through the rapid development of digital innovations and delivery of support to disciplines with a digital mandate to meet and support the business goals (Haffke et al., 2017a; Roy et al., 2015).

2.2.2. Bimodal IT as an intermediate state

Since the introduction of Bimodal IT in 2013, a lot of practitioners have developed their own concepts and frameworks around it. Some practitioners and Chief Information Officers (CIOs) even discusses that Bimodal IT is just a temporary or intermediate state for evolving to a higher stage of IT delivery, also known as "Multi-Modal IT" or "Multi-Speed IT" (Bayley & Shacklady, 2015; Colbert, 2016; Hinchcliffe, 2016; Keller et al., 2018). Within the Multi-Modal IT concept, the modes are not limited to Mode 1 and Mode 2. It incorporates multiple modes, as required by an organization, and can impact different areas of the IT architecture (Hinchcliffe, 2016; Keller et al., 2018).

This evolution is also understandable from a business perspective because most practitioners position Bimodal IT as a business operating model with two different speeds within an organization. Each mode has its own IT delivery approach, waterfall-driven versus agile delivery (Ketterer et al., 2016; Ram, 2017). Due to these deviated speeds, some practitioners indicate that the concept of Bimodal IT is earlier harmful rather than useful (Horlach et al., 2016; Ketterer et al., 2016). By dividing the IT organization into two modes of speeds, the risk exists that the collaboration between Traditional IT and Digital IT is negatively influenced and eventually jeopardized. This duopoly can lead to new organizational silos (Horlach et al., 2016; Katz, 2015). There is also a fear that the recruitment and retainment of talents will become a challenge for Traditional IT due to the perspectives this mode offers to the (new) workforce (Ketterer et al., 2016). Last, the fast delivery of digital initiatives could be affected. From a technical point of view, most digital initiatives interact with the organizational core back-end IT systems. Therefore, the governance structures and rules of Traditional IT can delay the implementation and deployment of digital increments due to rigid test and release cycles

(Haffke et al., 2017b; Horlach et al., 2016; Ketterer et al., 2016). Based on these reasons, it could be argued if the concept of Bimodal IT can serve as a long-term solution (Bayley & Shacklady, 2015; Ketterer et al., 2016).

Multi-Speed IT

To mitigate the problems of Bimodal IT, some practitioners and CIOs are proposing to move towards a multi-facet IT operating model, also referred to as "Multi-Speed IT" (Bayley & Shacklady, 2015; Franken et al., 2016; Radcliffe, 2017). The concept of Multi-Speed IT enables organizations to make changes at the speed they require (Radcliffe, 2016). By blending various delivery approaches, such as Scrum, DevOps, V-Model, and Waterfall delivery methods, an organization can choose the proper operating and delivery method for each type of change, and at the same time, govern the strategic goals and objectives (Bayley & Shacklady, 2015; Ram, 2017).

All-Agile approach

According to the BCG, organizations should move towards a concept of one speed. They refer to this concept as All-Agile (Ketterer et al., 2016). BCG (2016) discusses that Bimodal IT will ultimately divide the organization into two groups: "fast" and "slow," with its own distinctive culture and delivery approaches. This will disrupt the potential benefits of agile (Ketterer et al., 2016). However, through the adoption of a single-speed approach, based on agility, an organization will be able to cope with all the problems of Bimodal IT, and deliver a better customer experience and increase their competitiveness and resilience.

2.2.3. Comparison of Bimodal IT, Multi-Speed IT, and All-Agile

As mentioned in the previous section, the concept of Bimodal IT can be harmful for an organization because the collaboration between Traditional IT and Digital IT can fail due to organizational silos, diverse workforces, and delay in the acceptance of digital increments (Bayley & Shacklady, 2015; Haffke et al., 2017b; Horlach et al., 2016; Katz, 2015; Ketterer et al., 2016). Therefore, the established organizations should shift towards a multi-modal or unimodal concept (Bayley & Shacklady, 2015; Ketterer et al., 2016; Radcliffe, 2016; Ram, 2017).

While comparing the concepts, the conclusion can be drawn that from an architectural point of view, the established organizations are still making use of two separate IT platforms. One of the IT platforms focuses on IT systems that need to be stable and reliable, whereas the other IT platform covers the IT systems that depend on speed, innovation, and agile developments. Therefore, the concept of Bimodal IT is still applicable throughout the research.

2.2.4. Bimodal IT Alignment, Governance, and Management

Bimodal IT can help to bridge and minimize the gap between IT and organizational needs.

The concept extends the Business-IT alignment mechanisms, as reflected in the Strategic Alignment Model (Henderson & Venkatraman, 1993), in two ways (Horlach et al., 2016). First, it imposes an alignment between the Traditional IT unit and the Digital IT unit. This alignment ensures that the development and delivery of the digital increments are not slowed down or hindered. Second, the concept enforces an alignment of the strategical and operational activities between the two modes and the other organizational disciplines. Through the decentralization of the IT organization, a significant part of the Digital IT unit can become a part of the business units to enhance and support organizational agility (Horlach et al., 2016).

To cope with these adjustments, new governance and alignment mechanisms need to be developed and incorporated to achieve the required Business-IT alignment (Horlach et al., 2016). In addition, transparent and involved senior-leadership, enablement of self-organized teams, integrated processes, structures, skills, methods, IT architectures, IS infrastructures, and focus on innovation are necessity key factors to succeed within the Bimodal IT transformation (Haffke et al., 2017a; Horlach et al., 2016).

Software Development Life Cycle

A Software Development Life Cycle (SDLC) is a conceptual framework or process to support the development and maintenance of a software system (Baratchi, 2018a; Leau et al., 2012). It includes the models and methodologies to develop a software system and describes the set of activities performed in each phase of the software development process (Leau et al., 2012; Ruparelia, 2010).

The SDLC framework is mainly supported by four software development methodologies, namely (Baratchi, 2018a):

- Predictive: All features are provided at the same time with full fidelity;
- Iterative: All features are provided at low fidelity. Through iterations, higher fidelity is realized until the features have reached their full fidelity;
- Incremental: Provides only the most essential features of a usable application or IS with full fidelity. Later versions are used to add new features with full fidelity;
- Agile: Provides only the most essential features at low fidelity. Later versions improve the fidelity of the existing features and add new features until the application or IS has reached its goal.

In practice, the SDLC methodologies are utilized through traditional software development or agile software development (Leau et al., 2012; Stoica et al., 2013).

Traditional Software Development

Software development models, such as Waterfall, and V-Model, are part of the traditional software development methodology (Baratchi, 2018a; Leau et al., 2012; Ruparelia, 2010). Traditional methodologies are predictive and underpinned by clear and well-defined requirements (Baratchi, 2018a; Stoica et al., 2013). Therefore, the models are heavily relying on plans, processes, change control management, and comprehensive documentation (Leau et al., 2012; Stoica et al., 2013). The models follow a predetermined sequence of stages, in which the completion of one phase leads to the initiation of the next phase (Ruparelia, 2010).

Traditional models encompass the following phases (Baratchi, 2018a; Ruparelia, 2010; Sahil et al., 2017; Stoica et al., 2013):

- Requirements;
    - Analysis and Definition;
- Design;
- Implementation
    - Development or Build;
- Verification;
    - Testing and Validation;
- Deployment;
- Maintenance;
    - Operation, Evolution, and Optimization.

Traditional SDLC models are most suitable for projects that contain a stable product definition, and the requirements are well-known and well-defined, which is the case in large-scaled projects (Baratchi, 2018a; Leau et al., 2012; Stoica et al., 2013).

Agile Software Development

Agile software development methodologies are based on a combination of iterative and incremental processes, such as Extreme Programming (XP) and Scrum (Baratchi, 2018a; Stoica et al., 2013). The focus is on delivering software as soon as possible by breaking the software systems into small increments (Baratchi, 2018a).

The agile SDLC phases are revisited multiple times to ensure that a software system is improved over time and new features are added gradually (Leau et al., 2012). Agile software development methodologies emphasize on customer collaboration, self-organizing teams, minimal documentation, responsiveness to change, and working software (Baratchi, 2018a; Leau et al., 2012).

Compared to traditional software models, agile SDLC models are more suitable for small- and medium-scaled projects (Leau et al., 2012; Stoica et al., 2013). Since the agile methodologies are using an adaptive approach (Baratchi, 2018a; Leau et al., 2012), the methodologies can deliver quick results without emphasizing on well-documented requirements (Leau et al., 2012).

In practice, the agile software development methodologies could be broken down in certain steps. Figure 2-1 presents these steps. Due to the iterative nature, multiple stages can be simultaneously in progress to speed up the development process of a software system.

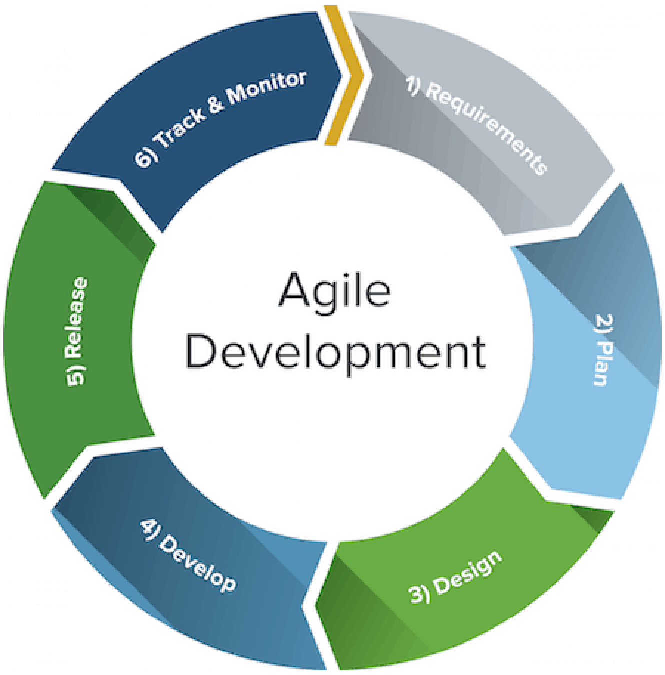

Figure 2-1 Steps to realize agile software development (Sahil et al., 2017)

Service-oriented Development

The SOA and MSA concepts are dealing with the monolithic application architecture by breaking it down into a set of services (Namiot & Sneps-Sneppe, 2014; Pahl, 2015). Services are loosely coupled, independent, and can be rapidly assembled to support various business processes (Kruiswijk, 2017a; Pahl, 2015). These characteristics lead to a service-oriented approach in which services are continuously and automatically deployed and grouped, through orchestration, to compose distributed applications (El-Sheikh et al., 2016; Xiao et al., 2016).

Continuous Integration

Continuous Integration (CI) supports the increase of organizational agility by minimizing the time to market of digital initiatives. The approach allows to build, compile, test, and integrate software functionalities automatically and on a regular base (Fowler, 2006; Olsson, Alahyari, & Bosch, 2012).

CI is described as a software development practice in which team members are enforced to integrate their work frequently, preferably multiple times per day (Balalaie et al., 2016; Fowler, 2006; Olsson et al., 2012). This action ensures that when a change is committed to version control, the codebase of the member is automatically integrated with the existing software codebase (Aderaldo et al., 2017). Through this integration approach, conflicts in the software code can be reduced and enable an organization to build and launch cohesive applications more rapidly (Fowler, 2006).

To define CI in this study, the description of Balalaie et al. (2016), Fowler (2006), and Olsson et al. (2012) will be used.

Continuous Delivery

Continuous Delivery (CD) is the next step after adopting CI. CD is a practice to enable on-demand and constant deployment of software functionalities (Balalaie et al., 2016; Olsson et al., 2012). The process supports the build, test, configure and deploy activities from a development to a production environment. When applying CD, the source code, configuration, and the application's environment must be separated from each other to ensure that each component layer can evolve independently (Balalaie et al., 2016; Humble & Farley, 2010) and can be released at any time (Baratchi, 2018b).

Virmani (2015) discusses that CD "tries to optimize the infrastructure management and the critical need to balance out time and resources (Virmani, 2015)."

This study will make use of the definition of Baratchi (2018b). Baratchi (2018b) indicates that CD supports the development of software in a way that it can be released at any time (Baratchi, 2018b).

Continuous Monitoring

A critical practice for ensuring CD is Continuous Monitoring (CM) (Balalaie et al., 2016). CM bridges the gap between development and operations. Therefore, CM is not only providing quality and performance-related feedback but also facilitates the detection of operational anomalies to prevent bottlenecks and problems within the infrastructure (Balalaie et al., 2016; Virmani, 2015).

2.2.5. Bimodal IT design

There are several ways to implement a Bimodal IT design. However, there are three archetypes, (A), (B), and (C), which are used the most in practice when introducing Bimodal IT internally within an organization (Haffke et al., 2017b). The first level of design is based on the development process of each IT project (archetype A). The second level introduces an operational split, between Mode 1 and Mode 2, within the IT organization (archetype B). And in the third level, Mode 2 is accommodated in a separate division within the organization (archetype C) (Haffke et al., 2017b). The three Bimodal IT designs are represented in figure 2-2.

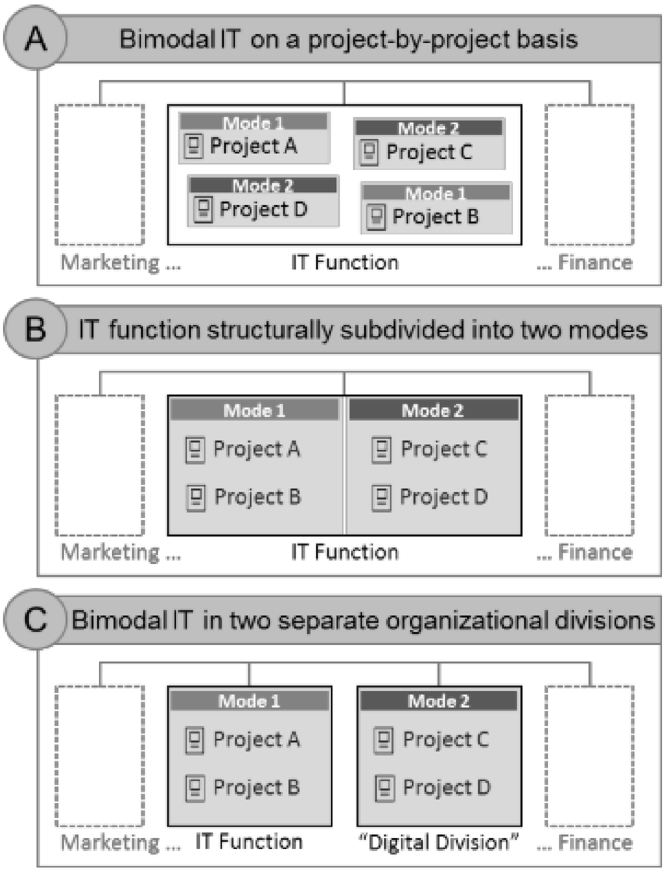

Figure 2-2 Archetypes of the three Bimodal IT designs (Haffke et al., 2017b)

Next to these three archetypes, an organization can also choose to outsource some IT activities because IT is not considered as their core business (Haffke et al., 2017a). Consequently, the organization has to decide which part to outsource, only Mode 2 (Archetype D) or both of the IT modes (Archetype E) (Horlach et al., 2017). Alternatively, an organization can also decide to transform its IT organization into a Unimodal, Agile IT organization. The responsibility of this IT organization is to increase business agility and time-to-market so that the organization can react and respond to the rapidly changing market(s) (Archetype F) (Haffke et al., 2017a; Horlach et al., 2017). All six archetypes will be discussed in the upcoming sections.

Archetype A: Project-by-Project Bimodal IT

The project-by-project Bimodal IT design is frequently used within organizations that are reluctant to engage in significant or massive changes (Haffke et al., 2017a). The design follows the traditional process-driven development methodologies and allows the IT organization to utilize agility for the development of fast-changing customer-centric applications (Horlach et al., 2017). At the start of each project, the project team can choose whether they want to use the traditional or the agile mode (Haffke et al., 2017a).

During project execution, interactions with other project team members, project board(s), or interdisciplinary steering committee(s) are encouraged to enable alignment and mitigate potential risks. Consequently, Mode 2 will follow the release cycle of Mode 1 for the deployment of the digital initiatives. This approach will ensure that failures and mismatches are minimalized during the implementation and acceptance of the increments within the IT environment (Horlach et al., 2017).

In short, this design allows the introduction of agility within an IT organization gradually (Haffke et al., 2017a).

Archetype B: Subdivisions Traditional IT and Digital IT

This archetype characterizes the separation of the two modes. To support the organizational digital initiatives, an organization and its business units can benefit by splitting up the structures and processes of the IT organization into two distinctive groups (Haffke et al., 2017a). One group, referred to as the "Traditional group," can continue exploiting the traditional IT infrastructure(s), services, and operations. In contrast, the second group, also known as the "Digital/Agile group," can focus on exploration, experimentation, and innovation (Haffke et al., 2017a).

It is important to note that each group requires a different skillset from its resources. Therefore, the set-up of this design can lead to tensions between the two groups because each group has its own culture, structures, and procedures (Haffke et al., 2017a). Clear leadership, interaction, communication, and cooperation are essential key elements to bridge these differences and to enable the success of the Bimodal IT design.

Archetype C: Digital IT as a separate division

In this design, the two modes are also split-up regarding the structures and processes. However, it is embedded into a separate division, also referred to as a "Digital Division," instead of in the traditional IT function (Haffke et al., 2017a). In most organizations, the leadership execution is also separated by making a Chief Digital Officer (CDO) responsible for the Digital Division and the CIO responsible for the traditional IT organization (Haffke et al., 2017a; Horlach et al., 2017).

Through the use of Scrum, the team members of the Digital Division are fostered to work in small (five to ten people) interdisciplinary teams that represent the business and IT (Horlach et al., 2017). Also, by including IT operational members into the teams through the use of the DevOps methodology, the Digital Division can operate separately from the traditional IT organization (Horlach et al., 2017). This team set-up will strengthen and secure the alignment between the development of the digital increments and the acceptance of the increments within the organizational core IT infrastructure.

The goal of the Digital Division is to ensure a "higher agility, flexibility and reactivity towards customers" (Horlach et al., 2017). Therefore, the role of the Digital Division is to decrease the distance between the organization and its ambitions, so IT is considered as a partner instead of a service provider (Horlach et al., 2017).

Archetype D: Outsourcing of Digital IT

Some organizations realize that they cannot implement the required IT agility themselves (Horlach et al., 2017). Occasionally, it also happens that an organizational discipline establishes a parallel IT organization within its own unit for the development of digital initiatives. To achieve this, they seek help from external parties without involving or informing the internal IT organization because the discipline is not convinced that the internal IT organization can meet their expectations and needs (Horlach et al., 2017).

To avoid these practices and cope with the rapidly changing environment, an organization can cooperate with an external party or establish a subsidiary to get the IT agility they need. This approach results in a partly outsourced IT organization. In this Bimodal IT design, the Traditional IT is performed in-house, whereas Digital IT is supplied externally by one or more third parties or via a subsidiary (Horlach et al., 2017). This archetype can help to increase the reliance that "IT can deliver a solution for their needs" (Horlach et al., 2017).

The alignment within this design is mainly achieved through contracts and agreements between the third-party and the organization (Horlach et al., 2017).

Archetype E: Outsourcing of both IT Modes

Organizations also have the option to outsource both of the IT modes (Horlach et al., 2017). This design can be interesting for organizations that do not develop and maintain their own IT infrastructure(s). The outsourcing of Traditional IT and Digital IT can be realized in the following two ways (Horlach et al., 2017):

1. A client-supplier relationship between the organization and the service provider(s);
2. The IT projects are led by the internal IT organization, while the IT resources for Traditional IT and Digital IT are supplied by one or more external parties.

In the first set-up, the organizational units are the client, and one or more external parties or subsidiaries are responsible for delivering the Traditional IT and Digital IT capabilities (Horlach et al., 2017). To control and guarantee the alignment and delivery of the IT projects, the internal IT organization gathers and provides the requirements of the specific services and acts as a governance instance to monitor the progress of all the ongoing projects (Horlach et al., 2017).

In the second set-up, the internal IT function is represented by one or more in-house project managers. These project managers are responsible for the steering and delivery of the Traditional IT and Digital IT projects (Horlach et al., 2017). In contrast, the project team(s) resources are supplied by one or more external parties. The advantage of these IT resources is that they are already equipped with the necessary skills to execute the two IT modes within an organization (Horlach et al., 2017).

Archetype F: Agile IT Organization

With this design, an organization can entirely focus on its digital objectives while eliminating the relationship of the traditional IT operational activities with the organizational units (Haffke et al., 2017a).

The transformation towards a Unimodal, Agile IT organization introduces an iterative development, delivery, and change management process for organizational IT initiatives, including modifying and optimizing business-critical IS (Horlach et al., 2017; Ketterer et al., 2016).

In addition, the archetype allows the establishment of Digital Business Units by integrating the resources of the Digital IT unit inside the organizational units (Horlach et al., 2017). Through the use of autonomous agile interdisciplinary teams, a fast delivery of high-quality IT increments can be ensured (Horlach et al., 2017).

To summarize, this design can help drive organizational agility and increase time-to-market via a rapidly responding IT organization (Haffke et al., 2017a; Horlach et al., 2017; Ketterer et al., 2016).

2.3. Bimodal Architecture

2.3.1. Bimodal IT layers and integration

Due to the emergence of digitalization, there is a raising need for customer-facing applications that can be built rapidly via simple services (Bils, 2014; Horlach et al., 2016; Xiao et al., 2016). At the same time, established organizations are running rigid core back-end IT systems with complex business rules, complicated relationships with other back-end and front-end IT solutions, and additional features that inhibit rapid development within their IT environment(s) (Xiao et al., 2016). The baseline of the Bimodal IT concept is that it leads to two different IT architectural layers of speed (Bils, 2014; Bossert et al., 2014). One IT layer focuses on the customer-centric IT systems. It provides the speed, agility, and IT innovations to support the organization with the new and changing business and technological conditions, such as Blockchain, Cloud, and Internet of Things (IoT). In contrast, the second IT layer focusses and provides support on the monolithic business-critical back-end IT systems.

However, on an enterprise application architecture level, the two IT layers of speed also interact with each other by exchanging objects and data and provide services to support various business functionalities (Franken et al., 2016; Yale et al., 2016). Therefore, established organizations need to shift their focus towards the rapid delivery of Application Programming Interfaces (APIs) (Xiao et al., 2016), such as web services (Kruiswijk, 2017a; Serrano et al., 2014).

Ideally, to reduce integration and maintenance efforts, increase agility, and to make IT more flexible and customer-centric, established organizations have to move from a monolithic, silo-based IT architecture towards a modular, component-based distributed IT architecture (Serrano et al., 2014; Xiao et al., 2016; Yale et al., 2016). By wrapping the legacy systems in a service layer, the business-critical core IT systems can be plugged into a service-oriented environment (Serrano et al., 2014). A MSA or SOA could assist in this transformation (Namiot & Sneps-Sneppe, 2014; Serrano et al., 2014; Yale et al., 2016).

2.3.2. Monolithic Architecture

A monolithic architecture is a client-server architecture that comprises the presentation, application, and database layers (Saraswathi, 2020). This architecture is also known as a 2-, 3- or N-tier architecture, in which the User Interface (UI) is handled through a client, such as a personal computer (PC), tablet, or mobile, and the business and data access logic through one or more servers (Saraswathi, 2020).

Dragoni et al. (2017) argue that a monolithic application can be defined as a software application that consists of tightly integrated modules, which cannot be executed independently because the modules share the same resources, such as memory and database(s) (Dragoni et al., 2017; Kratzke, 2018).

The problem with monolithic applications is that it becomes too large and sophisticated over time. Also, every change requires the adjustment and maintenance of all the layers of hardware and software. Therefore, this model has become inefficient and outdated within the cloud era (Saraswathi, 2020).

2.3.3. Service-oriented Architecture

SOA is a design paradigm and discipline for integrating business processes and supporting IT infrastructures within a landscape of heterogeneous systems that are under the control of different owners (Kruiswijk, 2017a). It is a method that allows various service types to interact independently with each other to address changing business needs and priorities.

Conclusively, a SOA has an enterprise scope and is responsible for the integration patterns between IT systems (Clark, 2018b).

Benefits of SOA

SOA offers several benefits to an organization. These benefits are (Kruiswijk, 2017a; Serrano et al., 2014):

- Enables modularization of complex systems through the integration of services from different vendors independent of platform and technology;
- Promotes loose coupling;
- Increases the efficiency by allowing applications to be reused;
- Improves flexibility and scalability;
- Promotes and improves interoperability between heterogeneous systems;
- Reduces maintenance costs;
- Enables agility via an incremental approach for adding new services in response to specific business needs.

Core components of SOA

The core components of a SOA environment are (Xiao et al., 2016):

- Services;
- Enterprise Server Bus (ESB);
- Service Registry.

Services

A service is a small, independent, and self-contained piece of functionality. It is built around a set of well-defined principles and adds value within a business context. Services facilitate loose coupling, composability, reusability, and interoperability between heterogeneous systems (Kruiswijk, 2017a; Xiao et al., 2016; Yale et al., 2016).

In this research, a service represents a small, independent, and self-contained piece of functionality and will refer to a business service, which indicates that the service is meaningful from a business perspective.

Enterprise Service Bus

A common way to implement services within a SOA environment is through an ESB (Xiao et al., 2016; Yale et al., 2016). Within the SOA environment, the ESB acts as a central hub for the connectivity and logical transformations, such as data mapping (Kruiswijk, 2017b). The ESB ensures that the service providers and consumers can communicate with each other. In addition, the ESB also provides tooling for administration, security, deployment, and monitoring, which helps to reduce costs, complexity, and integration problems (Serrano et al., 2014).

Service Registry

Services can be managed and monitored via a Service Registry tool. The Service Registry acts as a central repository and provides service discovery capabilities, lifecycle management, analytics, and virtualization. The tool also provides governance functions to enforce standards, policies, and contracts (Xiao et al., 2016).

2.3.4. Microservice Architecture

A MSA is derived from the concepts of SOA (Dragoni et al., 2017). This “new” software architecture deals with the problems that are faced within a SOA environment (Serrano et al., 2014). Whereas SOA emphasizes on services, the focus of the MSA is on the development of distributed and modularized application components, which are referred to as microservices (Xiao et al., 2016; Yale et al., 2016). Within this context, a microservice is a service-oriented application component, which is entirely independent in development, deployment, and scalability (Gartner, 2011).

Conclusively, a MSA has an application scope and deals with the internal parts of an individual IT system to ensure that they are resilient, scalable, independently deployable, and highly evolvable (Aderaldo et al., 2017; Clark, 2018b).

Benefits of a Microservice Architecture

A microservice aims to strip away the unnecessary levels of complexity to implement a single independent functionality (Dragoni et al., 2017). The advantages of microservices, in contrast to SOA, are (Dragoni et al., 2017; Xiao et al., 2016; Yale et al., 2016):

- Manages growing complexity by functionally decomposing systems into a set of services;
- A microservice is built around a bounded context in which related functionalities are combined into a single business capability, which is then implemented as a service;
- Each microservice is operationally independent of other services and is deployed independently;
- Improves maintainability;
- Quickly and easily implementation of asynchronous communication between systems.

Core components of a Microservice Architecture

Microservices have emerged as an architectural style for building distributed (cloud) applications that are resilient, scalable, independently deployable, and highly evolvable (Aderaldo et al., 2017). Each microservice is composed of three layers, an interface layer, a business logic layer, and a data persistence layer (Xiao et al., 2016; Yale et al., 2016), and runs in its own process and communicates via a lightweight mechanism (Aderaldo et al., 2017). Therefore, a microservice can be described as a small application that provides an independent, cohesive function or operation through the interaction of messages (Dragoni et al., 2017).

The research will use the definition of Dragoni et al. (2017) to describe a microservice.

A MSA structures an application as a set of loosely coupled collaborating services (Dragoni et al., 2017). To establish a MSA environment, on an enterprise level, the following design patterns and components are desirable within the architectural design (Aderaldo et al., 2017; Balalaie et al., 2016; Sill, 2016):

- Virtualization via Container technology;
- Container Orchestration;
- Versioning;
- Circuit Breaker;
- REST API;
- API Gateway;
- Automated Testing;
- Continuous Integration.

Container

A container provides an isolated and secure environment for running individual applications (Jaramillo et al., 2016). It encapsulates the application into a single executable unit of software by bundling the application code together with all the related configuration files, libraries, and dependencies so that it can run from any device (Pahl, 2015).

Commonly, containers are referred to as a lightweight virtualization technology because it shares the operating system (OS) resources with other containers on the same computing system (Jaramillo et al., 2016).

Containers are not required within the development and deployment process of microservices (Sill, 2016). However, when a service is deployed as a container, it will automatically inherit all the benefits that containerization has to offer, such as portability, scalability, fault isolation, hardware efficiency, automation of installation, and ease of management (Jaramillo et al., 2016). Conclusively, containerization will result in a faster development and more secure deployment of application components, independently of the underlying infrastructure (Aderaldo et al., 2017).

Container Orchestration

As mentioned in the previous section, microservices can be deployed as containers. This approach can cause an increase in container volume.

Container Orchestration is an approach to handle and manage large volumes of containers. A Container Orchestration platform can be used to automate the management, orchestration, scheduling, and deployment process of individual containers and scale-out a group of containers into an application (Pahl, 2015). In addition, the platform also offers a range of other capabilities, such as provisioning, redundancy, health monitoring, resource allocation, scaling, load balancing, file storage, and service discovery (Aderaldo et al., 2017; Pahl, 2015).

Version Control

Versioning is a crucial aspect throughout an applications’ lifecycle (Aderaldo et al., 2017). It enables the creation and management of multiple releases for a specific software package. Version Control provides an insight into the release history and can also help to roll-back to an older version of the application (Aderaldo et al., 2017; Jaramillo et al., 2016).

Circuit Breaker

Failed requests can lead to a bottleneck because the queue can be accumulated with pending requests. This occurrence might hold up critical system resources such as memory, threads, and database connections, resulting in cascading failures of other resources and eventually bringing the entire system down (Montesi & Weber, 2016). A Circuit Breaker can avoid this problem by monitoring and preventing a service from repeating its operation that will probably fail (Montesi & Weber, 2016).

RESTful API

Representational State Transfer (REST) is a common style to implement microservices (Dragoni et al., 2017; Mumbaikar & Padiya, 2013). REST APIs are built around resources, which can be an object, data, or service, and can be accessed via a client (Mumbaikar & Padiya, 2013).

API Gateway

An API Gateway is a single-entry point for all clients and is located between the clients and services. It guards the boundaries of an IT system and controls how it makes itself available to other IT systems within and beyond the organization (Clark, 2018a). The Gateway is responsible for handling API calls through the routing of requests and responses between a client and a collection of back-end services (Montesi & Weber, 2016; Namiot & Sneps-Sneppe, 2014). Besides routing, an API Gateway can also perform tasks, such as authentication, SSL termination, and throttling (Montesi & Weber, 2016; Xiao et al., 2016; Yale et al., 2016).

2.3.5. Serverless Architecture

Serverless Architecture is introduced in 2014 by Amazon as an alternative for hosting applications in the cloud (Lloyd et al., 2018). It eliminates the infrastructure management tasks by integrating the support activities, such as scaling, scheduling, patching, provisioning, etc., as features into the platform (Lloyd et al., 2018; Saraswathi, 2020).

The architecture incorporates third-party back-end components, such as compute, databases, storage, stream processing, message queuing, etc., and utilizes it in containers on a “Functions as a Service” (FaaS) platform (Kratzke, 2018). This allows developers to run their code on-demand and at scale in the cloud, independent of the used programming language. Therefore, serverless architecture can be described as an approach in which microservices are defined, orchestrated, and ran as containerized applications without the need to manage the underlying infrastructure (IBM Cloud Education, 2019).

Benefits of a Serverless Architecture

The advantages of a serverless architecture are (IBM Cloud Education, 2019; Kratzke, 2018):

- It enables the focus on coding instead of managing the infrastructure;
- Reduces hosting/operational costs because a user only pays when an action occurs;
- Resource efficiency through auto-scaling (up- or downscaling);
- Simplified deployment and operation;
- A better time to market due to development speed and simplified testing of FaaS functions.

2.4. Summary of literature review and literature gap

2.4.1. Summary

Currently, the concept of Bimodal IT is only used as an approach to deliver fast-changing customer-centric and business-critical IT systems without an adequately defined IT Architecture. Therefore, the concept of Bimodal IT is being criticized as an end-solution for digital transformation. Some practitioners are arguing that this approach can increase complexity and even destroy the organizational culture. On the other hand, practitioners and CIOs also acknowledge that the management of traditional IT is equally important as Digital IT due to the existing legacy environment(s) and the IT history of established organizations.

The baseline of the various concepts is that it leads to two different IT architectural layers of speed. Practitioners and researchers discuss that, on an architectural level, Bimodal IT is taking advantage of emerging tools and platforms and simultaneously maintains and evolves the traditional stable, mission-critical back-end IT systems.

Also, interaction and integration between the two IT layers are vital for stability and agility. Therefore, the back-end IT systems should be equally fast adjustable and provide the same integration mechanisms as the customer-facing IT systems. To achieve this goal, it is essential to provide multiple APIs that can connect the back-end IT systems with the front-end IT systems. Consequently, the IT organization needs to have tooling, procedures, and deployment models in place that can help to release new functionalities as soon as possible. To succeed within the transformation, the various IT teams should work together and find the correct balance between agility, delivery, stability, and operations.

To delight the customer, the IT organization needs to ensure that dependencies are managed and tracked, communication protocols are in place, and high-quality functionalities across Traditional IT and Digital IT is deployed and released in a synchronized and fast way.

2.4.2. Literature gap

The available literature focuses mainly on the organizational aspects of a Bimodal IT transformation. There is also a link to create IT systems more modular through services. However, the literature does not discuss or provide a guideline on how the architecture of an organization should be arranged or shaped to support this transformation.

Therefore, despite the available literature on the concept of Bimodal IT and the need for two separate modes of IT delivery, the literature does not provide any practical insights into how a Bimodal Architecture could be achieved on an enterprise architecture level or be established within an IT landscape. Also, concrete implementation details regarding the set-up and the foundation of a Bimodal IT platform are still lacking within the existing literature. Due to this gap, the research will focus on the design patterns of a Bimodal Architecture and will provide the core components of the platform through a RA. This RA will provide an insight into the layers, provide the essential components within each layer and reveal how the elements are integrated. It will also disclose the integration patterns between the layers. Through the design patterns and the RA, it will be possible for organizations to adopt the patterns of a Bimodal IT environment or to implement and utilize the platform within their IT landscape.

As indicated in the summary section, a Bimodal Architecture consists of three layers. These layers are:

- A Fast-Speed Layer (Mode 2);
- A Slow-Speed Layer (Mode 1);
- And a Communication Layer.

The Communication Layer is marked as the most crucial layer because it integrates and connects the Slow-Speed Architecture with the Fast-Speed Architecture through a service-based Architecture, such as a SOA or a MSA environment. This finding will serve as a base for the continuing of this study.

3. Conceptual model

In this chapter, a conceptual model for the Bimodal Architecture environment is presented.

3.1. Conceptual model of a Bimodal IT platform and related integration patterns

After conducting the literature review, a conceptual model has been developed that will serve as a base for the data collection and findings during the interviews. The conceptual model itself is represented in figure 3-1.

The conceptual model is derived from and developed through the findings within the literature review. The RA that is going to be developed in this study can be based on this conceptual model. In this scenario, the conceptual model can be enhanced with relevant sub-components in each layer, and is it possible to incorporate the development and deployment aspects that are related to a Bimodal IT platform. In addition, the conceptual model is also extended with another model to show the potential integration patterns between the various layers.

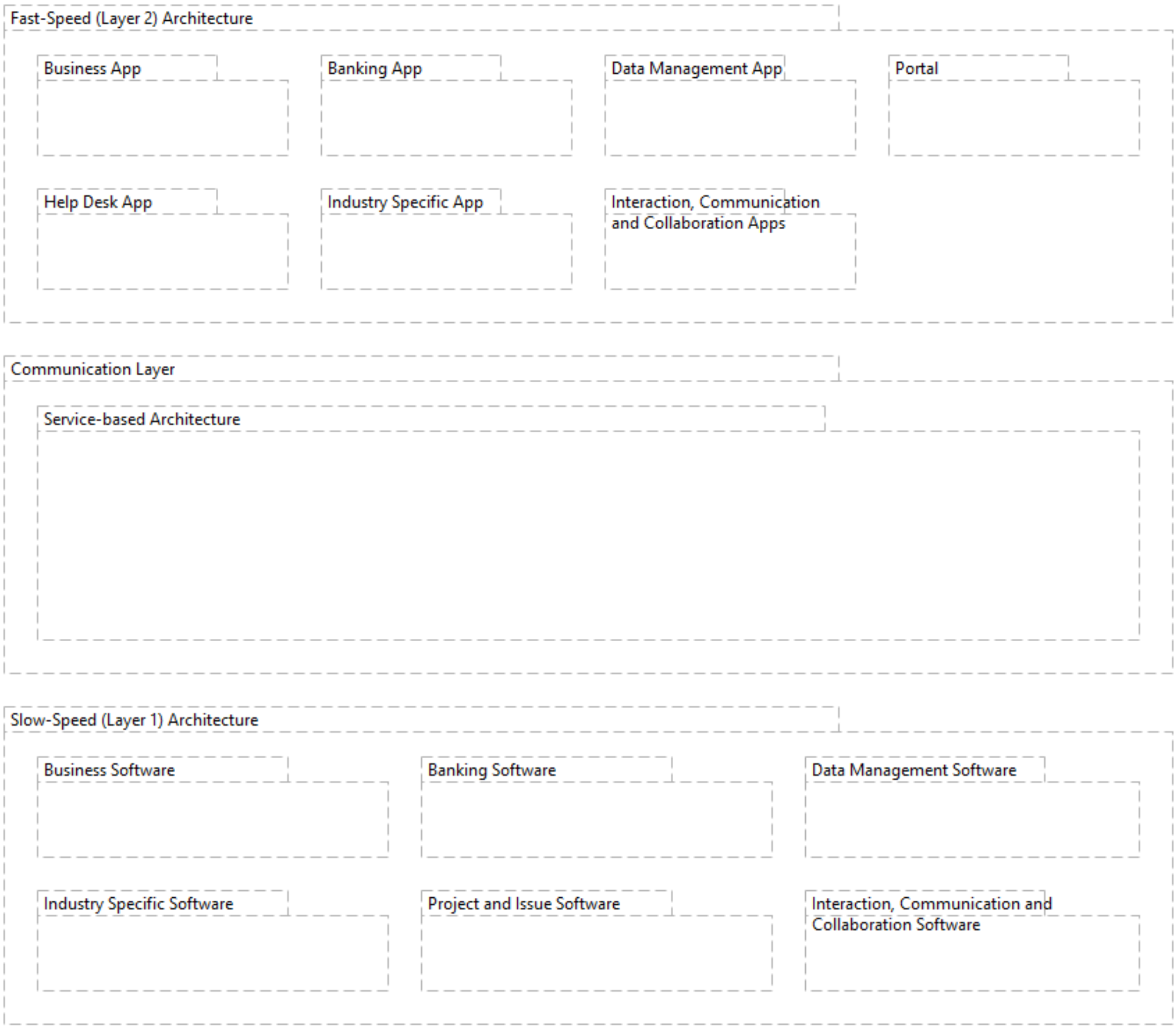

Figure 3-1 Conceptual model – Bimodal IT platform and its layers

As shown in figure 3-1, the conceptual model is composed of three layers. The Slow-Speed Architecture holds or comprises IT systems that are stable and do not frequently change, while the Fast-Speed Architecture embeds IT systems that are very responsive to change. Both layers are connected through a Service-based Architecture, such as a SOA or a MSA environment, which is part of the Communication Layer.

Figure 3-2 presents the interaction and integration patterns between the Slow-Speed and Fast-Speed Architecture through a Communication Layer. This model highlights how the IT systems of the Slow-Speed Architecture are exposing its data, objects, and functionalities towards the service-based environments and how the services are grouped and composed into IT systems within the Fast-speed Architecture to serve the user, customer, supplier, and other relevant stakeholders needs. Also, the model visualizes the phenomenon in which an IT system of the Fast-Speed Architecture gets stabilized or matured and eventually becomes part of the Slow-Speed Architecture.

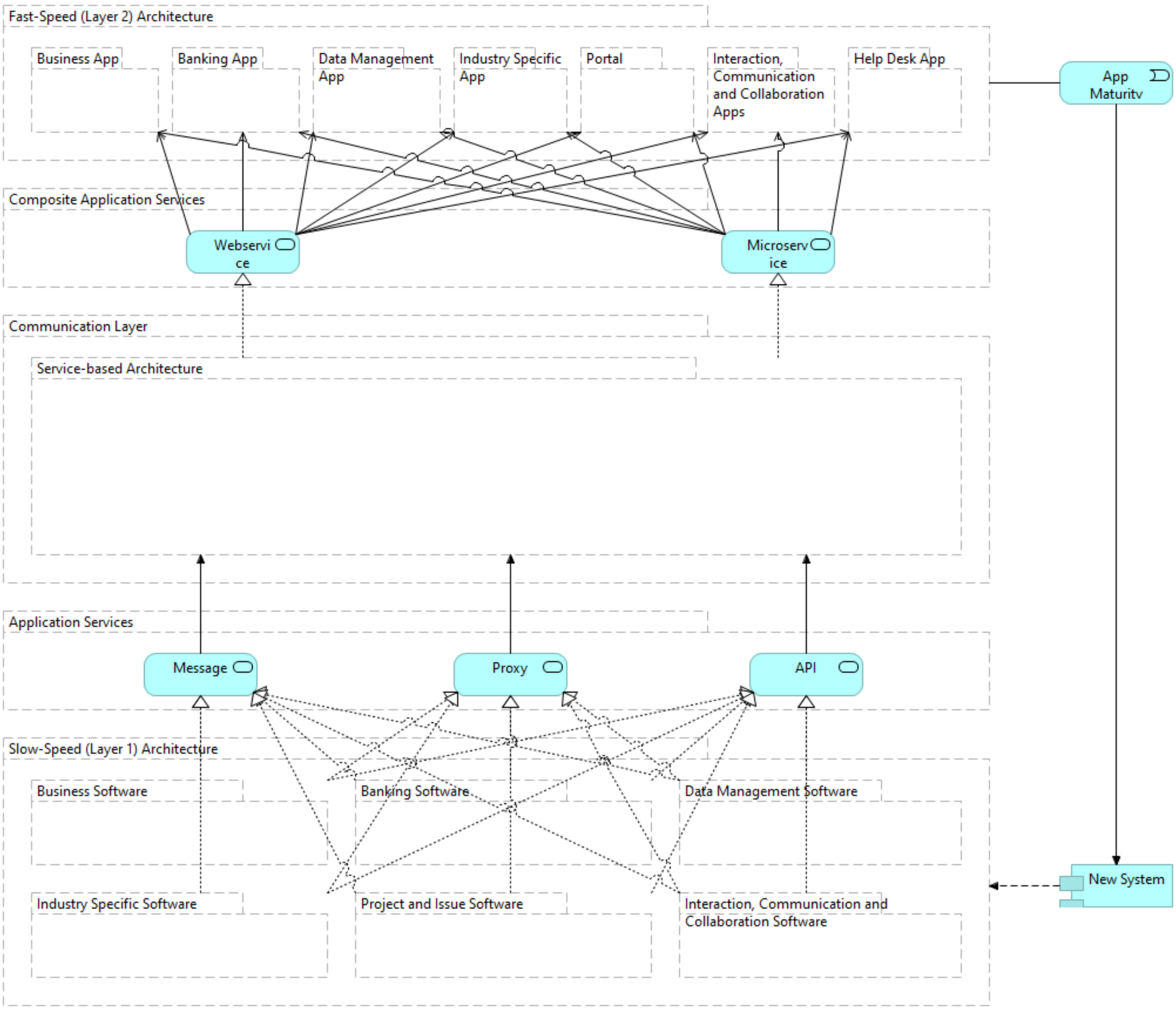

Figure 3-2 Conceptual model – Bimodal IT platform and its associated integration patterns

4. Methodology

In this chapter, the methodology is presented. It explains how the research is conducted and outlines the research design, philosophy, approach, strategy, and data collection method. The methodology is described through the stages of the Research Onion, which is developed by Saunders (Saunders et al., 2015). The Research Onion is represented in figure 4-1.

This chapter also includes the interview process, case selections, candidate selections, and the interview questions associated with this study.

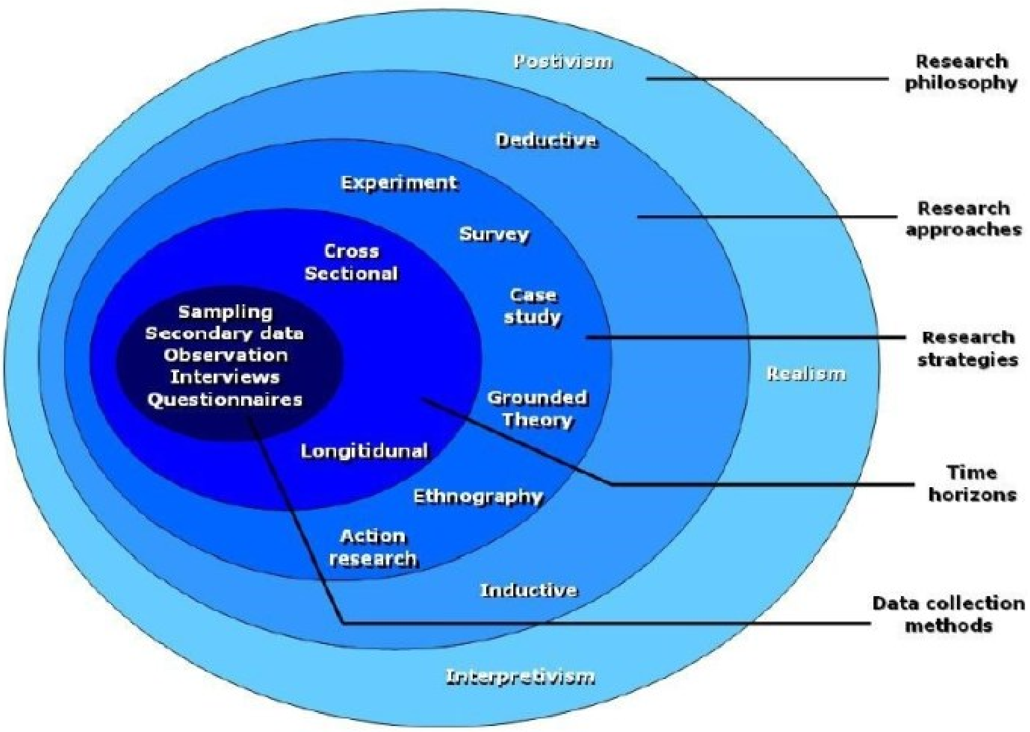

Figure 4-1 The Research Onion (Saunders et al., 2015)

4.1. Research scope

This research aims to identify, describe, and model the core components that are part and associated with the establishment of a Bimodal IT platform. From the literature review, it becomes clear that a Bimodal IT platform can include an extensive number of elements, aspects, patterns, objects, processes, data, methods, tools, resources, roles, systems, services, technical, IT infrastructural, and network components. To provide a clear, understandable, and high-level overview of the fundamental elements of a Bimodal IT platform, only the most important organizational, functional and technical elements will be identified, gathered, and described within this research. Some examples of these components and aspects are:

- The architectural layers of a Bimodal Architecture;

- The incorporated system categories within each layer and the associated IT systems (high-level);
- Characteristics of the different layers. The characteristics can help to indicate and explain why a specific IT system or functionality is embedded into a specific layer;
- Associated integration patterns of each layer;
- The service-based environments and related middleware components, such as an ESB;
- The integration patterns between the layers;
- The most popular or frequently used agile and traditional approaches. Each layer can use different SDLC methodologies to deliver an IT system;
- Tools and components that are related to the traditional and agile SDLC methodologies;
- And the related deployment models and their associated high-level components. Each architecture layer can use multiple and different hosting approaches.

Other patterns, aspects, elements, and components that are related to a Bimodal IT platform will not be described or discussed in this study. Some examples of these elements are:

- Team set-up and skill management;
- Roles and resources;
- Archetypes;
- IT Service Management and Product/Project Portfolio Management;
- Data flow patterns;
- Objects that are embedded within the different IT systems;
- Message definitions and mappings (web services and microservices);
- Security patterns and related aspects;
- Automation patterns, such as workflow, AI, etc.
- And low-level IT infrastructure and network components, such as OS, Memory, Firewalls, DNS, Load Balancers, and other relevant components because these components are required in any IT landscape.

Also, the steps and phases that are associated with the implementation of the identified core components are out of scope. Through this focus, the study can present the design patterns and associated core components that are required for the set-up of the foundation of a Bimodal IT platform without considering the implementation steps.

This research aims to define and disclose the design patterns and to provide a high-level design of a Bimodal IT platform through a RA. To achieve this, the identified core components will be incorporated into the RA. By elaborating on a high-level design, the RA can provide an insight into a Bimodal IT platform on an enterprise architecture level and reveal its core components, related aspects, and how the different elements are related to each other. Consequently, the

RA can be used as a guideline to help organizations structure, prepare, and manage future Bimodal IT initiatives.

Conclusively, this study will deliver the design patterns that are an integral part of a Bimodal Architecture and provide a high-level design through a RA to support the realization and exploitation of a Bimodal IT platform. Ultimately, through the RA, an answer can be given to the question of how a Bimodal Architecture could be implemented and deployed from a practical perspective within an organization and its IT landscape to support their digital transformation.

4.2. Research design

Since the literature review regarding Bimodal Architecture does not provide all the core components, extra information regarding the elements is required to determine the design patterns. This additional information can also help in the development of the RA.

The research design used in this study is a combination of a literature review on the technical, functional, and business aspects and the defined research purpose regarding the foundation and realization of a Bimodal IT platform through a case study. The aim of the study is to contribute to the understanding of how a Bimodal IT platform could be realized in a real-world context.

The focus has been set to reach out to a minimum of five and a maximum of ten organizations to identify, gather, and capture all the essential components. The goal is to identify all the elements that are minimally required to establish the foundation of a Bimodal IT platform.

The individual cases can help to generate in-depth insight into the design patterns regarding the foundation of a Bimodal IT platform. Based on this approach, the holistic multiple-case study is the most suitable design for this research (Yin, 2018).

This research can be quantified as a qualitative research. Qualitative research allows the development of rich and informative conclusions while examining complex problems. It also allows to engage with practitioners through constructive dialogs to create a shared understanding (Stettina & Hörz, 2015). Within the qualitative research, an interpretive approach is used to collect the data and examine the subjective perceptions, meanings, and feelings of the candidates who were, or still are, involved in the transformation towards Bimodal IT.

4.3. Case selection

As mentioned in the previous section, the research focuses on a maximum of ten organizations that already have implemented or adopted the concept of Bimodal IT. Each of the selected organizations is treated as a separate case. The aim of the cases is to identify the core components, which are minimally required to set-up the foundation of a Bimodal IT platform. The case selection focus has been limited between five and ten cases due to the given time frame and the fact that this selection range should be sufficient enough to conduct the study.

The selection criteria of the cases are based on the following requirements:

- The organizations are dealing with complex and multiple IT systems and architectures;
- They are making use of the Bimodal IT approach via one of the six identified archetypes;
- The organizations confirm that they have multiple IT architectural layers in place and are using the different IT environments.

From each case, the core components of the Bimodal IT platform will be gathered and identified. Subsequently, the findings from all the cases will be compared. The expectation and prediction regarding the findings are that the individual case studies provide similar results, also known as literal replication (Yin, 2018). Consequently, through the findings, more could be said about the essential foundation of a Bimodal IT platform. Also, the determined components, as well as via the literature review as through the cases, will be used together to develop the RA.

4.3.1. Case selection process

In general, complex IT environments are found in bigger organizations. Therefore, the case selection scope is set to large organizations with more than 250 employees and the usage of an API platform. Also, to ensure variability, the cases are chosen from different industries.

All organizations that fit the case criteria were taken into consideration to a maximum of ten cases. From this selection, nine organizations were chosen according to the availability of the participants. Table 4-1 presents an overview of the selected cases. Due to privacy reasons and ethical considerations, the organizations and their candidates are anonymized. Each organization is characterized through a company ID, such as C1, C2, etc. And the participants are characterized via candidate 1, candidate 2, etc.

The literature review is used to develop specific interview questions to identify and collect the core components from each case.

4.4. Research philosophy

In this research, the research philosophy interpretivism is used. Interpretivism emphasis that reality is subjective and subjected to change because humans create meanings (Saunders et al., 2015). The motivation for choosing this philosophy is that it uses small samples and provides in-depth insight into the experience of different groups of people. This approach allows to obtain and collect the core components from the different groups who were, or still are, involved within the implementation of a Bimodal IT platform. Furthermore, key elements, aspects, patterns, and used phases and steps during the transformation towards Bimodal IT can be collected via these participants because each group performs different roles, tasks, and activities during the transformation. These experiences and data can be used to complement, explain, and to support the RA.

To determine the main components of a Bimodal IT platform, it is important to collect the experiences on a broad as well as on a detailed level. Therefore, the selected cases are divided into two groups, namely:

- Enterprise/Organizational level;
- Project/Solution implementation/Development level.

The divergence between the Enterprise and the Project/Solution level ensures that the data gathering process meets its goal and guarantees a proper distribution between the participants.

4.5. Research approach

The theory for this study is developed via an inductive approach. A general inductive approach allows the production of reliable and valid findings through a set of procedures that are used for analyzing qualitative data (Thomas, 2006). The reason for using this approach is the ability to generate raw textual data into a summary. The approach also makes it possible to create clear links between the research objective and the summary findings obtained from the literature and interviews. The research will start via the collection of data from the different groups. Once the data is gathered, the theory can be built to answer the research questions (Saunders et al., 2015).

4.6. Research strategy

The design, implementation, and documentation of this research are based on the literature of Yin (Yin, 2018). Yin distinguishes four types of research designs for the conduction of case studies. These research designs are based on the characteristics of contextual conditions and the number of units of analysis. From these four research designs, the holistic multiple-case study is the most suitable design for this research (Yin, 2018)

The holistic multiple-case study design allows to handle each organization as a case, although each of the candidates is part of a different unit within the organization. Through the nine cases, it is possible to examine the design patterns of a Bimodal IT platform, such as the components that are required for the set-up of the foundation of a Bimodal IT platform and the interrelationship between these components.

4.7. Data collection

The data collection method is based on semi-structured interviews. This type of interview can help to anticipate on the answers and gather more information from its candidates. In addition, the goal is also to interview at least one candidate on enterprise-level and one candidate on the project/solution implementation/development level within each organization. However, it will not be possible to interview both roles within a particular organization in some cases. The preference goes to a candidate on the project/solution implementation/development level in these cases.

The aim of the sampling is based on a minimum of 15 and a maximum of 40 interviews. This criterion will help to gather and cover all the relevant experiences and to identify the elements of a Bimodal IT platform.

The interview covers four main topics:

- Understanding Bimodal IT;
- Determining the different architectural layers of a Bimodal Architecture and outline the associated components and integration patterns;
- Identifying the related SDLC processes and tools;
- Understanding and identifying the associated deployment models.

The interview has a fixed number of questions.

4.8. Sample selection

The sampling method in this research is based on non-probability sampling. Through the use of purposive and expert sampling, it is possible to identify and gather the components within a Bimodal Architecture environment. Purposive sampling is a non-random technique that is based on the judgment of the researcher. It is one of the most effective sampling methods in the case of limited resources and time (Etikan, Musa, & Alkassim, 2016). Expert sampling is essentially a subcategory of purposive sampling.

The objective of the interview is to gather the experiences of the candidates who were, or still are, involved in the transformation towards Bimodal IT. The participants that are selected for the interviews need to be a member who was actively involved or has active involvement within the planning, execution, or support of a Bimodal Architecture environment. This focus will ensure that the core components, which are part of the design patterns of a Bimodal Architecture, can be identified.

From each group, the potential participants could be:

- Enterprise/Organizational level:
    - CIO;
    - CDO;
    - Chief Technology Officer (CTO);
    - Enterprise Architect;
    - Information Manager;
- Project/solution implementation/development level:
    - Integration Architect;
    - Solution Architect;
    - System Integrator;
    - Innovation Lead;
    - Integration and Technical Lead;
    - Functional Consultant (needs to have experience with an API platform);
    - Technical Consultant (needs to have experience with an API platform);
    - Developer (APIs, web services, or microservices).

4.9. Candidate selection

The selection of the candidates is based on a number of criteria. These criteria are:

- The position and role of the candidate. The candidate works for an organization that complies with the case group criteria;
- The number of years of work experience, with a minimum of three years of experience;
- The candidate needs to be familiar with the different architecture layers;
- The candidate needs to be able to mention the components of each layer to a certain extent;
- And the candidate is familiar with the used SDLC methodologies and deployment models within their IT landscape.

4.9.1. Candidate selection process

Multiple organizations were screened through the researchers' network. Only the organizations that fit the case criteria were taken into consideration. Once the case focus of ten organizations was reached, the researcher started to screen the candidates to see whether they fit the selection criteria. The candidates that fit the criteria were asked via phone or mail if they would like to participate in this research.

4.10. Interview process

The interview process includes the set-up of the interview themes, conditions, and questions. The interview format comprises an introduction section explaining the purpose and the problem statement of the research. The format also includes the aim of the interview and comprises the conditions of the interview itself, such as the duration, confidentiality, the permission to record the interview, and the aim and usage of the gathered data.

4.10.1. Interview execution

The interview process started when a candidate agreed to participate in the study.

The first step was to plan and send over the appointment details in an email together with the information on the subject. The information sent included the introduction, the problem statement, and the aim of the interview. In some scenarios, the questions were sent beforehand.

One hour before the interview itself, some precautions were taken to meet the conditions as stated in the interview format. Once the interview started, a short introduction was given. During the introduction, the following elements were discussed:

- The purpose of the research;
- Terms and conditions of the research;
- Structure of the interview;
- Duration and confidentiality;
- Permission to record the interview.

After this step, the interview started with the candidate.

In total, 15 interviews were conducted in this study. The related cases and candidates are presented in table 4-1.

Company ID	Industry	Participant	Position/Role in the company	Role ID	Years of experience
C1	Consulting	Candidate 1	Developer	DEV-01	4 years
C1	Consulting	Candidate 2	Enterprise Architect	EA-01	15 years
C2	Utilities	Candidate 3	Integration & Technical Lead	ITL-01	12 years
C2	Utilities	Candidate 4	Enterprise Architect	EA-02	20 years
C3	Banking	Candidate 5	Solution Architect	SA-01	11 years
C4	Utilities	Candidate 6	Innovation Lead	INL-01	13 years
C4	Utilities	Candidate 7	Enterprise Architect	EA-03	42 years

C5	Consumer Goods	Candidate 8	Integration & Technical Lead	ITL-02	15 years
C5	Consumer Goods	Candidate 9	Enterprise Architect	EA-04	20 years
C6	Postal	Candidate 10	Project Manager	PM-01	8 years
C6	Postal	Candidate 11	Enterprise Architect	EA-05	23 years
C7	Products	Candidate 12	Integration Architect	IA-01	17 years
C8	Municipalities	Candidate 13	Data Warehouse Developer	DEV-02	12 years
C9	Hosting	Candidate 14	Integration & Technical Lead	ITL-03	10 years
C9	Hosting	Candidate 15	Enterprise Architect	EA-06	22 years

Table 4-1 Cases and related candidates

4.10.2. Interview questions

The main goal of the interview is to gather the core components of a Bimodal IT platform. However, to guarantee a logical flow and capture the candidates' full experience, some additional questions are included regarding the processes and impact of a Bimodal IT environment.

The interview questions are grouped together through a set of topics. The topics and their associated aim are shown in table 4-2.

Topic	Aim
General questions	Brief introduction of the candidate, company, and the IT organization.
General Bimodal questions/Understanding of Bimodal	Understand the Bimodal Architecture environment and identify the associated components and patterns.
Architecture layers (Slow-Speed and Fast-Speed domains) questions	Understand the different IT layers and gather which components are embedded within each layer.
Methodology (development process) questions	Understand which SDLC methodologies are used within each layer to deliver the IT systems.
Questions regarding tooling and components for software delivery within a Bimodal Architecture environment	Understand which tools and components are used for the development and delivery of the IT systems.
Deployment/Hosting model related questions	Understand and determine the deployment models behind each layer.

IT landscape and roadmap questions	Get a better overview of the organizational IT landscape, the IT strategy, and the roadmap.
Question regarding the future of Bimodal IT	Get insights into the future of Bimodal Architecture.
Bimodal IT implication questions	Get a better understanding of the processes, teams set-up, budgeting, and technical debt of a Bimodal Architecture environment.

Table 4-2 Interview topics and related aim

A number of interview questions have been prepared for each topic to get an answer on the GQ's. Each interview comprises the same topics, groups, and questions so that the interview is consistent and the gathered data can be analyzed and compared. The interview is semi-structured and contains only open questions. This approach allows to collect as much information as possible from the participants. The semi-structured approach also enables the researcher to ask additional questions to capture relevant and required data. The interview questions are stated in table 4-3.

Number	Questions
General questions:	
1	Personal information: years of experience, and current function/position/role within the company or project.
2	General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.
General Bimodal questions/Understanding of Bimodal:	
1	What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?
2	What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?
Presentation of the conceptual models	
3	Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?
Architecture layers (Slow-Speed and Fast-Speed domains) questions:	
Explanation Slow-Speed and Fast-Speed domain:	

IT systems within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the IT systems are subjected to rapid and frequent changes to meet customer, user, or supplier demands.	
Slow-Speed and Fast-Speed layer questions:	
4	Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?
5	How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?
Explanation Communication layer: *The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.*	
Communication layer question:	
6	In case of a communication/integration layer, such as a service-based architecture (for example, a SOA or MSA), what are the essential components or features of this layer? And why?
Methodology (development process) questions:	
7	Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?
8	What is the impact of a Bimodal Architecture environment on your software delivery/methodology?
9	Do you use different methodologies across the different layers?
10a	In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?
10b	Which layers are affected by CI/CD?
10c	What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?
11	In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?
Questions regarding tooling and components for software delivery within a Bimodal Architecture environment:	
12	Which other tools or components are used to realize and facilitate the Bimodal IT environment?
13	How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?
Deployment/Hosting model related questions:	
14a	Is there a relationship between the deployment/hosting model and Bimodal Architecture?

14b	If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?
IT landscape and roadmap questions:	
15	What were the major milestones that lead to the current set-up of the IT landscape?
16	What is the future roadmap of the IT landscape of your organization?
Question regarding the future of Bimodal IT:	
17	How do you see the future of Bimodal Architecture/Two-Speed IT?
Bimodal IT implication questions:	
18a	Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?
18b	How is the priority setting, budgeting, and contracting done in case of Bimodal IT?
19	Can you elaborate on the technical debt?

Table 4-3 Interview questions

Interview questions regarding Bimodal Architecture implications

The block "Bimodal IT implication questions" is classified as excessive in this research and is only asked when there is sufficient time left during an interview. The questions are taken into account in the interview because the questions can provide a better view on the implications of a Bimodal IT platform and how it influences an organization on a management and project level. As mentioned within the literature review, Bimodal IT impacts the processes, team set-up, costs, outsourcing decisions, etc. Therefore, it can be interesting to get more information on these topics since it can help to improve the RA within future versions.

However, as indicated, the questions are irrelevant for this research because the study will only focus on the core components of a Bimodal IT platform and how these components can be used to set-up, realize and facilitate a Bimodal Architecture environment.

4.11. Data analysis

This study uses qualitative content analysis as a method for analyzing the data retrieved from the interviews. Content analysis is used to evaluate patterns within the text data. It allows the identification of themes, patterns, and subjective interpretations of the content within the text data through a systematic classification process of coding (Hsieh & Shannon, 2005). The purpose of the analysis is to provide knowledge, insights, and representations of facts (Elo & Kyngäs, 2008).

Through the conventional content analysis, codes can be derived from the data. In contrast, the summative content analysis allows the identification and quantification of certain words and data content with the purpose to understand the underlying context(s).

4.11.1. Interview processing

To analyze the data, all the interviews are transcribed and coded. The transcripts are first categorized through the use of conventional content analysis. Each interview is coded by assigning unique keys (#-tag) to themes, key aspects, statements that are relevant to the GQ's, and components of the Bimodal IT platform. An example is the key "#Communication_layer_ESB_components." Subsequently, the data is combined in a codebook. Each key or a combination of keys refers to a tab within the codebook. For example, the tab "ESB components analysis" correlates with the key "#Communication_layer_ESB_components" from the text data.

Also, general comments and notes are classified because some responses do not provide a clear answer on a specific question. These statements lead to certain interpretations. An example of such a response has been given on question 14a by one of the candidates. The candidate responded, "Not all systems and software can be moved to the cloud due to their importance. However, we try to keep this at a minimum." This particular statement indicates that not all IT systems embedded within an IT landscape can or will move towards the cloud. Even if the determined IT strategy is solely focused on the cloud. These statements are interpreted in a general way and taken into consideration if they are indicated by four candidates or more.

The conventional content analysis enables the researcher to combine and compare the different responses to a particular question. It allows to group the themes, aspects, and patterns. Also, by combining the data, the gaps can be filled within the interviews. This reduces the effort in gathering extra information that was missed during another interview.

The second step is the execution of a summative content analysis within each tab. Through this approach, it is possible to identify and quantify certain words and data content. An example is the code "Bimodal IT comprises three layers," or the code "Biztalk - Integration Engine (comprising an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)." This process allows the researcher to capture the key aspects, design patterns, and components of a Bimodal IT platform. Any key aspect, design pattern, or component mentioned by four candidates or more is taken into account and discussed in this study.

The criteria regarding the referral of elements by four candidates or more do not apply to the obtained IT systems (captured through question 4 in the interview) and their related integration patterns (mainly derived via question 2 in the interview). The reason is that all the indicated IT systems and integration mechanisms by the candidates are already an integral part of the IT landscape. Since an extensive number of IT systems and integration mechanisms are provided, only a select number of components will be used in this study for the development of the RA. The selection itself will be based on the researchers' experience. The selected IT system components will be used to group, allocate, and model the various IT systems within the layers of the RA. Consequently, the associated integration mechanism components, which will be referred to as "middleware components" in this study, will be used to model and show how the different IT systems and their corresponding system domains are connected with each other within the Slow-Speed and Fast-Speed layers. Also, the middleware components will be used to reveal how the capabilities and functionalities of the IT systems embedded within the Slow-Speed Architecture are exposed as services.

Through the use of qualitative content analysis as a data analysis method, it is possible to identify and capture the design patterns and core components of a Bimodal IT platform. This approach also ensures traceability from the processed results to the original interview transcripts. Furthermore, the approach will help to increase the credibility and trustworthiness of the RA.

5. Results and findings

This chapter describes and presents the results and findings collected from the in-depth interviews. The first section discusses the conducted data analysis. It describes how the results are established throughout the codebook and discloses the correlation between the interview transcripts and the coding process. The second section outlines the results retrieved from the in-depth interviews themselves. And the third section presents the findings and the related design patterns and core components of a Bimodal IT platform.

5.1. Qualitative content analysis result

As discussed in chapter 4.11.1, all the interviews are transcribed and coded. The responses on each question are labeled with one or more keys (#-tag). Through this approach, every response is grouped into specific themes. Subsequently, the key aspects, patterns, and components are classified within the text data.

5.1.1. Conventional content analysis result

During the conventional content analysis, 96 unique keys are created to identify the themes and to group the text data. The keys are also used to create the different tabs within the codebook. In total, 29 tabs are created throughout the codebook. Since the keys are the link between the codebook and the text data, they can be used to trace the processed results back to the original interview transcripts. The keys themselves are presented in appendix 1, while the correlation between the keys and the tabs is presented in appendix 2.

5.1.2. Summative analysis result

Each tab within the codebook contains a subset of the aspects, patterns, and components of a Bimodal IT platform. For each aspect, pattern, or component, a code has been defined. Subsequently, the codes are categorized between high-level and sub-level codes. In total, 108 high-level codes and 149 subcodes are created. The codes are presented in appendix 3. After the identification of the codes, the codes are grouped into topics and quantified by counting how often the codes are mentioned within the text data. The results from the data analysis are discussed in chapter 5.2. The correlations between the tabs, topics, and codes are presented in appendix 4.

5.2. Data analysis result

For the data analysis applies that the related codes (aspects, patterns, and components) are mentioned in different tabs within the codebook. Therefore, the tabs, topics, and associated codes are combined, grouped, and divided into five clusters. The clusters are shown in figure 5-1 and discussed in separate subparagraphs. This approach ensures that the collected content, conduced analysis, and context is clear, understandable, and readable.

The correlation between the tabs, topics, and clusters is presented in appendix 5.

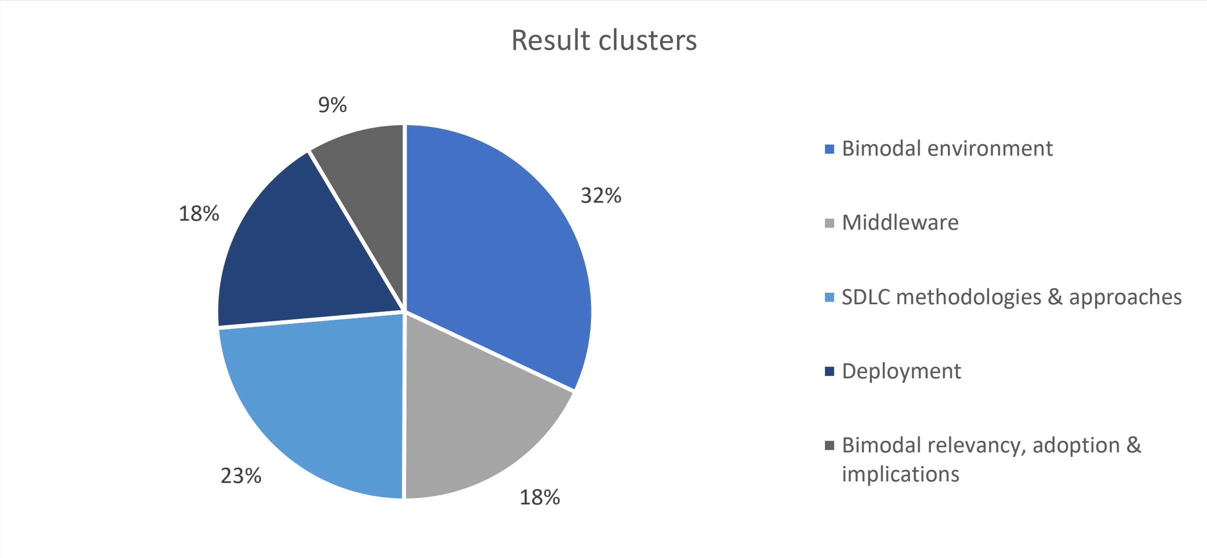

Figure 5-1 Result clusters based on the data analysis

The over-all count of the 108 high-level codes resulted in 747 responses. By plotting the responses over the five result clusters discloses that 239 responses can be allocated to the cluster “Bimodal environment,” 135 responses to the cluster “Middleware,” 176 responses to the cluster “SDLC methodologies and approaches,” 133 responses to the cluster “Deployment,” and 64 responses to the cluster “Bimodal relevancy, adoption, and implications.” The distribution of the responses per cluster is also included in figure 5-1.

5.2.1. Bimodal environment

Figure 5-2 presents an overview of the topics regarding a Bimodal environment. The figure also discloses the distribution of the topics within the cluster.

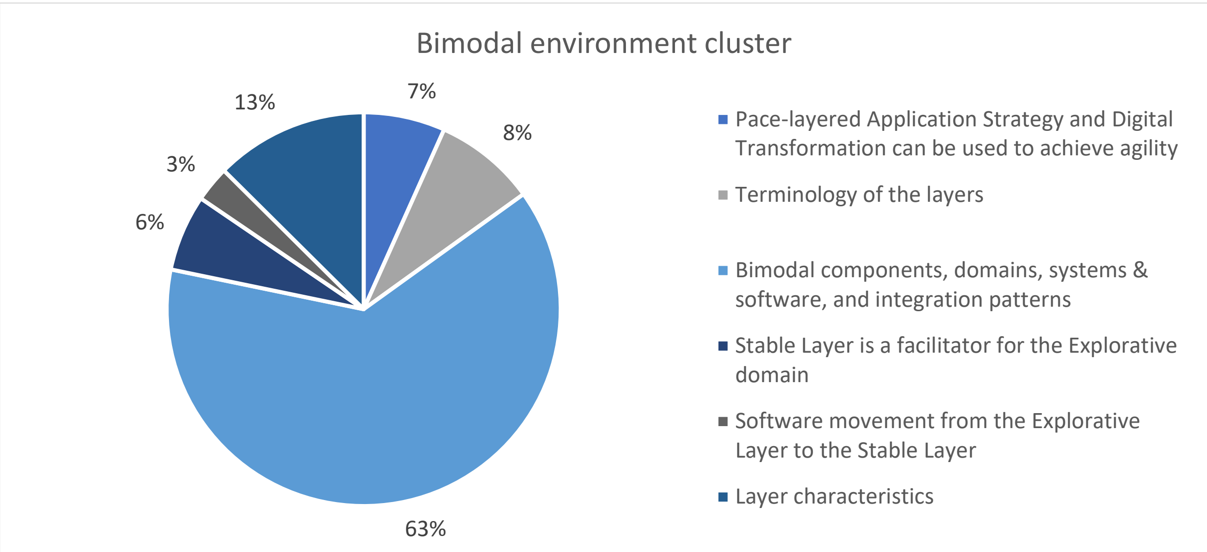

Figure 5-2 Bimodal environment cluster with related topics

Figure 5-3 complements the data analysis regarding a Bimodal Architecture environment and presents the codes from the data to describe the layers, integration patterns, characteristics, and the related IT systems embedded in a Bimodal IT platform.

Pace-layered Application Strategy and Digital Transformation can be used to achieve agility

The first topic focuses on alternative or additional concepts next to the concept of Bimodal IT. In total, 16 responses are provided regarding two concepts that also can help to achieve agility within an IT landscape of an organization in comparison to Bimodal IT.

As indicated, the study focuses on the Bimodal IT concept from Gartner. However, from the data, it is observable that the concepts “Pace-layered Application Strategy” and “Five Building Blocks of Digital Transformation” are indicated by nine candidates as two additional concepts to achieve agility and to support and manage the IT landscape(s). In addition, the code “Uses typologies of other concepts to describe systems” is mentioned seven times and indicates that the typologies from the Pace-layered Application Strategy are a popular way to describe the different kinds of IT systems and to allocate the IT systems into the different layers.

The Pace-Layered Application Strategy itself is a methodology to categorize, select, manage and govern IT systems to support business change, differentiation, and innovation (Gartner, 2012a). In contrast, the Five Building Blocks of Digital Transformation is used to drive digital transformation through an operational backbone, a digital platform, and an external developer platform. The concept shares insights about the customers’ value and provides an accountability framework to coordinate the efforts of autonomous teams (Ross et al., 2018).

Terminology of the layers

The second topic discusses the terminology of the layers within a Bimodal IT platform.

The code “Deviated terminology for Slow-Speed and Fast-Speed” is mentioned 14 times and indicates that the used typologies for the layers need to change. The reason is that the used terminologies do not justify the layers and can be misleading or even cause confusion within an organization. To support the adjustment of the terminologies, candidate 8 indicates that “the problem with the terminologies is that it implicates that some systems are slow while other systems are fast. That is not true. As indicated in the previous question, SAP can be described as stable, secure, and robust. However, SAP also contains modern and user-facing components, such as Fiori applications and OData services.”

As per the data analysis, the typologies “Stable” and “Explorative” are preferred over “Slow-Speed” and “Fast-Speed.” This does not apply to the terminology of the Communication Layer because the majority of the candidates prefer this terminology over other typologies.

To prevent confusion and to align the typologies of the layers with the text data, the terminologies “Stable Layer,” “Explorative Layer,” and “Communication Layer” will be used throughout the research. The Stable Layer will refer to “Mode 1” of the Bimodal IT concept, while the Explorative Layer will indicate “Mode 2”. The update on the terminologies will also help to explain and clarify the layers within the RA.

Bimodal components, domains, IT systems, and integration patterns

Figure 5-2 illustrates that the main topic of the cluster relates to the components, layers, groups of IT systems, and integration patterns of a Bimodal IT environment. From the 239 responses, 151 responses are dedicated to describe the main components, system domains, embedded IT systems, integration mechanisms, and integration patterns of a Bimodal IT platform.

First of all, the data determines that the platform consists of three layers as per the literature.

Secondly, the code “Stable Layer comprises Stabilized digital systems & software” is mentioned by all the candidates and discusses that some of the APIs, web applications, IT systems used for digitalization, and new technologies and solutions, such as sensors, AI, and other Systems of Innovation are embedded within the Stable Layer instead of the Explorative Layer. These IT systems, which will be referred to as “Operational value-adding and Stabilized Systems of Innovation” in this study, are responsible for the creation of operational value or support a specific operational goal. Therefore, the Operational value-adding and Stabilized Systems of Innovation deviate themselves from the IT systems embedded in the Explorative Layer. In addition, the Operational value-adding and Stabilized Systems of Innovation do not necessarily expose any services or functionalities towards other IT systems and domains.

For the integration applies that the connectivity and the interchangeability of the objects and data within and between the layers are mainly established through services. These services refer to web services or microservices.

The data also shows that other integration mechanisms and patterns are used to establish the connectivity and integration within the Stable Layer. The integration can be set-up through the use of Java Message Service (JMS), Enterprise Application Integration (EAI), EDIFACT standards, proxies, Remote Function Calls (RFCs), OData, an ESB, and other middleware components. However, this is not the case for the connectivity between the Stable and Explorative domains since this is only established through a service-based environment.

The fourth observation is related to the code “Discusses a separate internal and external API platform” and highlights the need for a separate internal and external API platform within an IT landscape. This statement is indicated by five candidates.

The codes “Discusses specific systems & software embedded in the Stable Layer,” “Discusses Stabilized digital systems & software,” and “Discusses specific systems & software embedded in the Explorative Layer” are mentioned by all the candidates and are used to define and allocate the IT systems within the different environments of a Bimodal IT platform.

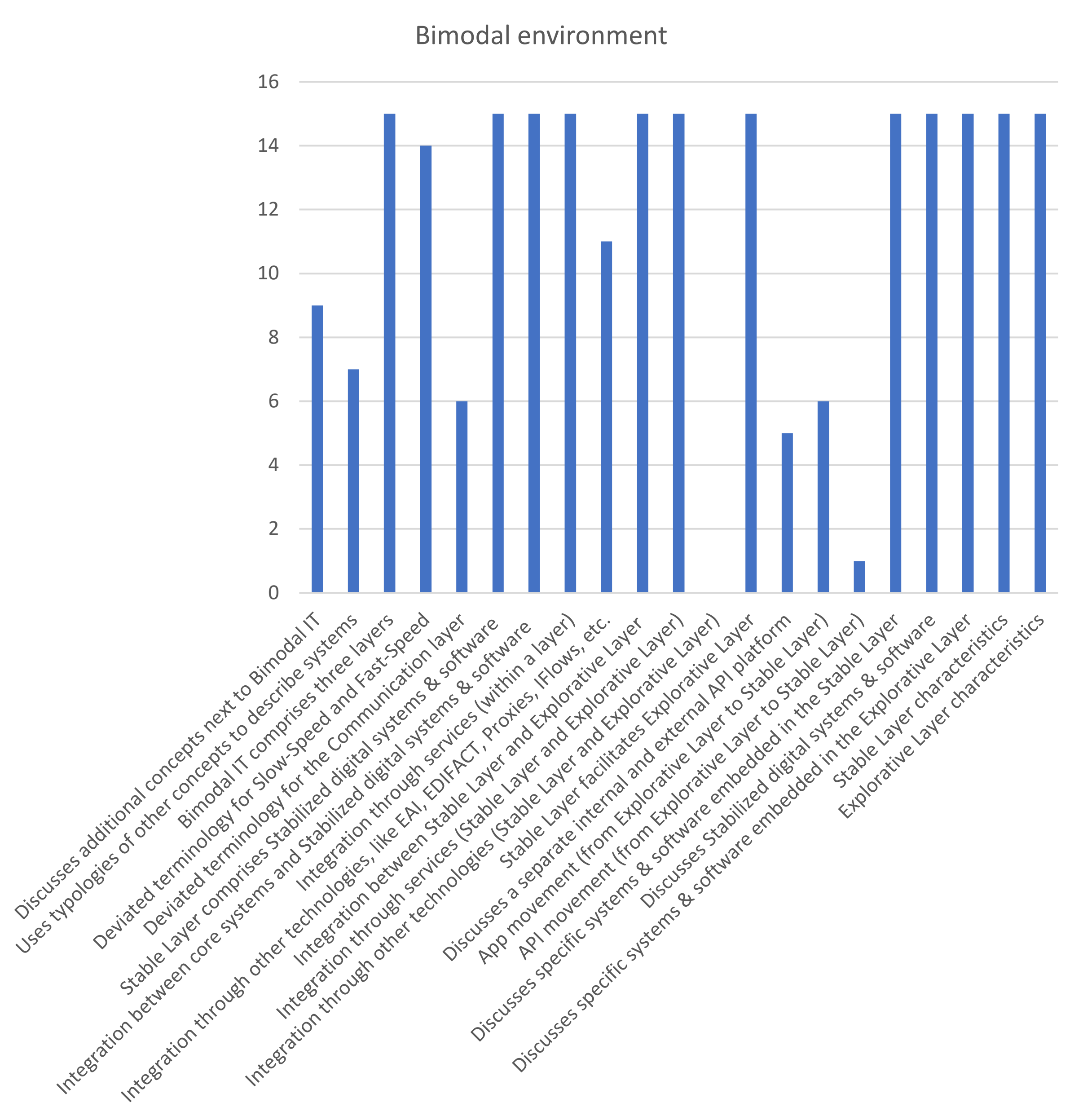

Figure 5-3 Result overview on the Bimodal environment

Stable Layer is a facilitator for the Explorative domain

Regarding the realization and utilization of the Explorative environment, the data discloses that the Explorative Layer cannot be realized and facilitated without the Stable Layer. All 15 candidates underline this statement.

Software movement from the Explorative Layer to the Stable Layer

The fifth topic discusses the movement of IT systems from the Explorative Layer to the Stable Layer. The data shows that seven candidates indicate that a web app, IT system used for digitalization, solution, technology, or API can move from the Explorative Layer towards the Stable Layer when it reaches a certain purpose.

The movement indicates that a web app, IT system used for digitalization, solution, technology, or API has been stabilized over time. There is no need for a rapid development or a CI/CD development process anymore. This also means that the web app, IT system used for digitalization, solution, technology, or API has started generating operational value through a simplified or even a more efficient process. Based on this clarification, candidate 15 even discusses that this movement is only applicable for the APIs. Candidate 15 explains that a web application or IT system invoking a particular API can remain in the Explorative Layer. The reason is that the solution (web app or IT system used for digitalization) is created for a certain purpose. Once the purpose changes or becomes obsolete, the business may still want to be able to change the specific solution in an agile way due to changed requirements.

Layer characteristics

The final topic within the cluster presents the characteristics of the layers. The codes "Stable Layer characteristics" and "Explorative Layer characteristics" are mentioned by all the candidates and are used to define and describe the characteristics of each layer. The characteristics help to separate the layers.

The characteristics of the layers are stated in table 5-1.

Characteristics Stable Layer	**Characteristics Explorative Layer**
Stable	Less stable
Reliable	Flexible
Robust	Fast-changing
Secure	Supports and stores minimal or no business relevant objects and data
Predictable	Errors are already factored in
Supports and stores business relevant objects and data	Only an agile way of development is supported
It does not change a lot once deployed in production	Solutions are centered around interaction and experience
Comprises fewer errors	New technology-oriented
Traditional and agile way of software delivery	Trend to have only a development and production environment
Characterized by long release cycles	Delivers business value

Solid test processes	Rapid delivery oriented. Often and frequent deployments
Problem-solution design	24/7 availability. It can be achieved through an off-line capability
Focus on standardization	Requirements and planning changes continuously
Delivers operational value	Features are small and independent
Implementation costs are high	Focus on innovation, experimentation, and exploration
Use of ITIL processes	Compromises modern systems and applications (cloud-based)
High availability and fault tolerance	Continuous integration/deployment in place to increase time-to-market
The Stable domain is a facilitator for the Explorative domain	
Highly influenced by third parties	
Core systems are not easily replaceable due to their importance	

Table 5-1 Characteristics Stable layer versus Explorative layer (details of analysis can be found in appendix 4 on page 190)

5.2.2. Middleware

Figure 5-4 presents an overview of the topics regarding the service-based environments, which are part of the Communication Layer. The figure also shows the distribution of the topics.

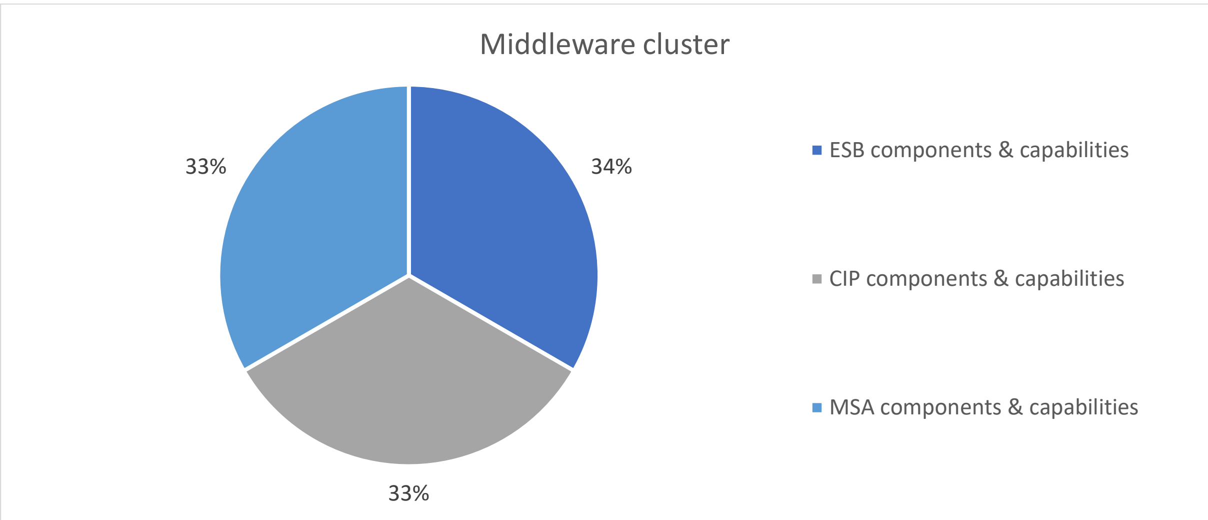

Figure 5-4 Middleware cluster with related topics

Figure 5-5, on the other hand, complements the data analysis regarding the middleware components. It presents the codes to define and describe the mechanisms and components that are embedded within a Communication Layer.

As indicated in chapter 2.4.2, the Communication Layer is the most fundamental layer within a Bimodal IT platform because it enables and ensures the integration, connectivity, and interoperability between the various layers. The layer can comprise a SOA, a Cloud Integration Platform (CIP), a MSA, or even all three service-based environments.

Since each topic gives an insight into a specific service-based environment, the 135 responses are distributed equally within the cluster.

ESB components and capabilities

The first topic discusses the components and capabilities of a SOA environment.

The analyzed text data reveals that all the candidates are already familiar with a SOA environment. The reason is that the majority of the selected cases are using "Microsoft Biztalk" or "SAP PI/PO" as an ESB solution to set-up and to facilitate the connectivity and integration between the IT systems within their IT landscape.

Due to the recognizability, the majority of the candidates are able to indicate all the functionalities, components, and capabilities of an ESB. The code "Mentions a part of the ESB components" determines that only three candidates are mentioning the components partly, while the code "Did not discuss any ESB capabilities" reveals that only two candidates are not able to mention the capabilities of an ESB environment.

The components of the ESB solutions "Microsoft Biztalk" and "SAP PI/PO" are presented in table 5-2. While the ESB capabilities are presented in table 5-3.

ESB components – Biztalk	ESB components – SAP PI/PO
Message Builder (comprising an Adapter Builder, Service Builder, and a Mediation Flow Builder)	Integration Builder (comprising an Enterprise Service Repository and an Integration Directory)
Integration Engine (comprising an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)	Integration Server (comprising an Adapter Engine, Integration Engine, and a Business Process Engine)
Integration Controller/Integration Gateway (comprising a Configuration Manager, Web	System Landscape (comprising a System Landscape/Tenant Manager)

Service Manager, Access Controller, Exception Handler, Event Handler, Mediation Flow Manager, System Landscape/Tenant Manager, Auditor, and a Logger)	
Service Repository	Configuration and Monitoring (comprising a Configuration Manager, Web Service Manager, Mediation Flow Manager, Event Handler, Adapter Manager, Access Controller, Exception Handler, Logger, and an Auditor)
Business Activity Monitoring	Business Activity Monitoring
Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL)	Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL)

Table 5-2 ESB components (details of analysis can be found in appendix 4 on page 192 - Biztalk versus SAP PI/PO)

ESB capabilities
Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Message Tracking, Throttling, and Exception Management)
Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)
Security (provides Authentication, Authorization, Encryption, Certification, and Message Security)
Transportation (allows HTTP, HTTPS, SOAP, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)

Table 5-3 ESB capabilities (details of analysis can be found in appendix 4 on page 193)

CIP components and capabilities

The second topic discusses the components and capabilities of a CIP environment.

Due to the movement towards the cloud, all the organizations with an on-premise environment are obligated to use a CIP to set-up the connectivity between their on-premise software and their cloud solutions.

The codes “Discusses all CIP components” and “Mentions all CIP capabilities” reveal that most of the candidates are able to indicate all the components and capabilities of a CIP environment. Only three candidates are not able to mention the components fully. Also, through the code "Did not discuss any CIP capabilities," it becomes clear that two candidates did not mention any capabilities of a CIP environment.

The analyzed data also discloses that "Azure Integration Services" and "SAP Cloud Platform Integration" are two famous CIP solutions to achieve and establish the connectivity and integration between on-premise and cloud software.

The components of the CIP solutions "Azure Integration Services" and "SAP Cloud Platform Integration" are presented in table 5-4. In contrast, the capabilities are presented in table 5-5.

CIP components – Azure Integration Services	CIP components – SAP Cloud Platform Integration
Cloud Integration Service - Message and Event Hub (comprises a Service Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Message Queues, Exception handler, Logger, and an Auditor)	Cloud Integration Service - Message and Event Hub (comprises an Integration Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Enterprise Service Repository, Message Queues, Exception handler, Logger, and an Auditor)
Cloud Integration Service - Connectivity Hub (comprises a Repository for Adapter Discovery, Adapter Builder, Adapter Configuration Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor)	Cloud Integration Service - Connectivity Hub (comprises a Repository for Adapter Discovery, Adapter Builder, Adapter Configuration Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor)
Cloud Integration Service - Integration Engine (comprises an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)	Cloud Integration Service - Integration Engine (comprises an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)
API Management Service (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager)	API Management Service (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager)
BPM Service (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)	BPM Service (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)
Gateway Service (comprises a Traffic Manager, Exception Handler, Policy	Gateway Service (comprises a Traffic Manager, Exception Handler, Policy

Manager, Version Manager, Call Handler, Logger, and an Auditor)	Manager, Version Manager, Call Handler, Logger, and an Auditor)

Table 5-4 CIP components (details of analysis can be found in appendix 4 on page 193 - Azure Integration Services versus SAP Cloud Platform Integration)

CIP capabilities
Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Service Discovery, Call handling, Message Tracking, Throttling, Rate Limits, Traffic Management, Exception Management, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts)
Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)
Security (provides Authentication, Authorization, Encryption, Certification, and Message Security)
Transportation (allows HTTP, HTTPS, SOAP, OData, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)

Table 5-5 CIP capabilities (details of analysis can be found in appendix 4 on page 194)

Since most CIP solutions are built upon or integrate parts of existing ESB solutions, it is evident that the ESB and CIP environments share the same components and capabilities in the core.

MSA components and capabilities

The third and last topic of this cluster focuses on the components and capabilities of a MSA environment.

As described in chapter 2.3.4, a MSA is primarily used to realize and facilitate microservices. Because microservices are used as an architecture style for building distributed (cloud) applications that are resilient, scalable, independently deployable, and highly evolvable (Aderaldo et al., 2017), it is very suitable for the realization and exploitation of the IT systems embedded within the Explorative Layer.

The theory and the text data show similar results. The retrieved data emphasizes that microservices is the preferred technical component to facilitate an Explorative Layer. Candidate 15 indicates that the Explorative environment comprises various integration patterns. These patterns include web services and especially microservices to connect and integrate the fast-changing IT systems.

From the analyzed data, it is also observable that the candidates are already familiar with a MSA environment. The code "Did not mention any MSA components" is only mentioned once, which indicates that the majority of the candidates are able to mention the components of a MSA environment.

The same outcome applies to the capabilities. The code "Did not discuss any MSA capabilities" shows that three candidates cannot mention any capabilities, while the code "Mentions a part of the MSA capabilities" indicates that only two candidates are not able to mention the capabilities of a MSA environment fully.

The components of a MSA environment are presented in table 5-6, and the related capabilities are presented in table 5-7.

MSA components
Nodes
Pods (*represents the running containers)*
Scheduler
Controllers
Key-value Database
API Gateway
API Manager
API Server
Resource Manager
Container Orchestration Engine
Storage
Queues
Container Registry
Container Repository (optional)
Network Manager
Infrastructure Security Manager
Provisioning Manager
Logs

Table 5-6 MSA components (details of analysis can be found in appendix 4 on page 194)

MSA capabilities
Container Orchestration Platform (provides Container Orchestration, Container clustering, Image Discovery, Container Security, Network settings, Key Vault, Load Balancing, Scheduling, Rollouts, Rollbacks, Self-Healing, and Workflow Management)
API Management (provides Service Discovery, Messaging, Routing, Call handling, Transformation, Rate Limits, Traffic Management, Throttling, Caching, Circuit Breaker,

Failover, Microservice Security, Configuration Management, Networking/IP, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts)
Container (comprises microservices)

Table 5-7 MSA capabilities (details of analysis can be found in appendix 4 on page 195)

In addition, the code “Uses Azure as a MSA” is mentioned 14 times and indicates that “Microsoft Azure” is a famous platform to realize, manage, and exploit microservices within the IT landscape of the selected cases. The reason behind this is, is that the best practice is to run microservices through the use of container technology. The containers themselves can be managed through a container orchestration platform, such as Azure, which establishes an automated process for orchestration, scheduling, and deployment of individual containers. It can also scale-out a group of containers into an application (Pahl, 2015).

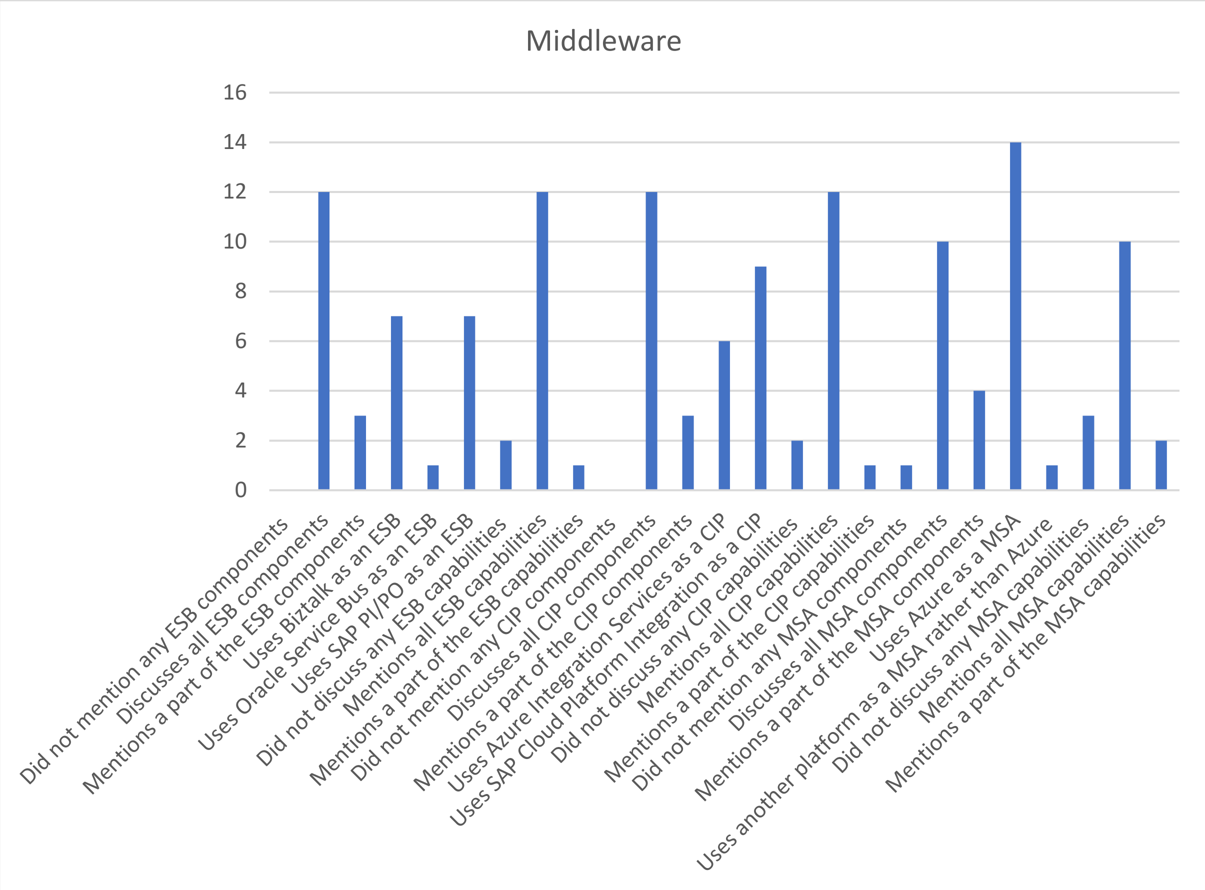

Figure 5-5 Result overview on Middleware

5.2.3. SDLC methodologies and approaches

Figure 5-6 presents an overview of the topics regarding the use of SDLC methodologies and approaches within a Bimodal IT platform. The figure also includes the distribution of each topic within the cluster.

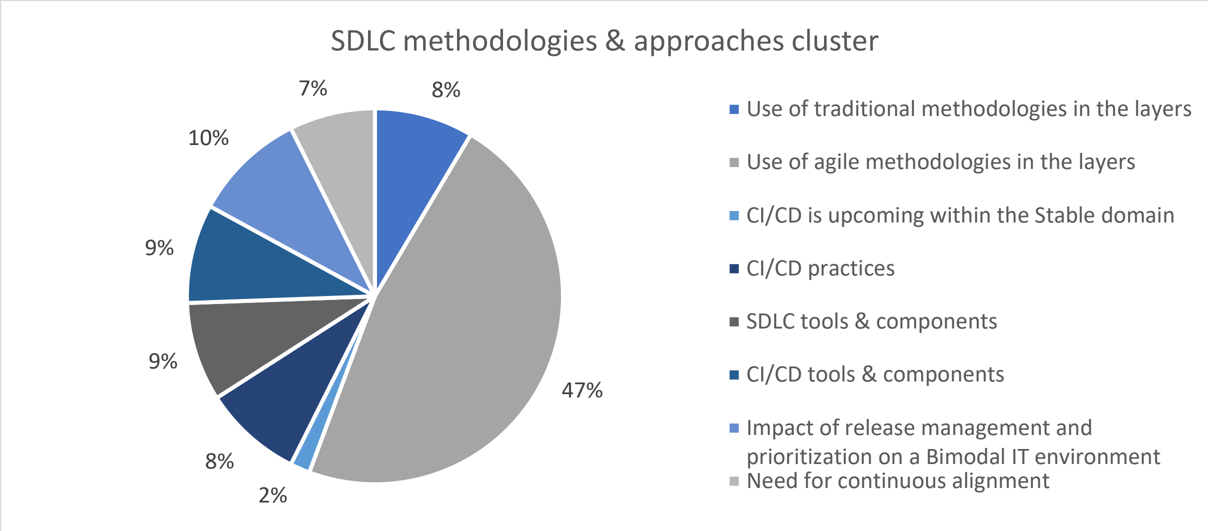

Figure 5-6 SDLC methodologies and approaches cluster with related topics

Figure 5-7 complements the data analysis regarding the usage of SDLC methodologies and approaches within the different layers. It also presents the codes to describe the needed tools and components for developing and delivering the IT systems within a Bimodal IT platform.

Use of traditional methodologies in the layers

The first topic focuses on the traditional methodologies and approaches used within the different layers, such as Waterfall, V-Shape, PRINCE2, and other related approaches.

Based on the data analysis, it becomes clear that only 15 responses can be allocated to this topic. The data analysis points out that only the code "Usage of traditional methodologies in the Stable domain" is mentioned by all the candidates. Due to this result, the conclusion can be drawn that the Stable Layer is the only layer within a Bimodal IT environment that makes use of traditional approaches when developing and delivering IT systems.

Use of agile methodologies in the layers

The second topic discusses the agile methodologies and approaches used within the different layers, such as Scrum, Kanban, DevOps, CI/CD, and other related approaches. The data analysis also reveals that this topic is the main topic of the cluster because 83 of the 176 responses can be allocated to this subject.

The first observation of the data shows that the Explorative Layer solely uses agile methodologies to develop and deliver IT systems within the layer. Also, the code "Mentions that CI/CD is used in the Explorative domain" is mentioned by all the candidates and discloses that CI/CD is the most conducive approach to support the development and delivery of the IT systems within the Explorative Layer.

The data reveals an intriguing and interesting result about the usage of agile approaches within the Stable Layer. As indicated in the previous section, the code “Usage of traditional methodologies in the Stable domain” is mentioned by all the candidates. However, this observation is also applicable for the code “Usage of agile methodologies in the Stable domain.” Based on these observations, the conclusion can be drawn that the Stable Layer uses traditional as well as agile approaches to guarantee the development and delivery of the IT systems within the layer.

Regarding the use of CI/CD within the Stable Layer, the data shows that the usage is not mentioned by any candidate.

The fourth observation discloses the SDLC methodologies used within the Communication Layer. Both the codes “Usage of agile methodologies in an ESB environment” and “Usage of agile methodologies in a CIP environment” are mentioned four times, indicating that agile approaches are a fitting method to set-up the integration patterns and mechanisms within an ESB and CIP environment. The same result applies to the MSA environment. The code “Usage of agile methodologies in a MSA environment” is mentioned by all the candidates. The candidates also indicate through the code “Mentions that CI/CD is used in a MSA environment” that CI/CD is the preferred method for developing and delivering software within a MSA environment. Since the approach itself is very suitable to support the creation, delivery, and exploitation of microservices.

CI/CD is upcoming within the Stable domain

As mentioned, CI/CD is currently not used or even not applicable within the Stable Layer. However, candidates 7, 10, and 14 discuss that CI/CD shows an upcoming trend within the Stable Layer because the approach is getting attention from the established software vendors. The expectation of candidate 10 is “that CI/CD will be adopted in every layer in the near future because the need for fast delivery is increasing. Also, the prominent and established vendors have noticed this. For example, SAP is offering CI/CD for Fiori developments in SAP S/4HANA.” However, the question remains to which extend CI/CD will be used within the Stable Layer because it is conceivable that an organization does not want to adjust its business-critical objects on a frequent base.

CI/CD practices

This topic focuses on the implications and common configuration practices of CI/CD. The code “Mentions CI/CD practices” shows that all the candidates mention one or more practices to facilitate CI/CD.

Table 5-8 presents all the practices to facilitate and ensure CI/CD.

CI/CD practices
A CI/CD strategy needs to be in place
A recovery strategy should be in place
A proper branching structure should be in place
Increments should be small. This minimizes the branches
Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages
Only required features and functionalities should be build and released
The developer should test the increment rigorously on his/her local environment
Within the pipeline, the proper tests should be configured and kicked off
A verification step should be in place before each deployment
Configurations and passwords are easily accessible and always up-to-date
Use of a Version Control System
Use of an Automation Server that can automatically build, test, publish and deploy the various increments
Use of a Release Manager tool to deploy the application on any environment
Provide KPIs to measure the pipeline executions
Monitoring and analytics need to be in place to measure and analyze the solutions
All the affected layers must confirm to an agile approach. Hereby, DevOps is the preferred method

Table 5-8 Overview of the CI/CD practices (details of analysis can be found in appendix 4 on page 197)

SDLC tools and components

The fifth topic defines and describes the SDLC tools and components.

As per the data analysis, all the candidates are able to indicate the SDLC tools and components related to the development and delivery of the IT systems within a Bimodal IT environment. Also, the tools and components do not distinguish between the different layers.

The SDLC tools and components are presented in table 5-9.

SDLC tools and components to realize, facilitate, exploit, and support the development and delivery of software within a Bimodal Architecture environment
A tool/system to manage the planning
A tool/system to manage the requirements
A tool/system that supports Project Management and/or IT Service Management activities
A tool/system for tickets
Functional and technical design tools
Collaboration tools
A Document Management System

Version Control systems/Repositories
Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, software, and APIs
Automation server (build, test, publish and deploy automation)
Middleware tools
Software Quality Assurance tools. This category also includes test tools
Release and Deployment tools
Monitoring and Analytics tools

Table 5-9 Overview of SDLC tools and components to develop and deliver software (details of analysis can be found in appendix 4 on page 195)

CI/CD tools and components

This topic defines and describes the CI/CD tools and components.

The analyzed data shows the same result in comparison to the indication of the SDLC tools and components. The reason is that the selected candidates have a high-level overview of the tooling and components used within a Bimodal IT platform.

The CI/CD tools, components, and functionalities are presented in table 5-10.

CI/CD tools and components to realize, facilitate, exploit, and support software
Use of local IDEs or DKs
Completed source code are submitted to a Repository
Board functionality. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug Tracking
Version Control systems/Repositories
Artifact Repository. To store, publish, and share software packages securely
Pipeline functionality. This functionality is an automation server and is used to automate the build, test, and publish activities
Release functionality. To automate the deployment step
Monitoring and analysis functionalities. These tools are used for monitoring, analysis, and visualization
Alert functionality. To notify a developer whether the build succeeded or failed
A tool to automate and test the various API calls

Table 5-10 Overview of CI/CD tools and components to develop and deliver software (details of analysis can be found in appendix 4 on page 196)

The CI/CD tools and components are primarily used to transform the business requirements and source code into an increment (software packages).

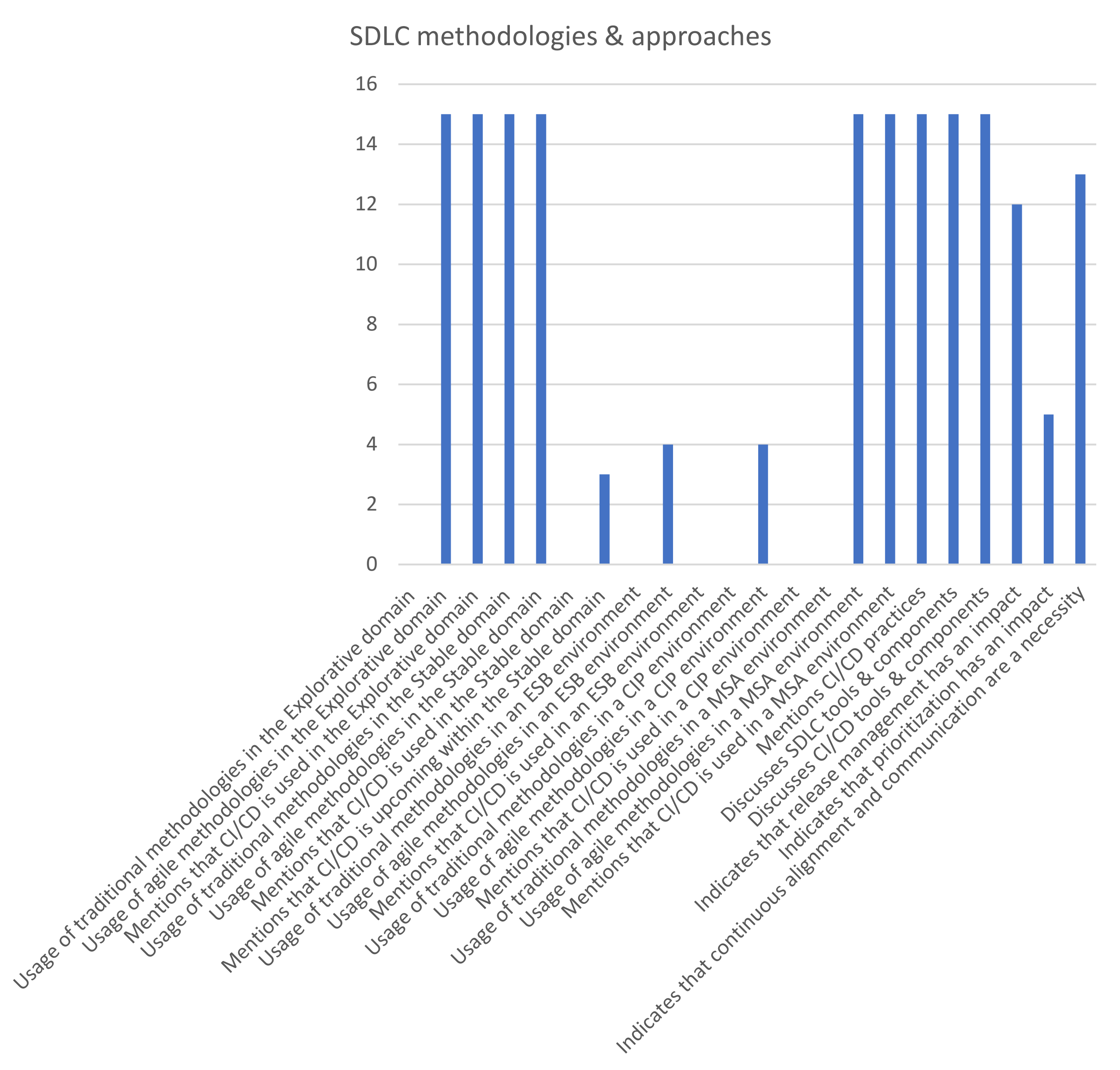

Figure 5-7 Result overview on the SDLC methodologies and approaches

Impact of release management and prioritization on a Bimodal IT environment

The aim of the seventh topic is to discuss the factors that can have an impact on a Bimodal IT platform.

From the data, it is observable that in total, 17 responses are used to indicate that release management and prioritization impact a Bimodal IT environment. The code “Indicates that release management has an impact” is mentioned 12 times and discusses that without a proper release management strategy, the delivery of software within a Bimodal Architecture environment cannot be guaranteed. The reason is that uncontrolled releases can result in integration issues or even jeopardize the operation after go-live due to misaligned functionalities and services. In contrast, the code “Indicates the prioritization has an impact” is mentioned five times and indicates that the user and functional requirements can be compromised during software development and delivery without proper prioritization.

Need for continuous alignment

The final topic of the cluster discusses how to prevent the impact within a Bimodal IT environment.

The code "Indicates that continuous alignment and communication are a necessity" is mentioned 13 times. The candidates indicate that the different teams need to be continuously aligned due to the different layers and approaches within a Bimodal IT environment. By focusing on continuous alignment, the multiple teams will be able to manage the various IT projects, oversee the different deployments, and reduce interoperability issues. Besides, the alignment will also help to keep the several teams informed, up-to-date, and aware of the planning and requirements changes.

5.2.4. Deployment

Figure 5-8 presents an overview of the topics regarding the deployment approaches of a Bimodal IT platform. The figure also includes the distribution of the topics within the cluster.

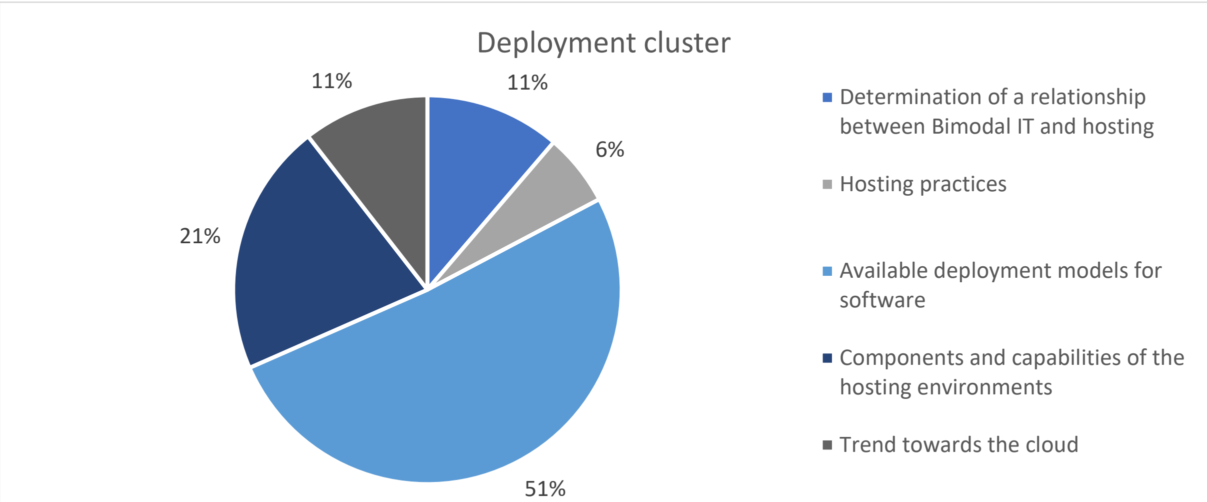

Figure 5-8 Deployment cluster with related topics

Figure 5-9 complements the data analysis regarding the available deployment models for the different system domains within a Bimodal IT environment. It also presents the codes to define and describe the hosting platforms for each layer.

Determination of a relationship between Bimodal IT and hosting

The first topic focuses on a determinable relationship between the available hosting approaches and a Bimodal IT platform. In total, 15 responses are provided to indicate if this relationship is identifiable or not.

The code "Mentions that there is no relationship between Bimodal IT and hosting" is mentioned by all the candidates and discloses that there is no significant relationship identifiable between a specific deployment model and the layers within a Bimodal IT platform.

However, in practice, the statement is discussable because a lot of organizations associate cloud with agility and the rapid development and deployment of software. In contrast, an on-premise or a data center environment is seen as slow and rigid. From the researchers' experience, most organizations are hosting their fast-changing IT systems mainly in the cloud. Whereas the business-critical IT systems are primarily deployed on-premise or in a data center. This observation could lead to a relationship, but this is not the case as per the data analysis. Also, the code "Discusses that there is a relationship between Bimodal IT and hosting" is not mentioned by any candidate, which supports the statement that there is no relationship identifiable between a Bimodal IT platform and the deployment models.

Hosting practices

The second topic focuses on the hosting practices. The topic refers to the question if the selected cases for this study have general deployment practices in place for the deployment of the different kinds of IT systems. In total, eight responses are provided to answer this question.

Regarding the hosting practices, the data analysis shows an intriguing and interesting result. The code "There are practices in place for hosting different kinds of systems" is mentioned four times, whereas the code "There are no common practices in place regarding to hosting" is also mentioned four times. This deviation in the result can be clarified via the statements provided by candidates 7 and 9. Both candidates indicate that organizations with a clear IT strategy mostly have deployment practices in place, while for the organizations without a proper IT strategy or vision apply that the availability of common deployment practices is unclear or not even present at all.

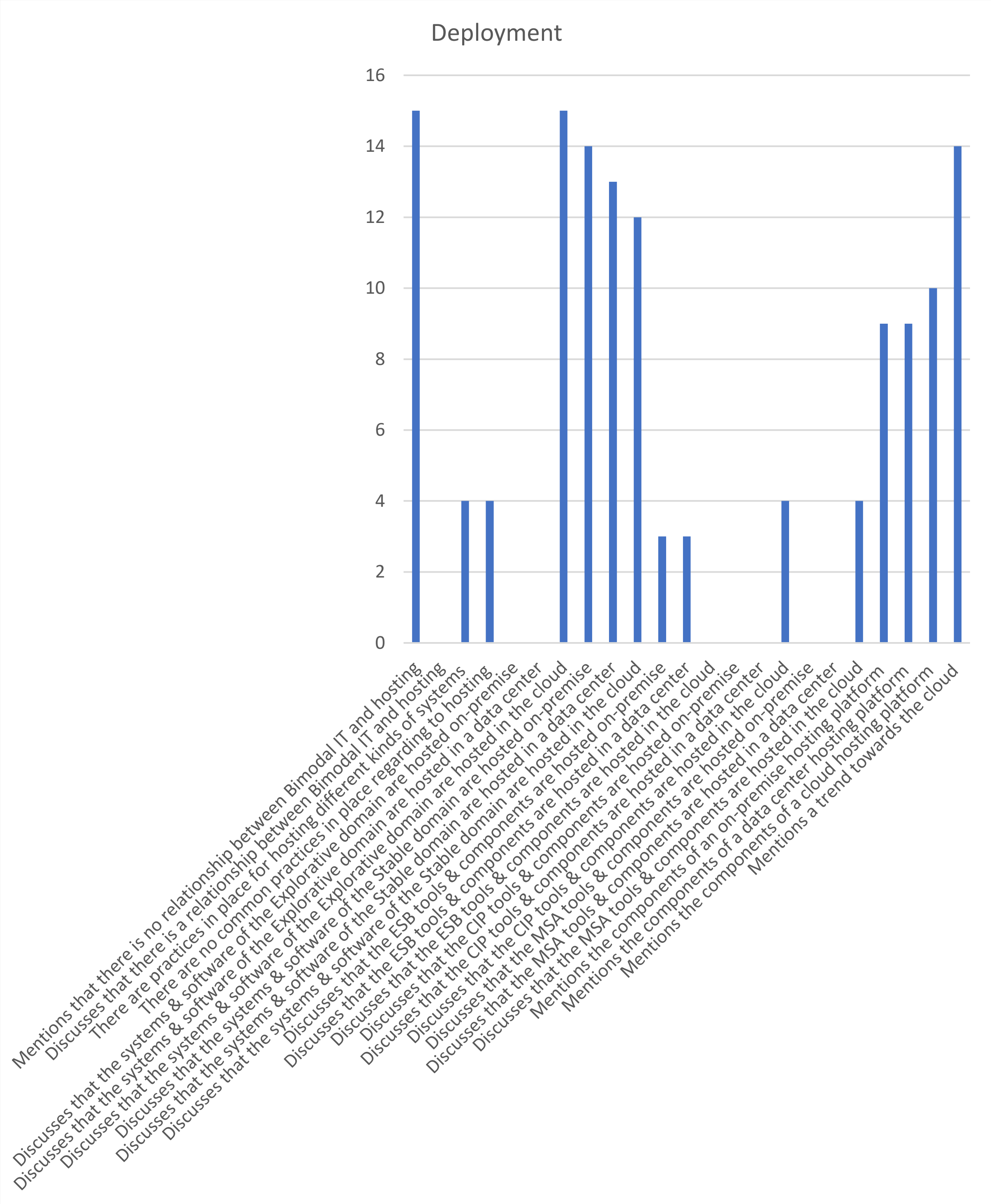

Figure 5-9 Result overview on hosting

Available deployment models for software

The third topic discusses and describes the multiple deployment models for software. The most common way to host software is through an on-premise, data center, cloud, or via a combination of these environments. Figure 5-8 also elucidates that this is the main topic of the cluster because 68 responses of the 133 responses are allocated to this subject.

The first observation relates to the code “Discusses that the systems & software of the Explorative domain are hosted in the cloud.” This code is mentioned by all the candidates and reflects that the IT systems embedded in the Explorative Layer are generally deployed in the cloud.

For the IT systems embedded in the Stable Layer, the data analysis tells another story. The codes “Discusses that the systems & software of the Stable domain are hosted on-premise,” “Discusses that the systems & software of the Stable domain are hosted in a data center,” and “Discusses that the systems & software of the Stable domain are hosted in the cloud” are mentioned at least by 12 candidates, indicating that the IT systems of the Stable Layer can be hosted through all the available deployment models.

In addition, the data also provides an insight into the deployment models regarding the embedded service-based environments within the Communication Layer. By counting the codes “Discusses that the ESB tools & components are hosted on-premise” and “Discusses that the ESB tools & components are hosted in a data center,” the data analysis discloses that six responses are favoring the deployment models on-premise and data center over the cloud in case of an ESB solution. In contrast, the code “Discusses that the ESB tools & components are hosted in the cloud” is not mentioned at all. Based on these results, the conclusion can be drawn that ESB solutions are primarily deployed in an on-premise or data center environment.

Regarding the deployment approaches of a CIP or MSA environment, the conducted data analysis allows to draw the conclusion that these environments are primarily deployed in a cloud environment. This conclusion is based on the zero responses provided on the codes “Discusses that the CIP tools & components are hosted on-premise,” “Discusses that the CIP tools & components are hosted in a data center,” “Discusses that the MSA tools & components are hosted on-premise,” and “Discusses that the MSA tools & components are hosted in a data center.” However, based on the researchers’ experience, this does not necessarily mean that a CIP or a MSA solution cannot be deployed in an on-premise or data center environment.

Components and capabilities of the hosting environments

The fourth topic focuses on the components and capabilities of the described hosting platforms. As per the data analysis, 28 of the 133 responses are allocated to this topic.

The first observation reveals that both the codes “Mentions the components of an on-premise hosting platform” and “Mentions the components of a data center hosting platform” are mentioned nine times. The reason is that most of the selected cases have their (core) IT systems, or a part of it deployed on-premise or in a data center environment. Therefore, most of the candidates are able to indicate the components and capabilities of an on-premise or data center environment.

The components of an on-premise and data center hosting platform are presented in table 5-11. The related capabilities are presented in table 5-12.

Components of an on-premise or data center deployment model
Server Farm (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)
VPN or (Reverse) Proxy

Table 5-11 Overview of the infrastructural components of an on-premise or data center deployment model (details of analysis can be found in appendix 4 on page 199)

Capabilities of an on-premise or data center deployment model
Load balancing
Data access
File access
Data cache
Web cache
Process scheduling

Table 5-12 Overview of the capabilities regarding an on-premise or data center deployment model (details of analysis can be found in appendix 4 on page 199)

The second observation relates to the components and capabilities of a cloud environment. The code "Mentions the components of a cloud hosting platform" is mentioned by ten candidates and indicates that the majority of the participants are able to define and describe the components and capabilities of a cloud environment. Also, due to a trend towards the cloud, the components are becoming more well-known and recognizable.

The components of a cloud hosting platform are presented in table 5-13, while the capabilities are presented in table 5-14. Also, cloud hosting companies are making mainly use of subscription models to provide their services. Therefore, the available subscription models are presented in table 5-15.

Components of a cloud deployment model
Physical servers located in a hosting center (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)
Containerization technology

Table 5-13 Overview of the infrastructural components of a cloud deployment model (details of analysis can be found in appendix 4 on page 199)

Capabilities of a cloud deployment model
Images (creation and discovery)
Event-driven serverless compute platform

Cloud solution is offered through a Public, Private, or Hybrid model

Table 5-14 Overview of the capabilities regarding a cloud deployment model (details of analysis can be found in appendix 4 on page 199)

Subscription methods of a cloud deployment model
Software as a Service (SaaS)
Platform as a Service (Paas)
Integration Platform as a Service (iPaaS)
Function as a Service (FaaS)
Back-end as a Service (BaaS)
Infrastructure as a Service (IaaS)

Table 5-15 Overview Subscription methods within a cloud deployment model (details of analysis can be found in appendix 4 on page 199)

Trend towards the cloud

As indicated in the previous section, the data analysis shows a movement towards the cloud. This trend is acknowledged through 14 responses. However, few candidates indicate that some IT systems will never move towards the cloud due to their importance and application. Therefore, organizations need to determine which IT systems can move towards the cloud and which IT systems need to stay or be deployed in an on-premise or data center environment.

5.2.5. Bimodal relevancy, adoption, and implications

Figure 5-10 presents an overview of the topics regarding the relevancy, adoption, and implications of a Bimodal IT platform. The figure also presents the distribution of each topic.

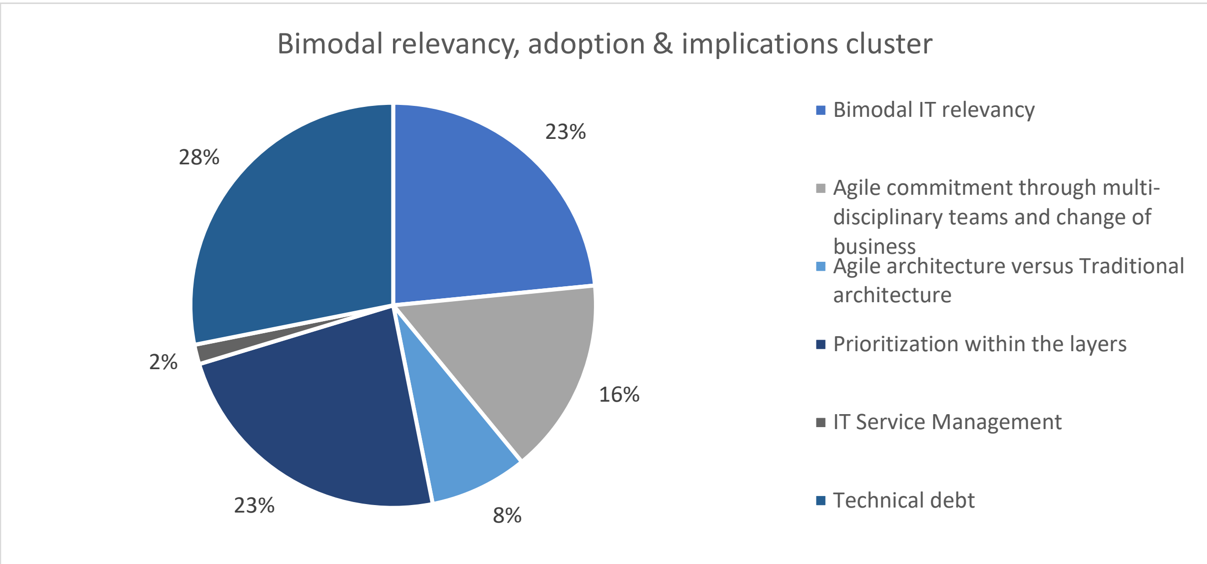

Figure 5-10 Bimodal relevancy, adoption, and implications cluster with related topics

Figure 5-11 complements the data analysis by providing the codes to describe the future, relevancy, processes, and implications of a Bimodal IT environment.

Bimodal IT relevancy

The first topic discusses the relevancy of a Bimodal IT platform. The data analysis discloses that a Bimodal IT environment will stay relevant within the future. To support this statement, the code "Bimodal IT will stay relevant in the future" is mentioned by 13 candidates. Only two candidates indicate something different. Candidate 4 discusses that "the original concept is nice, but it will not work in practice. The Pace-layered methodology is more suitable." While candidate 7 indicates that "the Modes will vanish. The fact is that vendors and suppliers of Mode 1 systems need to change and adjust their business models to compete with the competitors of Mode 2. On the other hand, we will still need stable and secure core systems to store our business-critical objects and data. Also, innovation will never stop. So, the future will determine what will happen with the Bimodal IT concept."

Agile commitment through multi-disciplinary teams and change of business

The second topic focuses on the team set-up and the business to realize and facilitate agility. From the 64 responses, ten responses are dedicated to describe this topic.

The code "Need for a multi-disciplinary team set-up as per the agile manifesto" is mentioned six times. It shows that there is a need for a multi-disciplinary team set-up as per the agile manifesto to realize and facilitate agility within a Bimodal IT environment. On top of that, four candidates discuss that if an organization wants to commit to the agile way of working, the business should change itself.

Agile architecture versus Traditional architecture

The third topic discusses that an Agile architecture diverse from a Traditional architecture. In total, five candidates indicate that a sub-architecture or another viewpoint needs to be created to facilitate the iterative way of working since an Agile architecture is different in comparison to a Traditional architecture.

Prioritization within the layers

As for the prioritization, nine candidates indicate that the priority is commonly set by a Product Owner (PO) within the Explorative Layer. However, this is not the case for the Stable Layer because six candidates indicate that the business conducts the priority setting within the Stable Layer instead of a PO. This way of working does not comply with the best practices regarding agile. Therefore, it is essential that the business change itself to commit to the agile way of working within a Bimodal IT environment.

IT Service Management

The fifth topic focuses on IT Service Management within a Bimodal IT environment. This topic is only referred by one candidate.

Candidate 7 indicates that IT Service Management does not distinguish between the Stable and Explorative domains. Therefore, IT Service Management can be used in all the layers of a Bimodal IT platform.

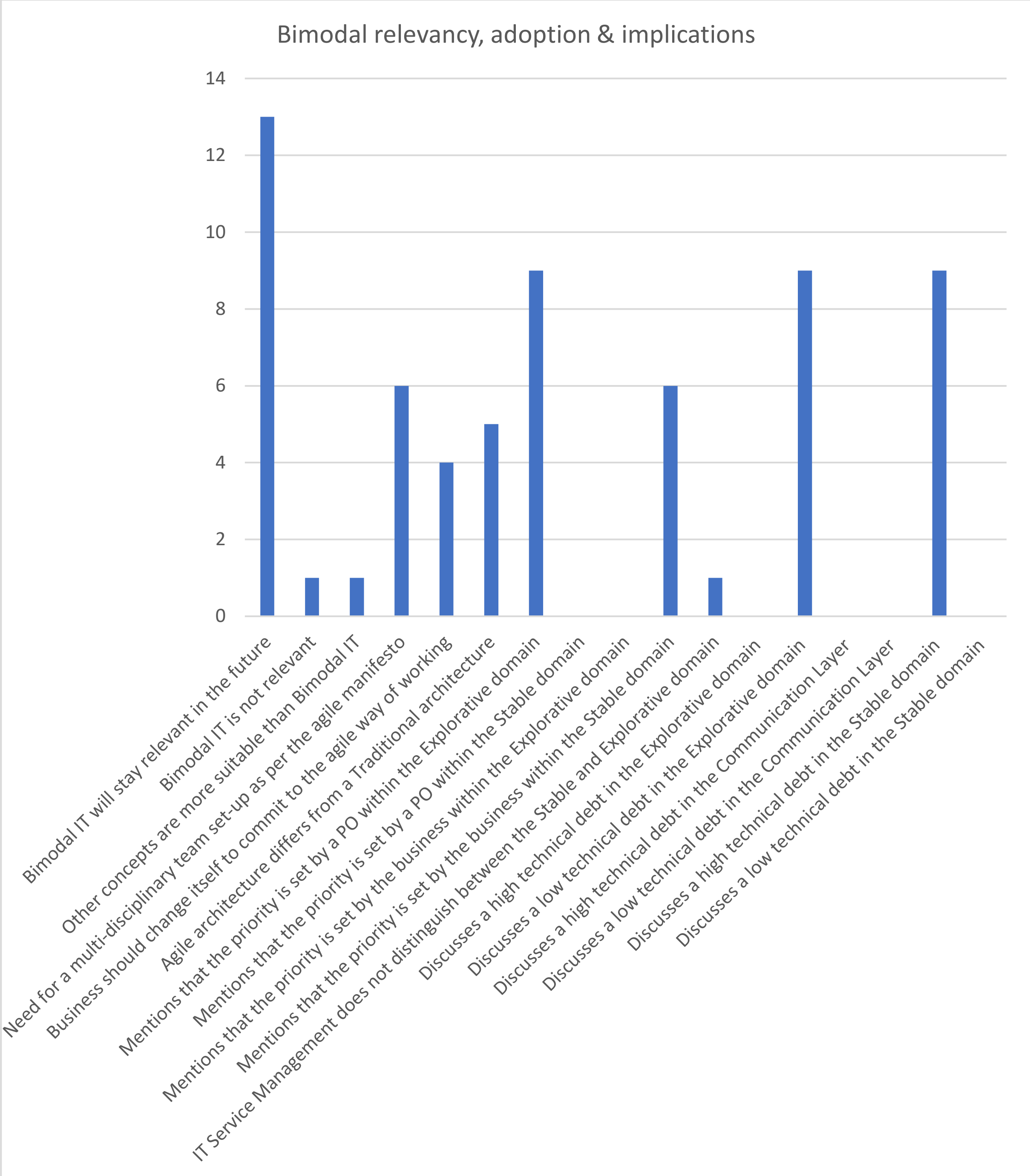

Figure 5-11 Result overview on the relevancy, adoption, and implication of a Bimodal IT environment

Technical debt

The final topic of the cluster discusses the technical debt of the different layers. In total, 18 of the 64 responses are discussing the availability of some technical debt within the layers of a Bimodal IT platform.

The data analysis reveals through the code “Discusses a low technical debt in the Explorative domain” that the Explorative Layer has a low technical debt. Nine candidates support this statement.

For the Stable Layer applies that it has a high technical debt. The reason for this debt is related to the embedded IT systems within the layer since the majority of IT systems are complex and contain custom codes to realize the business requirements. Besides, there is also an urgent need to expose the various functionalities as services to support digitalization. This statement is also supported by nine candidates.

5.3. Findings

In this section, the findings are presented. The findings are based on the results discussed in chapter 5.2 and on the minimum quantification by four candidates, as indicated in chapter 4.11.1. The major findings of the data analysis are discussed in chapter 8.

5.3.1. Bimodal environment findings

Pace-layered Application Strategy and Digital Transformation as an alternative for Bimodal IT

The first finding is related to alternative or additional concepts rather than Bimodal IT. The data analysis reveals that six candidates (candidates 1, 5, 8, 9, 10, and 11) mention that the Digital Transformation is a famous concept next to Bimodal IT to achieve agility and to support and manage the current IT landscape. For the Pace-layered Application Strategy applies that this concept is only mentioned by three candidates (candidates 4, 9, and 12). However, four candidates (candidates 3, 6, 7, and 8) are using the terminologies of the Pace-layered Application Strategy to explain and allocate the different types of IT systems into the various layers of a Bimodal IT platform.

Comparison Bimodal IT with Pace-layered Application Strategy and Digital Transformation

The Pace-layered Application Strategy concept from Gartner (Gartner, 2012b) and the concept of Digital Transformation developed by Ross and colleagues (Ross et al., 2018) are similar to the concept of Bimodal IT. The reason is that all the three concepts are trying to deal with the same phenomena, namely to improve and enhance the customer experience and create new revenue streams as well as “keeping the lights on” within the organization and its departments (Haffke et al., 2017a; Horlach et al., 2017). Since the concepts are emphasizing on multiple layers, business-critical versus customer-facing IT systems and services (Bils, 2014), the concepts can be used interchangeably because they serve the same purpose.

To prevent confusion, and because the concepts are derived from the initial concept of Gartner, the concept of Bimodal IT will still be in the lead as the main concept within the research. Also, to group, indicate, and allocate the different IT systems within the RA, the system categories of the Pace-layered Application Strategy concept will be used as a design pattern within the layers to capture and model these elements.

To define the categories in this study, the definitions of the system categories will be derived and based on the definitions provided by Gartner (Gartner, 2012a). The Systems of Record refers to IT systems that support core transaction processing and store and handle business-critical objects and data. These IT systems need to be stable and guarantee a smooth operation. If this is not the case, it can jeopardize the whole operation and become an obstacle for the continuation. The Systems of Differentiation represent IT systems that extend and enable unique business processes and industry-specific capabilities. These IT systems need to be configurable and flexible until a certain point. And the final category, namely the Systems of Innovation, represents IT systems used to engage, interact, and increase user and customer intimacy. These IT systems address new business requirements, opportunities and support the digitalization strategies and activities of an organization.

Bimodal IT comprises three layers

Despite the different concepts, every candidate agrees that a Bimodal IT environment comprises three layers. This is in line with the theory described by Serrano et al. (2014). Serrano et al. (2014) argue that all business-critical IT systems should be wrapped in a service layer and plugged into a service-oriented environment to enable digitalization (Serrano et al., 2014). The layers of a Bimodal IT platform are:

- A Stable Layer;
- An Explorative Layer;
- And a Communication Layer.

The Stable Layer holds Operational value-adding and Stabilized Systems of Innovation

All the candidates also point out that not every API, web application, and new technology and solution that falls under the umbrella "Systems of Innovation" is part of the Explorative environment. APIs, web applications, IT systems used for digitalization, and new technologies and solutions, such as an expense app, a time register app, a safety app, or sensor technology that help the business to create operational value by providing extra information, increasing responsiveness, delivering operational efficiency, or supports an organizational goal are part of the Stable environment and are referred to as "Operational value-adding and Stabilized Systems of Innovation."

Stable facilitates Explorative

From the data, it becomes clear that it is not possible to realize and facilitate the Explorative Layer without the Stable Layer. All the candidates support this finding.

Integration patterns of the Stable Layer

The data discloses that the Stable layer has multiple integration patterns in place to establish the connectivity and integration between the Systems of Records, Systems of Differentiation, and Operational value-adding and Stabilized Systems of Innovation (these IT systems add value by simplifying or improving the operations). From the data analysis, it is observable that 11 candidates (candidates 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, and 15) indicate that manual integration, point-to-point integration (JMS), use of international standards for data exchange (EDIFACT), EAI, and an ESB are common practices for setting up the connectivity and integration between the IT systems. However, the integration patterns themselves are not limited to these mechanisms because the Stable Layer can also make use of JDBC, proxies, OData sharing through a gateway (Reimer et al., 2013), and other middleware components to establish the integration and interoperability between the IT systems.

In addition, all the candidates indicate that the functionalities embedded in the Stable Layer are exposed as services to realize, facilitate and support digitalization.

Integration patterns of the Explorative Layer

Also, all the candidates discuss that the Explorative Layer has multiple integration patterns in place to establish the connectivity and integration between the multiple Systems of Innovation (IT systems intended for external stakeholders, such as customers). The integration patterns themselves are based on services. Mainly on microservices.

Separate API platforms

The data analysis discloses the need for a separate internal and external API platform. The two API platforms can help to realize and facilitate the split between the Operational value-adding and Stabilized Systems of Innovation and the IT systems embedded in the Explorative Layer. This statement is underlined by five candidates (candidates 2, 8, 10, 11, and 13).

Communication Layer links the other layers

All the candidates support the fact that the Stable Layer and Explorative Layer are connected through the Communication Layer. Also, the Communication layer realizes the API platform(s) through the use of one or more service-based environments.

Movement of stabilized and matured IT systems from the Explorative Layer

This finding relates to the movement of matured web apps, IT systems used for digitalization, and technologies towards the Stable Layer. In total, six candidates (candidates 5, 6, 7, 8, 9, and 12) indicate that the stabilized and matured IT systems can move from the Explorative Layer to the Stable Layer when it reaches a certain purpose. The candidates also discuss that the solutions should be embedded in a different system category compared to the Systems of Record and Systems of Differentiation because the matured IT systems fulfill a different purpose and do not necessarily expose any services or functionalities towards other IT systems and domains.

Stable Layer characteristics

Table 5-16 presents the significant characteristics of the Stable Layer. The table also includes how often the candidates mention a particular characteristic.

Characteristics Stable Layer	**Counts and associated candidates**
Stable	Mentioned 15x
Reliable	Mentioned 15x
Robust	Mentioned 12x by the candidates 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, and 15
Secure	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14, and 15
Supports and stores business relevant objects and data	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
It does not change a lot once deployed in production	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Comprises fewer errors	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14, and 15
Traditional and agile way of software delivery	Mentioned 15x

Solid test processes	Mentioned 8x by the candidates 1, 2, 4, 5, 9, 11, 12, and 13
Problem-solution design	Mentioned 7x by the candidates 1, 2, 4, 5, 11, 12, and 13
Focus on standardization	Mentioned 6x by the candidates 2, 5, 6, 7, 12, and 13
Delivers operational value	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Implementation costs are high	Mentioned 10x by the candidates 2, 6, 8, 9, 10, 11, 12, 13, 14, and 15
High availability and fault tolerance	Mentioned 12x by the candidates 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, and 15
The Stable domain is a facilitator for the Explorative domain	Mentioned 4x by the candidates 1, 3, 4, and 7
Core systems are not easily replaceable due to their importance	Mentioned 12x by the candidates 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15

Table 5-16 Significant characteristics of the Stable Layer

Explorative Layer characteristics

In comparison to the characteristics of the Stable Layer, table 5-17 presents the significant characteristics of the Explorative Layer. The table also shows how often a particular characteristic is mentioned by the candidates.

Characteristics Explorative Layer	**Counts and associated candidates**
Flexible	Mentioned 14x by the candidates 1, 2, 3, 5,

	6, 7, 8, 9, 10, 11, 12, 13, 14, and 15
Supports and stores minimal or no business relevant objects and data	Mentioned 13x by the candidates 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, and 15
Errors are already factored in	Mentioned 10x by the candidates 1, 2, 5, 8, 9, 10, 11, 12, 14 and 15
Only an agile way of development is supported	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Solutions are centered around interaction and experience	Mentioned 6x by the candidates 1, 2, 5, 11, 12, and 13
Trend to have only a development and production environment	Mentioned 5x by the candidates 1, 2, 4, 5, and 9
Delivers business value	Mentioned 14x by the candidates 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15
Rapid delivery oriented. Often and frequent deployments	Mentioned 15x
24/7 availability. It can be achieved through an off-line capability	Mentioned 13x by the candidates 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15
Requirements and planning changes continuously	Mentioned 8x by the candidates 1, 2, 3, 4, 7, 8, 9, and 14
Features are small and independent	Mentioned 11x by the candidates 3, 4, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Focus on innovation, experimentation, and exploration	Mentioned 12x by the candidates 1, 2, 4, 6, 7, 8, 9, 10, 12, 13, 14, and 15

Continuous integration/deployment in place to increase time-to-market	Mentioned 9x by the candidates 1, 2, 5, 10, 11, 12, 13, 14, and 15

Table 5-17 Significant characteristics of the Explorative Layer

Embedded IT systems in the Stable Layer

As indicated in chapter 4.11.1, only a subset of the IT systems embedded within the Stable Layer are selected and presented in this study. The reason is that the text data provides more than 170 IT systems and functionalities, and by considering all the components will add unnecessary complexity into the RA. A subset of the components will be sufficient to explain which IT systems are or should be part of this layer. In addition, the obtained IT systems that share similar functionalities or capabilities are grouped together into generic system domains to increase the selection range and to cover as much as IT systems possible.

Table 5-18 discloses the embedded IT systems within a Stable Layer.

Embedded IT systems within the Stable Layer
ERP software (comprising SAP S/4HANA, SAP ECC, Dynamics 365 Business Central, Oracle eBS, and AFAS)
Business Intelligence and Business Analytics software (comprising MS Power BI, SAP BI, Oracle Data warehouse, Cognos, MDM, Report Builders, and Dashboard Builders)
CRM software (comprising Salesforce, SAP CRM, SAP C4C, SAP Service Cloud, Dynamics 365 CRM, and Dynamics 365 Customer Service)
Industry-Specific Software (comprising SAP IS-U, a BRP ((Basic Registration of Persons)) system, a Social Security System, Mecoms, a Sorting application, a Safety App, a Meter Reading App, a Digiduinkaart App, an App for Reporting bulk waste, a Track and Trace API ((for internal agents)), and an App for Reporting vandalism)
HRM software (comprising SAP SuccessFactors, Dynamics 365 Human Resources, Youforce, Workday, a Time and Expense App, and a Mileage Tracking App)
Planning software (comprising Anaplan, Luminate Planning, and ClickSoftware)
Project and ITSM software (comprising JIRA, TopDesk, and ServiceNow)
SCM software (comprising SAP Ariba, SAP Ariba Mobile App, Dynamics 365 Supply Chain Management, and Oracle SCM)
PLM software (comprising Teamcenter)
MES software (comprising Net@Pro)
Finance and Billing software (comprising Esize, Dynamics 365 Finance & Operations, SAP Simple Finance, and SAP Hybris Billing)
Banking software (comprising Aquarius, Avaloq Core, a Foreign Exchange (FX) Trade API, and Tikkie API and associated App)

Operational value-adding software (comprising a Track and Trace API and functionality ((for the customer contact agent)), and SAP Fiori)
Web and Mobile enabling software (comprising MS Power Platform)
Interaction, Communication and Collaboration software (comprising Microsoft Outlook and Genesys)
Booking and Reservation software (comprising Egencia Travel Management and Egencia Travel Management App)
IoT solutions (comprising Sensor APIs and Chatbots)

Table 5-18 Overview of relevant IT systems that are embedded within the Stable Layer

Embedded IT systems in the Explorative Layer

Table 5-19 presents the embedded IT systems within an Explorative Layer.

Since the text data provides 88 IT systems and functionalities in total, the presented data in the table is also in line with chapter 4.11.1 and discloses only a selected subset of the IT systems embedded within the Explorative Layer. For the different IT systems that are embedded in this layer also apply that the IT systems that share similar functionalities or capabilities are grouped together into generic system domains to cover as much as IT systems possible.

Embedded IT systems within the Explorative Layer
Portal software (comprising Sitecore)
Interaction, Communication and Collaboration software (comprising Teams, Skype, WebEx, Facebook, LinkedIn, and WhatsApp)
DMS software (comprising SharePoint)
Business software (comprising Power Apps, Office 365, Mendix Apps, a Track and Trace App ((consumers and business)), and Azure Digital Twins)
Business Intelligence and Business Analytics software (comprising Google Analytics, Grafana, Power BI Reports, and Dashboards)
Industry-Specific software (comprising a Mail and Parcel App, a Distribution App, a Car parking App, a Visit sign up App, a Tourism Info App, a Neighborhood prevention App, and a Volt Station App for Utilities)
Banking software (comprising a Mobile Banking App, a Grip App, and a Retirement App)
Geographic Information System (GIS) software (comprising ArcGIS ((Esri)) and GeoWEB)
Smart Device software (comprising iOS Apps, Android Apps, and Windows Phone Apps)
IoT solutions (comprising HoloLens, 3D printing, and Augmented Reality solutions)

Table 5-19 Overview of relevant IT systems that are embedded within the Explorative Layer

5.3.2. Middleware findings

Service-based environments

All the candidates support the statement that a Communication Layer can comprise three types of service-based environments. These service-based environments are:

- A SOA environment;
- A CIP environment;
- And a MSA environment.

ESB components

Table 5-20 presents the most significant components of an ESB environment. It also presents the candidates and the number of responses provided for each component.

Components of an ESB	Counts and associated candidates
Message/Integration Builder (comprising a Service Builder, Mediation Flow Builder, and a Service Repository)	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Adapter Builder/Adapter Manager	Mentioned 15x
Integration Server/Integration Engine (comprising an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)	Mentioned 15x
System Landscape (comprising a System Landscape/Tenant Manager)	Mentioned 15x
Integration Controller/Integration Gateway (comprising a Configuration Manager, Web Service Manager, Access Controller, Exception Handler, Event Handler, Mediation Flow Manager, Auditor, and a Logger)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Business Activity Monitor	Mentioned 15x
Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL)	Mentioned 15x

Table 5-20 Significant components of an ESB

ESB capabilities

Table 5-21 presents the essential capabilities of an ESB environment. It also presents the candidates and the number of responses provided for a particular capability.

Capabilities of an ESB environment	Counts and associated candidates
Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Message Tracking, Throttling, and Exception Management)	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Security (provides Authentication, Authorization, Encryption, Certification, and Message Security)	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
Transportation (allows HTTP, HTTPS, SOAP, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15

Table 5-21 Significant capabilities of an ESB environment

CIP components

Table 5-22 includes the most significant components of a CIP environment. The table also includes the candidates and the number of responses provided for a particular component by these candidates.

Components of a CIP	Counts and associated candidates
Cloud Integration Service - Message and Event Hub (comprises a Service/Integration Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Message Queues, Exception handler, Logger, and an Auditor)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Cloud Integration Service - Connectivity Hub (comprises an Adapter Repository, Adapter Builder, Adapter Configuration Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Cloud Integration Service - Integration Engine (comprises an Adapter Engine, Orchestration Engine, Transformation Engine,	Mentioned 15x

Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)	
API Management Service (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager)	Mentioned 13x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15
BPM Service (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)	Mentioned 15x
Gateway Service (comprises a Traffic Manager, Exception Handler, Policy Manager, Version Manager, Call Handler, Logger, and an Auditor)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15

Table 5-22 Significant components of a CIP

CIP capabilities

Table 5-23 presents the significant capabilities of a CIP environment. The table also comprises the candidates and count of each capability.

Capabilities of a CIP environment	**Counts and associated candidates**
Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Service Discovery, Call handling, Message Tracking, Throttling, Rate Limits, Traffic Management, Exception Management, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Security (provides Authentication, Authorization, Encryption, Certification, and Message Security)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15
Transportation (allows HTTP, HTTPS, SOAP, OData, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)	Mentioned 12x by the candidates 1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 14, and 15

Table 5-23 Significant capabilities of a CIP environment

MSA components

Table 5-24 presents the most significant components of a MSA environment. The table also includes the candidates and how often a particular component is mentioned.

MSA components	Counts and associated candidates
Nodes	Mentioned 10x by the candidates 1, 2, 4, 5, 6, 8, 10, 13, 14, and 15
Pods (*represents the running containers)*	Mentioned 10x by the candidates 1, 2, 4, 5, 6, 8, 10, 13, 14, and 15
Scheduler	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Controllers	Mentioned 13x by the candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, and 15
Key-value Database	Mentioned 10x by the candidates 1, 2, 4, 5, 6, 8, 10, 13, 14, and 15
API Gateway	Mentioned 12x by the candidates 1, 2, 4, 5, 6, 8, 9, 10, 11, 13, 14, and 15
API Manager	Mentioned 10x by the candidates 1, 2, 4, 5, 6, 8, 10, 13, 14, and 15
API Server	Mentioned 12x by the candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 13, 14, and 15
Resource Manager	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Container Orchestration Engine	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Storage	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Queues	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Container Registry	Mentioned 13x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, and 15
Container Repository (optional)	Mentioned 11x by the candidates 1, 2, 4, 5, 6, 8, 10, 11, 13, 14, and 15
Network Manager	Mentioned 11x by the candidates 1, 2, 3, 4, 5, 6, 8, 10, 13, 14, and 15
Infrastructure Security Manager	Mentioned 11x by the candidates 1, 2, 3, 4, 5, 6, 8, 10, 13, 14, and 15
Provisioning Manager	Mentioned 10x by the candidates 1, 2, 4, 5, 6, 8, 10, 13, 14, and 15

Logger	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15

Table 5-24 Significant components of a MSA

MSA capabilities

Table 5-25 presents the essential capabilities of a MSA environment and also includes the candidates and how often a particular capability is mentioned.

Capabilities of a MSA environment	Counts and associated candidates
Container Orchestration Platform (provides Container Orchestration, Container clustering, Image Discovery, Container Security, Network settings, Key Vault, Load Balancing, Scheduling, Rollouts, Rollbacks, Self-Healing, and Workflow Management)	Mentioned 10x by the candidates 1, 2, 5, 6, 8, 9, 10, 13, 14, and 15
API Management (provides Service Discovery, Messaging, Routing, Call handling, Transformation, Rate Limits, Traffic Management, Throttling, Caching, Circuit Breaker, Failover, Microservice Security, Configuration Management, Networking/IP, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts)	Mentioned 10x by the candidates 1, 2, 5, 6, 8, 9, 10, 13, 14, and 15
Container (comprises microservices)	Mentioned 11x by the candidates 1, 2, 5, 6, 8, 9, 10, 11, 13, 14, and 15

Table 5-25 Significant capabilities of a MSA environment

5.3.3. SDLC methodologies and approach findings

Solely use of agile methodologies within the Explorative Layer

The data confirms that the Explorative environment solely uses agile approaches to realize, facilitate, exploit, and support the embedded IT systems within the layer. Besides this finding, the data also reveals that CI/CD is the most popular agile technique to deliver the increments within the Explorative domain. This is because the CI/CD approach lends itself as an ideal candidate to realize and utilize IT systems within the particular layer through microservices. All the candidates state both findings.

Stable Layer uses a mix of methodologies

The data discloses that the Stable Layer makes use of traditional as well as agile methodologies. All candidates discuss that both types of methodologies are used to develop and deliver the increments within the layer. Candidate 9 indicates that "the SDLC approach is chosen based on the goal, complexity, and length of the IT project. For example, the most convenient SDLC approach for adopting or implementing a System of Record is the Waterfall methodology, while a System of Differentiation is implemented through Scrum." While, candidate 12 discusses that "a traditional approach is used for the implementation and data migration of a core system. For large IT projects applies that they are developed and delivered using a project management approach. Once a core system is deployed into a production environment, an agile approach is used to maintain and exploit its functionalities and to develop new features."

All the candidates also support the statement that there is no CI/CD pipeline available within the Stable domain.

Only agile methodologies are used within the Communication Layer

As discussed, a Communication Layer can comprise one or more service-based environments. Therefore, the first finding is supported by four candidates (candidates 1, 2, 10, and 11) and relates to the increments (web services) of an ESB or a CIP environment by stating that the associated increments are solely developed and delivered via an agile approach. The same observation applies to the MSA increments (microservices) because all the candidates agree that a MSA environment only makes use of agile approaches to support the development, delivery, and exploitation of its increments. Therefore, the conclusion can be drawn that the increments of the Communication Layer are solely developed and delivered through agile approaches.

In addition, six candidates (candidates 1, 2, 3, 10, 11, and 13) indicate that microservices are mainly delivered through CI/CD. Based on the previous finding and the statement that a MSA environment solely makes use of agile approaches, the conclusion can be drawn that CI/CD is the most popular approach for realizing and delivering the increments within a MSA environment.

CI/CD configuration practices

Table 5-26 presents the main practices of a CI/CD environment. Also, the table contains the candidates and how often a particular practice is mentioned.

CI/CD practices	Counts and associated candidates
A CI/CD strategy needs to be in place	Mentioned 15x
A recovery strategy should be in place	Mentioned 13x by the candidates 1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, and 15

A proper branching structure should be in place	Mentioned 11x by the candidates 2, 4, 7, 8, 9, 10, 11, 12, 13, 14, and 15
Increments should be small. This minimizes the branches	Mentioned 15x
The developer should test the increment rigorously on his/her local environment	Mentioned 14x by the candidates 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15
Within the pipeline, the proper tests should be configured and kicked off	Mentioned 14x by the candidates 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15
A verification step should be in place before each deployment	Mentioned 15x
Configurations and passwords are easily accessible and always up-to-date	Mentioned 10x by the candidates 1, 2, 3, 4, 6, 7, 8, 9, 11, and 12
Use of a Version Control System	Mentioned 15x
Use of an Automation Server that can automatically build, test, publish and deploy the various increments	Mentioned 15x
Monitoring and analytics need to be in place to measure and analyze the solutions	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, and 15
All the affected layers must confirm to an agile approach. Hereby, DevOps is the preferred method	Mentioned 10x by the candidates 1, 2, 3, 5, 6, 7, 8, 9, 11, and 12

Table 5-26 Main practices of CI/CD

SDLC tools and components for the development and delivery of software

Table 5-27 presents the significant SDLC tools and components to realize, facilitate, deliver, exploit, and support the IT systems within a Bimodal Architecture environment. In addition, the table also includes the candidates and how often a particular tool or component is mentioned.

SDLC tools and components	Counts and associated candidates
A tool/system to manage the planning	Mentioned 13x by the candidates 1, 2, 4, 5,

	6, 7, 8, 9, 10, 12, 13, 14, and 15
A tool/system to manage the requirements	Mentioned 13x by the candidates 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, and 15
A tool/system that supports Project Management and/or IT Service Management activities	Mentioned 15x
A tool/system for tickets	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, and 15
Functional and technical design tools	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, and 15
Collaboration tools	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, and 15
A Document Management System	Mentioned 15x
Version Control systems/Repositories	Mentioned 9x by the candidates 1, 2, 4, 5, 6, 7, 8, 9, and 13
Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, software, and APIs	Mentioned 15x
Automation server (build, test, publish and deploy automation)	Mentioned 11x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 12 and 13
Software Quality Assurance tools. This category also includes test tools	Mentioned 15x
Release and Deployment tools	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, and 15
Monitoring and Analytics tools	Mentioned 15x

Table 5-27 Significant SDLC tools and components for the development and delivery of software within a Bimodal IT platform

Universiteit Leiden

CI/CD tools and components for the development and delivery of software

Table 5-28 presents the significant tools and components within a CI/CD environment to develop and deliver multiple increments.

The table also comprises the candidates and the number of responses regarding a particular tool or component.

CI/CD tools and components	Counts and associated candidates
Use of local IDEs or DKs	Mentioned 8x by the candidates 1, 2, 4, 10, 11 12, 13, and 15
Completed source code are submitted to a Repository	Mentioned 8x by the candidates 1, 2, 4, 10, 11 12, 13, and 15
Board functionality. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug Tracking	Mentioned 12x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, and 15
Version Control systems/Repositories	Mentioned 14x by the candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, and 15
Pipeline functionality. This functionality is an automation server and is used to automate the build, test, and publish activities	Mentioned 15x
Release functionality. To automate the deployment step	Mentioned 15x
Monitoring and analysis functionalities. These tools are used for monitoring, analysis, and visualization	Mentioned 15x
A tool to automate and test the various API calls	Mentioned 9x by the candidates 1, 2, 4, 5, 9, 12, 13, 14, and 15

Table 5-28 Significant CI/CD tools and components to realize, facilitate, exploit, and support software

Continuous alignment is a necessity to reduce the impact in a Bimodal IT environment

From the analyzed data, it is observable that continuous alignment and communication are a necessity to reduce the impact within a Bimodal IT platform. Due to the layers and the use of multiple approaches within a Bimodal IT environment, 13 candidates (candidates 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, and 15) explain that the different teams need to communicate frequently and align continuously to create awareness, ensure alignment, and avoids errors and issues during releases. Without a proper alignment, the whole operation can be jeopardized because it can lead to uncontrolled releases or integration issues.

Through the adoption of continuous alignment, the multiple teams are able to manage the different IT projects, oversee the different deployments, and reduce interoperability issues. Besides, the alignment will also help to keep the several teams informed and up-to-date.

Release management and prioritization have an impact on the software delivery

As described in the previous section, the data shows that the delivery of software within a Bimodal Architecture environment cannot be guaranteed without a proper release management strategy. This finding is stated by 12 candidates (candidates 1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14, and 15). Also, five candidates (candidates 2, 4, 11, 13, and 15) discuss that prioritization has an impact on the software delivery within a Bimodal Architecture environment. They indicate that without proper prioritization, the user and functional requirements can be compromised during software development and delivery.

5.3.4. Deployment findings

No relationship quantifiable between a Bimodal IT platform and deployment models

All the candidates agree that there is no significant relationship determinable between a specific deployment model and the layers within a Bimodal IT platform. In general, the IT systems embedded in the Explorative Layer are hosted in the cloud, whereas the critical core IT systems are deployed on-premise or in a data center.

Systems and software of the Explorative Layer are deployed in the cloud

The data underlines the statement that the IT systems embedded in the Explorative Layer are commonly deployed in the cloud. This finding is supported by all the candidates.

The deployment of the systems and software of the Stable Layer depends on the IT strategy

From the data, it becomes clear that at least 12 candidates (candidates 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, and 15) are indicating that the IT systems of the Stable Layer can be hosted through a variety of deployment models. Candidate 5 indicates that “in the case of a critical information system, only the deployment models on-premise, data center, or a private cloud environment are eligible. The reason is that an organization wants to have control over these systems since the systems are supporting core business processes and store critical data.” Consequently, candidates 7 and 9 discuss that it depends on the IT strategy and vision of an organization how and where a particular IT system is deployed.

The CIP and MSA environments are commonly deployed in the cloud

The data also shows that the CIP and MSA environments are mainly deployed in the cloud. This finding is supported by four candidates (candidates 3, 8, 9, and 14).

Components of an on-premise or a data center hosting platform

Table 5-29 presents the significant components of an on-premise or a data center environment. The table also includes the related candidates and the number of responses provided for a particular component.

Components of an on-premise or data center deployment model	Counts and associated candidates
Server Farm (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
VPN or (Reverse) Proxy	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14

Table 5-29 Significant components of an on-premise or a data center hosting platform

Capabilities of an on-premise or a data center hosting platform

Table 5-30 presents the main capabilities of an on-premise or a data center platform. The table also shows the candidates and the number of responses provided for a particular capability.

Capabilities of an on-premise or data center deployment model	Counts and associated candidates
Load balancing	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
Data access	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
File access	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
Data cache	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
Web cache	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14
Process scheduling	Mentioned 9x by the candidates 2, 3, 5, 8, 10, 11 12, 13, and 14

Table 5-30 Significant capabilities of an on-premise or a data center hosting platform

Components of a cloud hosting platform

Table 5-31 presents the main components of a cloud environment. Also, the table comprises the candidates and the number of responses provided for a particular component.

Components of a cloud deployment model	Counts and associated candidates
Physical servers located in a hosting center (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Containerization technology	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14

Table 5-31 Significant components of a cloud hosting platform

Capabilities of a cloud hosting platform

Table 5-32 presents the essential capabilities of a cloud environment and shows the candidates and how often a particular capability is mentioned.

Capabilities of a cloud deployment model	Counts and associated candidates
Images (creation and discovery)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Event-driven serverless compute platform	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Cloud solution is offered through a Public, Private, or Hybrid model	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14

Table 5-32 Significant capabilities of a cloud hosting platform

Subscription models of a cloud hosting platform

Table 5-33 presents the most significant subscription models used within a cloud environment. The table also includes the candidates and the number of responses provided for a particular subscription method.

Subscription methods of a cloud deployment model	Counts and associated candidates
Software as a Service (SaaS)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14

Platform as a Service (Paas)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Integration Platform as a Service (iPaaS)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Function as a Service (FaaS)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Back-end as a Service (BaaS)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14
Infrastructure as a Service (IaaS)	Mentioned 10x by the candidates 1, 2, 3, 5, 8, 10, 11 12, 13, and 14

Table 5-33 Significant subscription models within a cloud hosting platform

Trend towards the cloud

From the data, it is visible that a movement towards the cloud is taking place. Candidate 15 discusses that "cloud deployments models are becoming more prominent." And candidate 5 even highlights that organizations are entering an era in which all the IT systems are moving towards the cloud. The movement is acknowledged by 14 candidates (candidates 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15).

5.3.5. Bimodal relevancy, adoption, and implication findings

Bimodal IT will stay relevant in the future

The data reveals that Bimodal IT will stay relevant as a concept in the future because the need for multiple IT environments within an organization will never disappear. The Stable Layer will stay in the lead for the core IT systems, services, and Operational value-adding and Stabilized Systems of Innovation, whereas the Explorative Layer will facilitate the digital offerings, products, and services. For the Communication Layer applies that it will ensure, realize, and exploit the communication and integration patterns between and within the Stable and Explorative domains. This finding is stated by 13 candidates (candidates 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 15).

Need for a multi-disciplinary team set-up and change of business to become agile

The data indicates through six candidates (candidates 1, 2, 8, 9, 12, and 15) that there is a need for a multi-disciplinary team set-up as per the agile manifesto. On top of that, four candidates (candidates 4, 8, 9, and 15) discuss that if an organization wants to commit and comply with the agile way of working, the business should change itself.

Need for a sub-architecture or viewpoint to facilitate the iterative way of working

Because an Agile architecture differs from a Traditional architecture, five candidates (candidates 1, 2, 4, 7, and 8) indicate that a sub-architecture or another viewpoint needs to be created to realize and facilitate the iterative way of working within an organization.

The Explorative Layer has a low technical debt

The analyzed data discloses that the Explorative environment has a low technical debt. In the case of issues or problems, the increment can be fixed very fast. Mostly in another sprint. The finding itself is supported by nine candidates (candidates 1, 2, 4, 6, 7, 8, 9, 12, and 15).

The Stable Layer has a high technical debt

The final finding indicates that the Stable Layer has a high technical debt because the related IT systems are complex and mostly contain custom code. Also, there is an urgent need to expose the various functionalities as services. Subsequently, these services can be used to facilitate the IT systems within the Explorative Layer. The finding is supported by nine candidates (candidates 1, 2, 4, 6, 7, 8, 9, 12, and 15).

6. Reference Architecture

This chapter describes and presents the modeled RA. The first section provides a guideline on how to read and interpret the created RA. The second section describes the added value of the RA. The third section presents the high-level designs of the architectural layers and the related aspects to support a Bimodal IT platform. Through a set of models, it illustrates and discusses the various layers and their associated system domains and integration patterns. Subsequently, the role of the SDLC environments within a Bimodal IT platform is discussed and visualized. Also, the related tools and components are presented through a model, which supports the delivery of the increments. And the final part of the section presents the models regarding the deployment approaches and how these environments are related and used by and within a Bimodal IT platform.

6.1. Dynamic set-up of the Reference Architecture

Chapter 1.6.4 indicates that the RA will be based and built up through the literature findings and the retrieved elements from the text data. In addition, chapters 1.3 and 4.1 discuss that the focus of the study is to provide the core components, which are minimally required to set-up the foundation of a Bimodal IT platform. The RA itself is developed on these pillars. It has the goal to provide a high-level overview, application and technical architecture, and insight into the components and capabilities that are marked as essential for the foundation, set-up, and realization of a Bimodal IT platform. The first version of the RA, which is developed in this study, can serve established organizations with their transformation and is mainly intended for enterprise, solution, and integration architects. The follow-up versions of the RA can be used by other levels within the organization, such as project managers because these versions will provide more detailed views tailored on the business units, projects, and processes.

It is important to notice that some of the components, sub-components, capabilities, and integration patterns mentioned in the RA are prescriptive but are not limited to these modeled elements. The reason is that some of the elements also depend on the software vendor, the implementation partner, or the internal IT organization and on how a specific solution or environment is set-up and delivered. For example, suppose an organization uses MuleSoft tooling to set-up a SOA environment. In that case, the organization will get an ESB solution and a separate service registry tool to enable service discovery. In contrast, if an organization chooses to use SAP tooling for the set-up of a SOA environment, they will only get one IT system (SAP PI/PO) that comprises all the required components and capabilities.

From an architectural perspective, the RA can be divided into six tiers. The first tier comprises the fundamental layers of the platform. These layers are prescriptive because they represent the first building blocks of a Bimodal IT platform. The second tier defines the system categories that are an integral part of the layers. The system categories are distinguished through Systems of Record, Systems of Differentiation, Operational value-adding and Stabilized

Systems of Innovation, and Systems of Innovation. The service-based environments, which are part of the Communication Layer, can be distinguished in a SOA, CIP, or MSA environment. The third tier describes the different groups of IT systems embedded within the Stable Layer and Explorative Layer. The third tier can also be enhanced with missing components, such as the addition of another group of IT systems that are not obtained in the study. The fourth tier prescribes the core components, sub-components, capabilities, and integration patterns of a SOA, CIP, and MSA environment. In addition, the tier also describes the essential sub-components, capabilities, and related integration patterns of a Stable Layer and Explorative Layer. These elements are mainly based on the findings in chapter 5.3.1. Also, the allocation of the various IT systems within the Stable Layer or the Explorative Layer needs to be based on the layers' characteristics, as mentioned in tables 5-16 and 5-17. The reason is that the layers' characteristics decide in which layer a specific IT system should be embedded. On top of that, the tier can be enhanced further with missing elements and allows the movement of elements from one component to another. For instance, some capabilities need to move when a split is made between an ESB solution and a separate Service Registry component. In essence, the fourth tier allows the modeling of all the IT systems and their capabilities and functionalities in each layer of the platform. The fifth tier prescribes the used SDLC approaches and related tools to support and exploit the layers. And the final tier describes the related hosting models for each layer and its associated environments.

Since the elements are based on the literature and interview findings, there is a chance that some components, capabilities, and integration patterns are not mentioned, modeled, or considered within the RA because the study does not capture these elements. However, this does not mean that the missing elements, such as IT systems, groups of IT systems (system domains), certain capabilities, or integration patterns and mechanisms, cannot be added or modeled within the RA. It is also even possible to shift or move some of the existing elements within the RA to meet the organizational demands and needs, such as the movement of certain capabilities from an ESB component towards a Service Registry component.

Conclusively, the RA itself is set-up dynamically and acts as a framework to provides a baseline in the realization, adoption, and use of a Bimodal IT platform within an organization. An organization can use the RA to plot the IT systems from their IT landscape and verify if they comply to the patterns of a Bimodal IT environment or if they miss any essential element. In addition, the RA can be enhanced with additional design patterns to meet the demands and needs of an internal IT organization and help them to set-up, realize, adopt, or fit the platform into the organization.

6.2. Potential value of the Reference Architecture

Based on the researchers' experience, most of the established organizations already have multiple views and models regarding their IT landscape. The developed RA in this study could be seen as an addition to those existing models and views because the RA provides another way to visualize the existing IT landscape. In addition, the RA can also highlight and disclose some hidden patterns that are currently not covered by the existing views and models. First of all, the RA could reveal if an IT landscape already contains patterns that can be related to Bimodal IT. The second value could provide insight into the stable and explorative IT systems or help to categorize the various IT systems within the IT landscape. The third value could show how the different IT systems are integrated and which integration patterns are preferred or most used.

6.3. Architectural layers of a Bimodal IT platform

6.3.1. First tier of the platform

The literature and obtained results from the text data in chapter 5.3.1 show that the core of a Bimodal IT platform is build up through three layers. These layers are:

- A Stable Layer;
- An Explorative Layer;
- And a Communication Layer.

Figure 6-1 presents the first tier of the RA and discloses the interrelationship of the layers with the platform.

Figure 6-1 First tier – Context view – Layers and their interrelationship with a Bimodal IT platform

6.3.2. Second and third tier of the platform

The section "Pace-layered Application Strategy and Digital Transformation as an alternative for Bimodal IT" in chapter 5.3.1 describes that certain system categories can be used to explain and allocate the various IT systems within the layers of a Bimodal IT platform. The underlying reason is that each environment supports a different goal and comprises its own set of architectural principles. Through table 5-18, it becomes clear that the system categories Systems of Records, Systems of Differentiation, and the Operational value-adding and Stabilized Systems of Innovation are part of the Stable Layer. While table 5-19 determines that the category Systems of Innovation is an integral part of the Explorative Layer. Both tables also disclose that the Stable Layer and Explorative Layer hold and support multiple groups of IT systems to realize and exploit the digitization and digitalization strategies and activities of a specific organization. For instance, the Systems of Record comprises all the business- and mission-critical IT systems, while the Operational value-adding and Stabilized Systems of Innovation cover the value-adding business applications and (internal) APIs to leverage and improve the information need, responsiveness, and efficiency of an organization. In contrast, chapter 5.3.2 discloses that the Communication Layer comprises one or more service-based environments to realize, facilitate, and guarantee the connectivity and interoperability between and within the layers.

Therefore, the layers, associated system categories (Systems of Record, Systems of Differentiation, Operational value-adding and Stabilized Systems of Innovation, and Systems of Innovation), and the related service-based environments (SOA, CIP, and MSA) can be identified as the main components of a Bimodal IT platform.

Also, the system categories Systems of Record and Systems of Differentiation are combined into one category within the second tier of the RA. The reason for this aggregation is based on the retrieved findings from the text data. Table 5-18 reveals that the Systems of Differentiation mostly cover Software as a Service (SaaS) solutions that extend the existing business processes and are used as a source for some of the business-critical objects and data. The SaaS solutions Salesforce, SAP C4C, Dynamics 365 Supply Chain Management, and SAP Simple Finance are some of these examples.

In the third tier of the RA, it is possible to group and model the various IT systems. The modeling of the several groups of IT systems can be done in multiple ways because each IT landscape is different and is capable of holding an extensive number and diverse range of IT systems. Also, in this research, the terms "group of IT systems," "groups," and "domains" will be used interchangeably to indicate identical or similar types of IT systems that can be grouped together.

The modeling of the groups themselves can happen, for example, through generic system domains, such as "CRM Software," "SCM Software," "HRM Software," etc. In this scenario, each system category of the second tier will enclose the same groups of IT systems. The groups, in their turn, can only capture IT systems that are relevant within a specific environment. By taking the SCM IT systems from table 5-18 as an example, the category Systems of Records and Systems of Differentiation can capture the IT systems SAP Ariba and Oracle SCM in the group “SCM Software.” While the category Operational value-adding and Stabilized Systems of Innovation can capture the IT system SAP Ariba Mobile App in the group “SCM Software.”

Another option is to model one specific group of IT systems within the category Systems of Records and Systems of Differentiation that combines several capabilities, such as CRM, HRM, SCM, PLM, MES, and Finance and Billing. This group of IT systems, for example, called “Business Software,” will enable the organization to capture all the relevant IT systems, including the IT systems SAP Ariba and Oracle SCM. In contrast, the IT system SAP Ariba Mobile App can be captured in a new group, for example, called “Field Service and GIS Apps” within the category Operational value-adding and Stabilized Systems of Innovation.

Due to the dynamic set-up of the RA, organizations are able to create unique and meaningful groups of IT systems and allow them to capture and model the implemented IT systems within the RA. Also, when a particular IT system is modeled within a specific group, it needs to comply to the layers’ characteristics, as mentioned in tables 5-16 and 5-17. If this is not the case, the particular IT system cannot be placed or plotted within that particular environment and group.

Figure 6-2 and 6-3 illustrate the main building blocks of a Bimodal IT platform. Tables 5-16 and 5-17 provide the characteristics of the Stable and Explorative domains, which can be used to model and allocate the different groups of IT systems within the second tier.

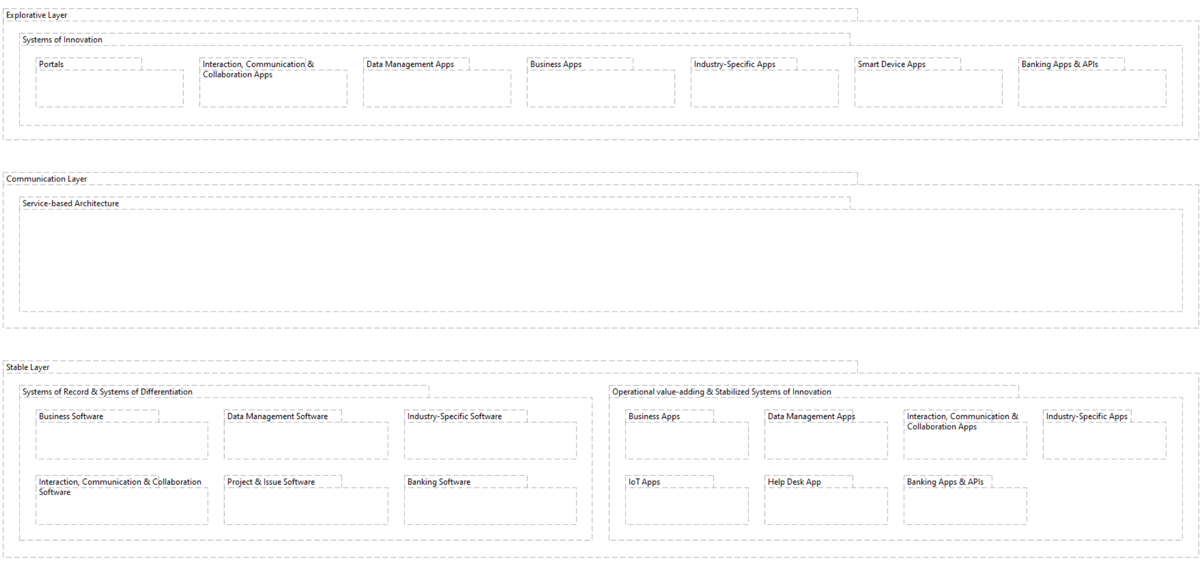

Figure 6-2 Second and Third tier – Context view – Layers, related system categories, and potential groups of IT systems

The modeled groups in figure 6-2 are derived and based on the generic system domains indicated in tables 5-18 and 5-19. These groups are taken as an example within the model and express how the various groups of IT systems can be modeled within the second tier of the RA. However, as explained, the third tier is not limited to these particular groups of IT systems.

Figure 6-3 Second and Third tier – Context view – Communication Layer and its associated service-based environments

6.3.3. Fourth tier of the platform – Stable Layer and Explorative Layer

As indicated in the previous section, each group of IT systems can contain multiple IT systems. The IT systems themselves, however, can be only part of one system category at a time. For instance, when a Salesforce system is being implemented, it can be plotted in one of the groups of the category Systems of Innovation. However, when the purpose of the IT system changes, such as it becomes the source for the customer data, it needs to move towards a group in the category Systems of Record and Systems of Differentiation because the Salesforce system starts to comply to one or more criteria of that particular category and related layer. In this scenario, the Salesforce system could be added into the group "Business Software" of the system category Systems of Record and Systems of Differentiation.

When looking from a functional or capability perspective, the Salesforce system can be divided and modeled in different groups of IT systems after the go-live. The capabilities Sales, Service, and Marketing can be added in the group "Business Software," whereas the Social and Telephony capabilities can be added in the group "Interaction, Communication, and Collaboration Software" of the category Systems of Record and Systems of Differentiation.

Another possibility is to model the IT systems as different types of IT systems. For example, an ERP system, such as SAP S4/HANA, can be broken down into different types of IT systems due to its modules. Each module, on his turn, could be embedded in a different group of IT systems.

Due to the dynamic set-up of the RA, the IT systems can be modeled in multiple ways.

Model regarding the different kind of IT systems within a Stable Layer

In general, for the system category Systems of Record and Systems of Differentiation applies that it comprises all the business- and mission-critical IT systems and all the SaaS-related solutions. This is in line with the data retrieved from table 5-18 because the data discloses that the majority of the indicated IT systems capture or store business-critical objects and data of a specific organization. Therefore, all the IT systems embedded in this layer comply with one or more criteria mentioned in table 5-16.

In addition, chapter 5.3.1 also indicates that the Stable Layer includes the system category Operational value-adding and Stabilized Systems of Innovation. This category encloses all the operational value-adding business applications and (internal) APIs, such as a Safety App, a Chatbot, a Track and Trace API**,** and IoT solutions to simplify and improve the operations.

Figure 6-4 illustrates how different types of IT systems are embedded within the Stable Layer. It also shows how the several types of IT systems are related to the system categories and the associated groups of IT systems. The groups, types, and IT systems themselves are not limited to the examples provided within the model.

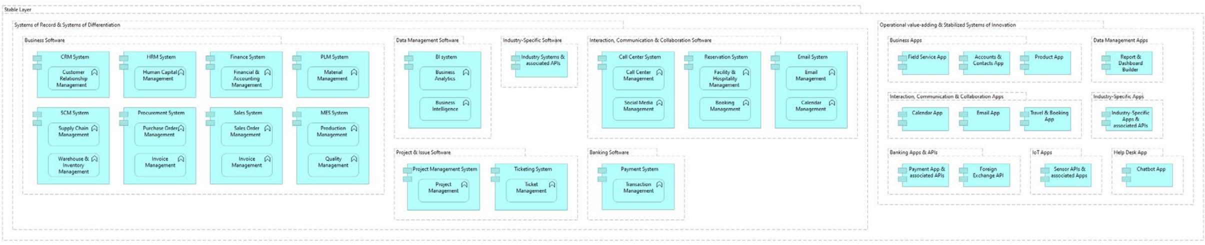

Figure 6-4 Fourth tier – Detailed view – Stable Layer comprising multiple types of IT systems and their related groups

Model regarding the integration patterns within a Stable Layer

Also, all the IT systems embedded within the Stable Layer hold and expose different kinds of functionalities, objects, and data. The IT systems themselves can be connected through multiple integration patterns to help to achieve the goals and to execute the business processes within an organization. These integration patterns can also enable an organization to take advantage of unified data models and to explore with multiple APIs.

Chapter 5.3.1 describes that the integration and interoperability between the IT systems embedded within the Stable Layer can only be achieved through multiple integration patterns. These integration patterns are not limited to manual integration, point-to-point integration, international standards for data exchange, EAI, or an ESB. The different mechanisms allow the IT systems to expose their functionalities, objects, and data to the other IT systems within the layer. To connect the IT systems of the category Systems of Record and Systems of Differentiation with the Explorative Layer or with the IT systems embedded within the category Operational value-adding and Stabilized Systems of Innovation, the patterns can make use of Remote Procedural Calls (RPCs), Simple Object Access Protocol (SOAP), OData, or other standards. In these cases, the relevant objects and data are exposed and wrapped into a particular message, proxy, or a gateway service and send towards a service-based environment within the Communication Layer. The big software vendors also offer out-of-the-box adapters to connect their IT products directly with a service-based environment within the Communication Layer.

Figure 6-5 portrays the common integration patterns within a Stable Layer. These patterns are in line with chapter 5.3.1.

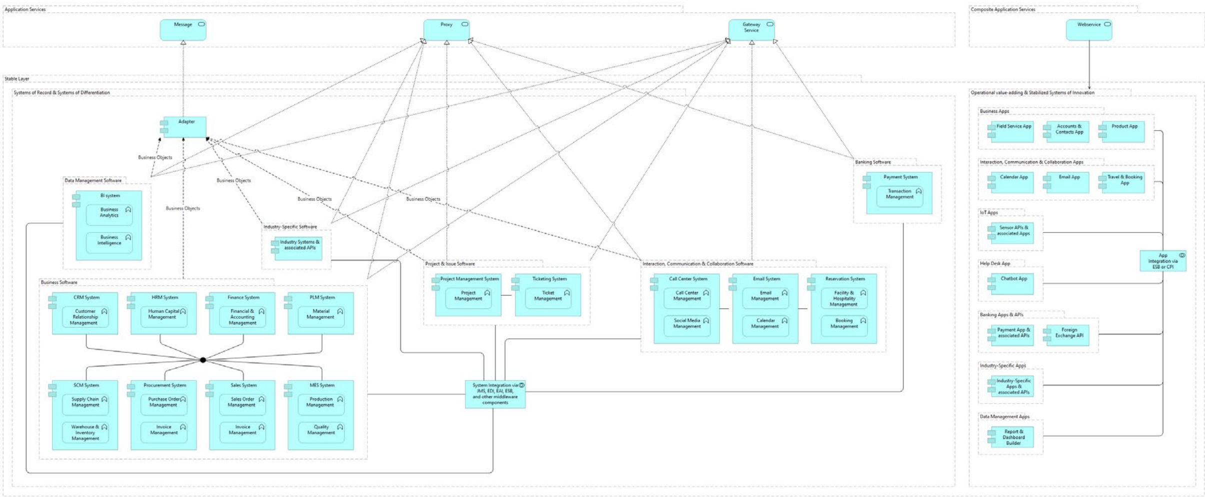

Figure 6-5 Fourth tier – Detailed view – Stable Layer comprising multiple types of IT systems and their associated integration patterns

Figure 6-17 presents a better view on how the different system categories are connected through the use of a service-based environment.

Model regarding the different kind of IT systems within an Explorative Layer

The Explorative Layer covers only one system category, namely the Systems of Innovation. All the embedded IT systems are focused on rapid delivery and fast releases because they must be able to cope with rapidly changing business requirements and new opportunities. All the IT systems embedded in this layer comply to one or more criteria mentioned in table 5-17.

Figure 6-6 illustrates how some of the IT systems mentioned in table 5-19 are embedded within the Explorative Layer. The figure also shows how the different types of IT systems are related to the system category and the associated groups of IT systems. For the modeled groups, types, and IT systems also apply that they are not limited to the examples provided within the model.

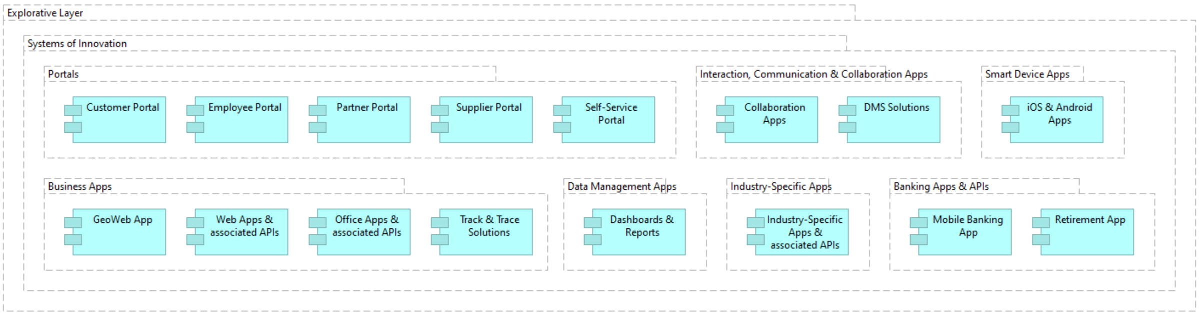

Figure 6-6 Fourth tier – Detailed view – Explorative Layer comprising multiple types of IT systems and their related groups

Model regarding the integration patterns within an Explorative Layer

As indicated in chapter 5.3.1, the Explorative Layer also has multiple integration patterns in place to establish the connectivity and integration between the various IT systems. The associated integration patterns themselves, however, are mainly based on microservices.

Figure 6-7 discloses the main integration patterns within an Explorative Layer. These patterns are also in line with chapter 5.3.1.

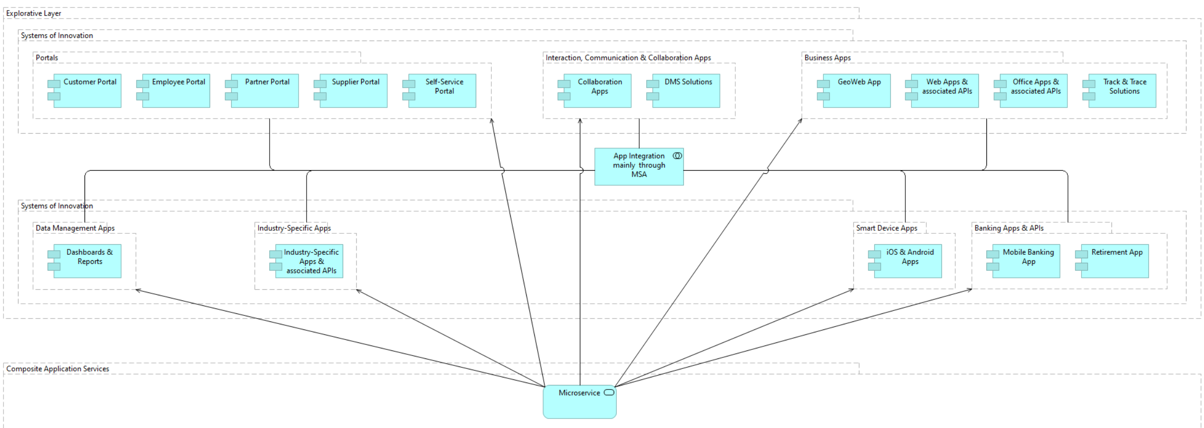

Figure 6-7 Fourth tier – Detailed view – Explorative Layer comprising multiple types of IT systems and their associated integration patterns

6.3.4. Fourth tier of the platform – Communication Layer

Chapters 2.4.2 and 5.2.2 indicate that the Communication Layer is the most fundamental layer of the platform. The layer enables and ensures the integration, connectivity, and interoperability between the different system categories, which are an integral part of the Stable Layer and Explorative Layer. As shown in figure 6-3, the layer itself can comprise in total three service-based environments. These service-based environments are:

- A SOA environment;
- A CIP environment;
- And a MSA environment.

Model regarding the components of an ESB solution

An ESB is used as a technical implementation to deliver SOA. Therefore, an ESB solution can be defined as the core component of a SOA environment.

Figure 6-8 discloses the main sub-components of an ESB solution. All the modeled sub-components are retrieved from the text data and can be found back in table 5-20.

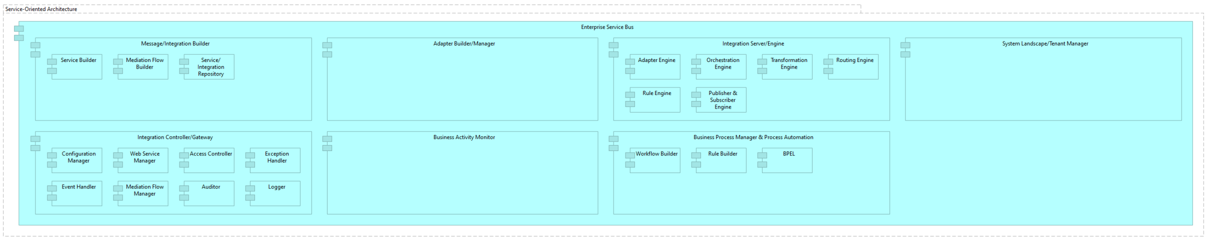

Figure 6-8 Fourth tier – Detailed view – Core components of an ESB solution

The modeled sub-components are derived from the ESB solutions “Microsoft Biztalk” and “SAP PI/PO” because, as per chapter 5.2.2, the majority of the selected cases are using these two IT systems to set-up the connectivity within their IT landscapes. Through the dynamic set-up, it is possible to enhance the model with additional sub-components, such as a separate Service Registry component if another ESB solution is used.

Model regarding the capabilities of an ESB solution

Figure 6-9 presents the capabilities of an ESB. The capabilities are based on the text data and retrieved from table 5-21.

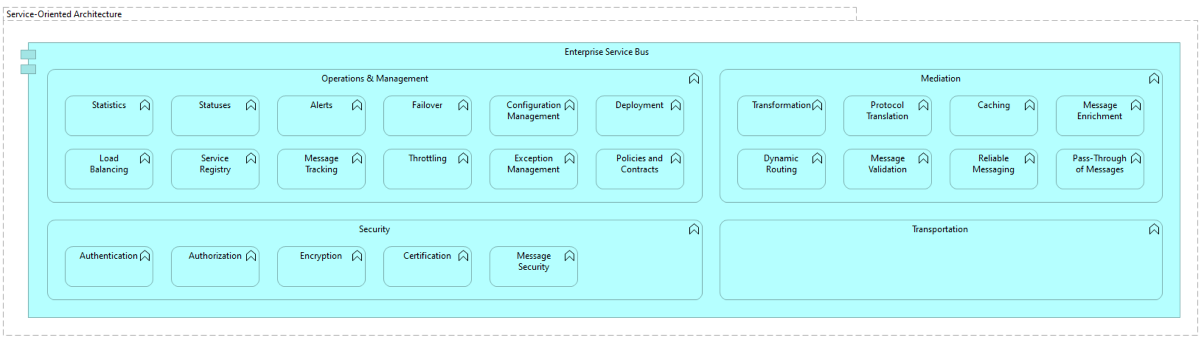

Figure 6-9 Fourth tier – Detailed view – Capabilities of an ESB solution

Within the model, a capability is mentioned, which is not indicated in table 5-21. The capability Policies and Contracts are retrieved from the literature instead of from the text data. Xiao et al. (2016) mention in chapter 2.3.3 that a Service Registry tool also enforces standards, policies, and contracts (Xiao et al., 2016). Since the Service Registry capabilities are an integral part and are already built-in within the ESB solutions “Microsoft Biztalk” and “SAP PI/PO,” this specific capability is also considered within the model.

Model regarding an ESB solution and its integration patterns within a SOA environment

As indicated in chapter 5.3.1, an ESB solution can be used as one of the middleware components to connect the various IT systems through services, mainly web services. The exposed functionalities, objects, and data from the IT systems in the category Systems of Record and Systems of Differentiation are wrapped into a message, proxy, or a gateway service and send towards the ESB. The ESB converts these formats into web services and executes the related mediation and orchestration activities and tasks.

Figure 6-10 portrays the integration patterns within a SOA environment.

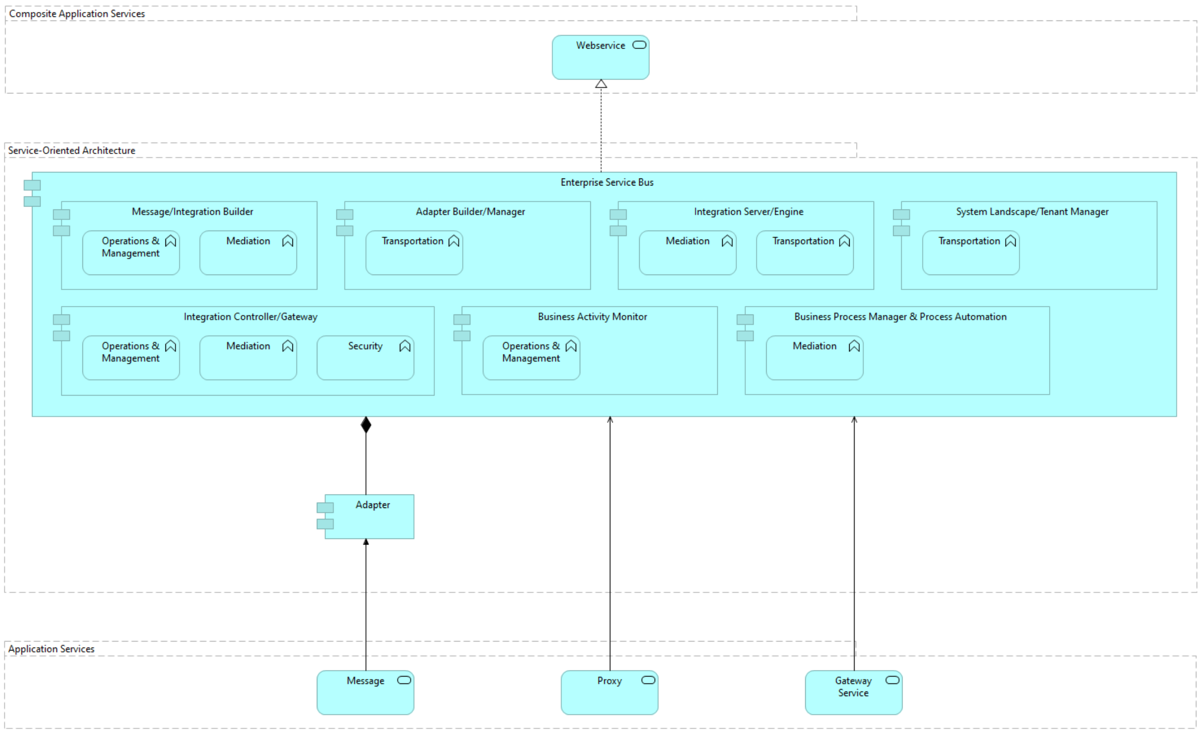

Figure 6-10 Fourth tier – Detailed view – ESB solution and its integration patterns within a SOA environment

The link between the sub-components and capabilities is derived from the available blogs and technical guides provided by the software vendors SAP and Microsoft on their ESB products (Brata De, 2014; Microsoft, 2020).

Model regarding the components of a CIP solution

A CIP environment can be realized through a CIP solution. Chapter 5.2.2 determines that “Azure Integration Services” and “SAP Cloud Platform Integration” are two famous CIP solutions to achieve and establish the connectivity and integration between on-premise and cloud software.

SAP Cloud Platform Integration combines Cloud Integration, API management, Open Connectors, and Integration Advisor functionalities to connect the on-premise and cloud-based solutions (SAP, 2020). The Cloud Integration functionality itself provides the ability to integrate SAP, non-SAP, cloud, and on-premise applications and processes through messages. API management provides access to simple, scalable, and secure digital assets through APIs and lets internal and external developers consume it. The Open Connectors functionality provides pre-built connectors and adapters to establish seamless connectivity between SAP and non-SAP applications. And the Integration Advisor accelerates the development of business-oriented interfaces, mappings, runtime artifacts and reduces integration efforts.

Azure has also released a single platform for the integration of on-premise and cloud-based solutions (Microsoft, 2018). The platform combines API Management, Logic Apps, Service Bus, and Event Grid functionalities. The Service Bus is used to connect the on-premise and cloud-based IT systems and services through highly secured messages. The Event Grid ensures the connection between Azure and third-party services through a fully managed event routing service, which is based on a publish-subscribe model. Logic Apps is used to create workflows and to orchestrate business processes. And the API Manager is used to publish the APIs. The APIs can be used by developers when connecting to back-end IT systems and services.

Both the CIP solutions are a successor to the preceding ESB solutions. Therefore, there are a lot of similarities between the sub-components of both IT systems. This is also in line with chapter 5.2.2.

Figure 6-11 presents the essential sub-components of a CIP solution. All the modeled sub-components are based on the text data and retrieved from table 5-22.

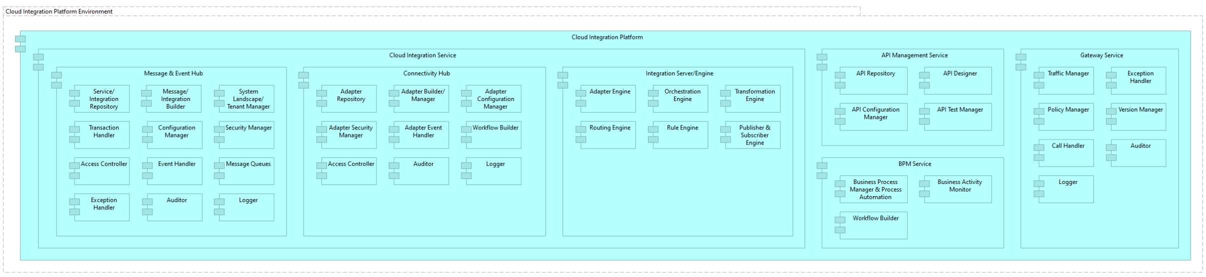

Figure 6-11 Fourth tier – Detailed view – Core components of a CIP solution

Model regarding the capabilities of a CIP solution

Figure 6-12 shows the capabilities of a CIP solution. All the capabilities are retrieved from table 5-23.

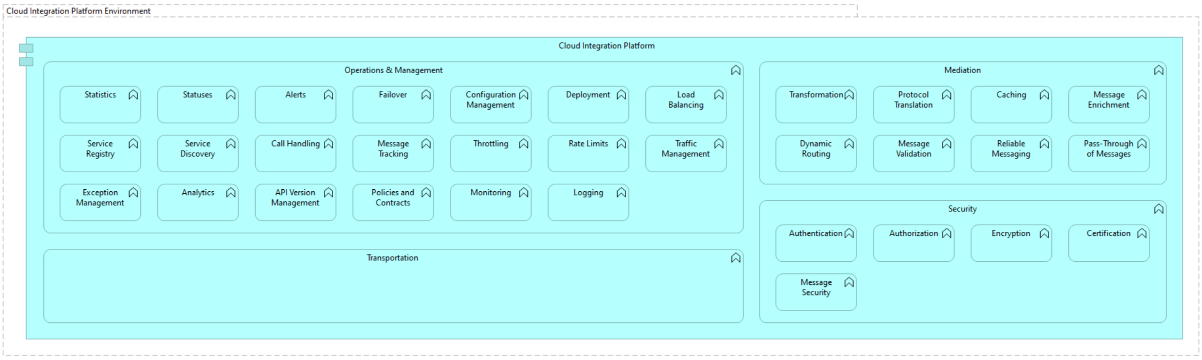

Figure 6-12 Fourth tier – Detailed view – Capabilities of a CIP solution

Model regarding a CIP solution and its integration patterns within a CIP environment

In essence, the CIP solutions fulfill the same purpose as the ESB solutions. Since the CIP solutions are a logical replacement for the ESB solutions, the CIP environment shares the same integration patterns as a SOA environment. Figure 6-13 illustrates the integration patterns within a CIP environment.

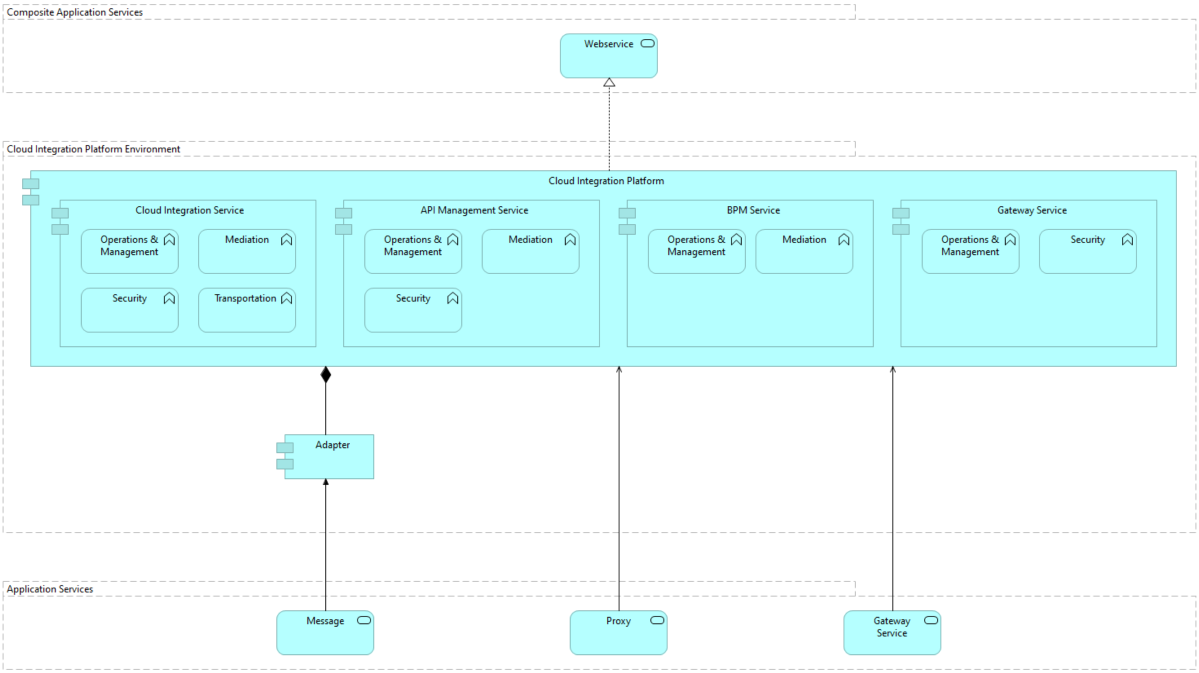

Figure 6-13 Fourth tier – Detailed view – CIP solution and its integration patterns within a CIP environment

Model regarding the components within a MSA environment

As described in chapter 2.3.4, a MSA is used to develop distributed and modularized application components, which are referred to as microservices (Xiao et al., 2016; Yale et al., 2016).

A best practice to deploy microservices is through the use of containers because this deployment model allows to automatically inherit all the benefits that containerization has to offer, such as portability, scalability, fault isolation, hardware efficiency, automation of installation, and ease of management (Jaramillo et al., 2016). Also, containerization leads to a faster development and more secure deployment of application components independently of the underlying infrastructure (Aderaldo et al., 2017).

The explanation and reason to use a Container Orchestration platform is described in chapter 2.3.4. In brief, it could be said that the platform is able to assist in the case of increased container volumes. It can help in the management and handling of individual containers.

Figure 6-14 discloses all the embedded core components within a MSA environment. The modeled sub-components of the Container Orchestration Platform are retrieved from table 5-24, while the essential sub-components of a container are retrieved from chapter 2.3.4.

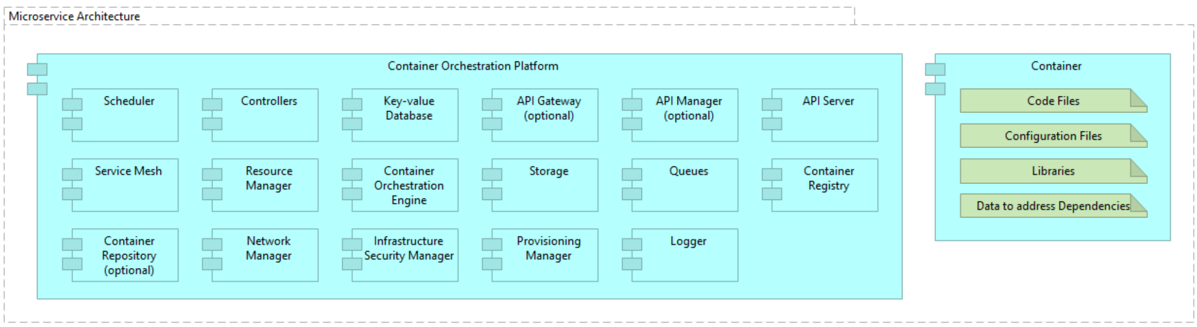

Figure 6-14 Fourth tier – Detailed view – Core components within a MSA environment

Some of the sub-components within the Container Orchestration Platform are indicated as optional. The reason is that these sub-components can be marked as double or excessive within the platform because there is already another sub-component available that provides similar functionality to manage and handle the containers. However, since at least four candidates indicate the sub-components, the elements are modeled and added into the RA.

Also, a sub-component has been added into the model, which is not mentioned in table 5-24. The sub-component Service Mesh is added into the model based on the researchers' experience. The reason is that a Container Orchestration Platform can use an API server and a Service Mesh to establish the communication between microservices.

As indicated in chapter 2.3.4, an API Gateway deals with the boundaries of a specific IT system, while the Service Mesh intercepts all the internal communication patterns between the microservices within a particular IT system (Clark, 2018a). In addition, without a Service Mesh, it is not possible to guarantee and ensure an acceptable level of performance (Chandramouli & Butcher, 2020).

Since the API Gateway and the Service Mesh fulfill different roles, they need to collaborate with each other within a Container Orchestration Platform to build sustainable IT systems.

Model regarding the deployment approach of a Container Orchestration Platform solution

Table 5-24 also mentions the components Nodes and Pods. From a deployment perspective, a Container Orchestration Platform is deployed through a cluster. The cluster itself consists of several nodes that represent a virtual or physical machine. Within the nodes, pods are hosted in which one or more containers are running. The view regarding the deployment of a Container Orchestration Platform is presented in appendix 6.

Model regarding the capabilities of a Cloud Orchestration Platform solution

Figure 6-15 presents the capabilities of a Cloud Orchestration solution. These capabilities are retrieved from table 5-25.

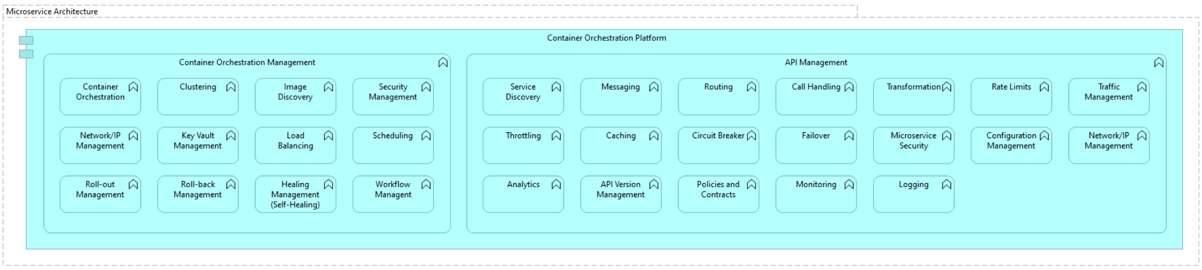

Figure 6-15 Fourth tier – Detailed view – Capabilities of a Container Orchestration Platform solution

Model regarding a Cloud Orchestration Platform solution and its integration patterns within a MSA environment

Chapter 5.2.2 indicates that a MSA environment is mainly used to realize and integrate IT systems within an Explorative Layer. To support this purpose, the platform needs to expose the received functionalities, objects, and data as microservices. Figure 6-16 illustrates how the platform and associated sub-components, capabilities, and patterns are embedded and used within a MSA environment to realize microservices.

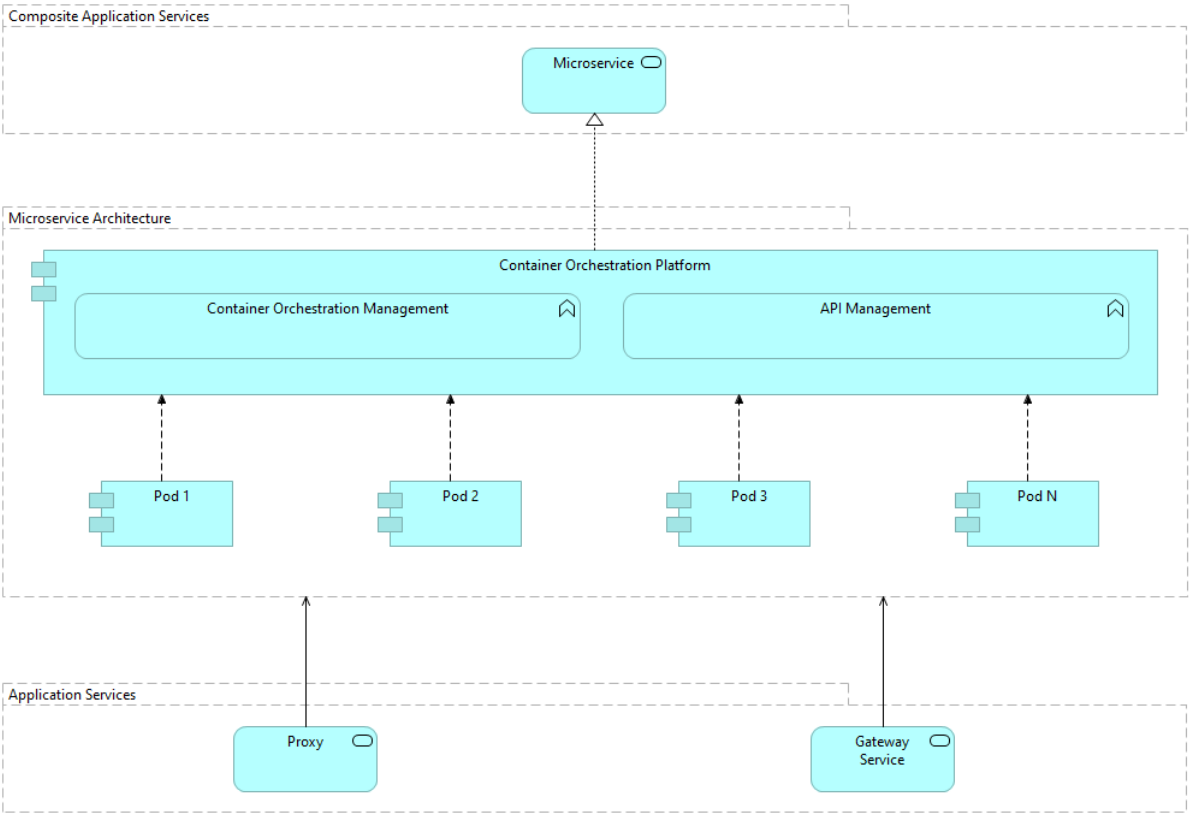

Figure 6-16 Fourth tier – Detailed view – Container Orchestration Platform solution and its integration patterns within a MSA environment

The model itself illustrates how the exposed functionalities, objects, and data from the IT systems in the category Systems of Record and Systems of Differentiation are wrapped into a proxy or a gateway service and send towards a Container Orchestration Platform. The platform converts the formats into a readable REST format and deploys the services into a container so that the RESTful APIs can be orchestrated, assembled, and exposed as microservices. These microservices, in their turn, are used to build and support fast-changing IT systems.

As per chapter 5.2.2, most of the selected cases are using “Microsoft Azure” within their IT landscape to establish a MSA environment. Therefore, the link between the sub-components and capabilities is derived from the available developer guides provided by Microsoft on its MSA product (Microsoft, 2019).

Model regarding the integration patterns between the Stable and Explorative layers

Figure 6-17 portrays the layers and its integration patterns on an abstract level. It illustrates how the four tiers and their elements are related to each other. Also, the view shows that an IT system created within the category Systems of Innovation can move to the category Operational value-adding and Stabilized Systems of Innovation within the Stable Layer when it reaches a certain level of stability or maturity. This is in line with chapter 5.3.1.

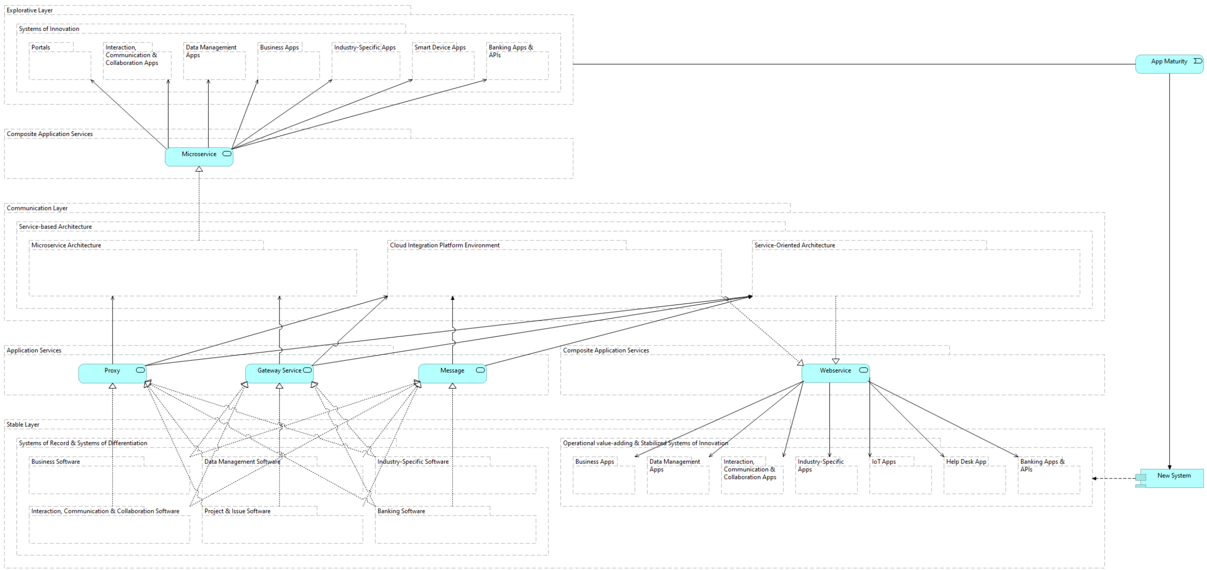

Figure 6-17 Fourth tier – Context view – Integration patterns between a Stable Layer and an Explorative Layer

6.3.5. Fifth tier of the platform – SDLC approaches and related tools

Based on the literature and the gathered text data, it becomes clear that the SDLC approaches and related tools to realize, deliver, and exploit the IT systems and services within the platform are also an important aspect to consider within the RA. To cope with the SDLC aspect, a fifth tier has been added into the RA.

The fifth tier is based on the elements that are captured and discussed in chapter 5.3.3. The chapter highlights that each environment of the platform uses another approach and tooling to develop and deploy the related increments. Therefore, it is important to distinguish and allocate the appropriate approaches and tools to the various environments within a Bimodal IT platform, so the different methodologies and their related tools can assist in the realization, delivery, and support of high-quality and sustainable IT systems and services.

The SDLC environments and related groups of IT systems mentioned within the fifth tier of the RA are prescriptive because, in the end, these elements determine how the IT systems and services are built and delivered within each environment of a Bimodal IT platform.

In total, three SDLC environments can be distinguished through chapter 5.3.3. These environments are:

- A traditional SDLC environment;
- An agile SDLC environment;
- And a separate CI/CD environment.

In addition, the fifth tier can also help to categorize and distinguish the IT systems into IT systems that support the platform and IT systems that are used by the platform to guarantee and ensure the operation and continuation of an organization. In essence, the project management, design, development, and deployment tools can also be embedded within the layers of a Bimodal IT platform. This visualization, however, can increase the complexity and make it difficult to model and understand the RA.

Model regarding the different kind of IT systems within a traditional and an agile SDLC environment

First of all, the traditional and agile SDLC environments are combined into one SDLC environment within the RA. This environment is named “Traditional/Agile SDLC environment.” The reason for the aggregation is that both environments use similar tools to realize, exploit, and maintain the IT systems and services within the various environments of a Bimodal IT platform.

The Traditional/Agile SDLC environment includes all the IT systems that are needed within a traditional or agile approach to guarantee and ensure the development and delivery of a single IT system or service. These tools also support and cover all the phases and steps associated with their SDLC approaches, such as planning, analysis, design, build, test, deployment, and monitoring.

Figure 6-18 shows some common tools that can be used to develop and deliver the related increments. All the modeled sub-components are based on the text data and retrieved from table 5-27.

Consequently, the obtained sub-components from table 5-27 are grouped into specific SDLC domains (groups of IT systems) within the model. These domains indicate the different types of IT systems that are used within the phases and steps of the associated SDLC approaches to develop and deliver a single increment. However, the modeling of several groups of IT systems can be done in multiple ways because each organization uses different tools for the development and delivery of an increment. Also, due to the SDLC domains, some of the modeled IT systems can overlap other project management, design, development, and deployment tools when looking at their capabilities and functionalities.

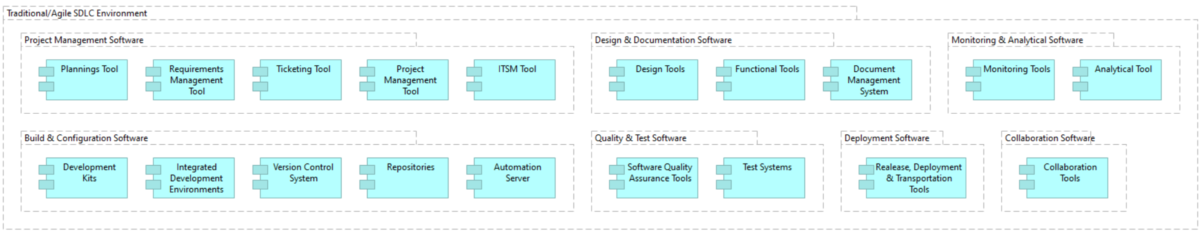

Figure 6-18 Fifth tier – Detailed view – SDLC environment comprising multiple types of IT systems and related groups to support the traditional and agile SDLC approaches

For the SDLC domains apply that these sub-components are minimally required within a Traditional/Agile SDLC environment. Due to the dynamic set-up and to accommodate the organizations, it is possible to rename or create corresponding SDLC domains within the RA. It is also possible to combine the groups or extend the view with additional groups of IT systems when certain sub-components are lacking within the model.

Model regarding the interaction of the tools and flow of an increment within a traditional and agile SDLC environment

Figure 6-19 illustrates the different groups of IT systems and how they interact with each other within the Traditional/Agile SDLC environment to deliver a single increment.

Regarding the deployment of a finished IT product, the model shows that an increment can be released to a Stable Layer or a service-based environment within the Communication Layer. These patterns are in line with chapter 5.3.3. This chapter indicates that the Stable Layer uses a mix of traditional and agile methods to deliver its increments. While the service-based environments SOA and CIP are using an agile approach to deliver the web services. The agile approach CI/CD is excluded from these approaches because this particular technique is primarily focused on the delivery of fast-changing IT systems and services.

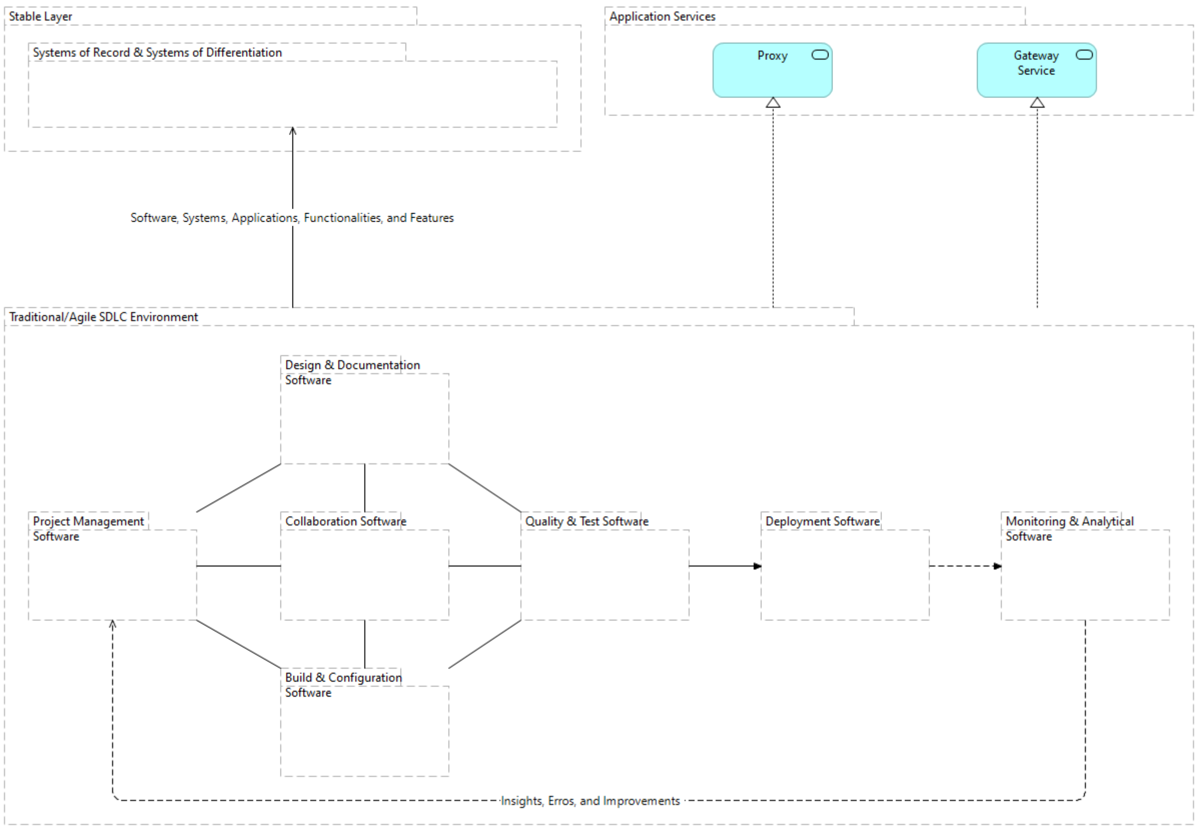

Figure 6-19 Fifth tier – Detailed view – Interaction of the tools and flow of an increment within a traditional or an agile SDLC environment

Also, when looking at the model, a flow can be determined. This flow implicates how an increment could be built, exploited, and maintained by using the tools in a particular sequence. Therefore, almost all of the IT product development flows can be traced back to the phases and steps that are used within a traditional or agile approach as described in the sections “Traditional Software Development” and “Agile Software Development” of chapter 2.2.4.

Model regarding the different kind of IT systems within a CI/CD environment

The CI/CD environment includes all the IT systems that are required to execute the CI/CD approach and to guarantee and ensure the development and delivery of a single IT system or service.

Figure 6-20 portrays the tools that are used during the development and delivery of the related increments. All the modeled sub-components are based on the text data and retrieved from table 5-28.

As described in the section “Detailed view regarding the components of a traditional and agile SDLC environment,” the tools retrieved from table 5-28 are also grouped into specific SDLC domains within the model.

For the model also applies that an organization can rename or add new groups of IT systems within the model when certain sub-components are missing.

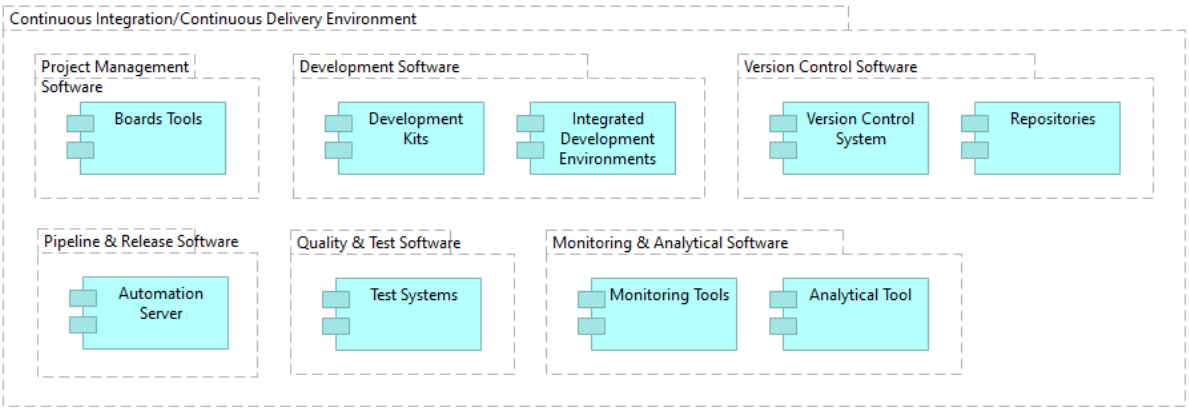

Figure 6-20 Fifth tier – Detailed view – CI/CD environment comprising multiple types of IT systems and related groups to support the CI/CD approach

Model regarding the interaction of the tools and flow of an increment within a CI/CD environment

Figure 6-21 illustrates the different groups of IT systems and how they interact with each other within the CI/CD environment to deliver a single increment. The figure also shows that a finished IT product can only be released through a container, which is in line with chapter 5.3.3.

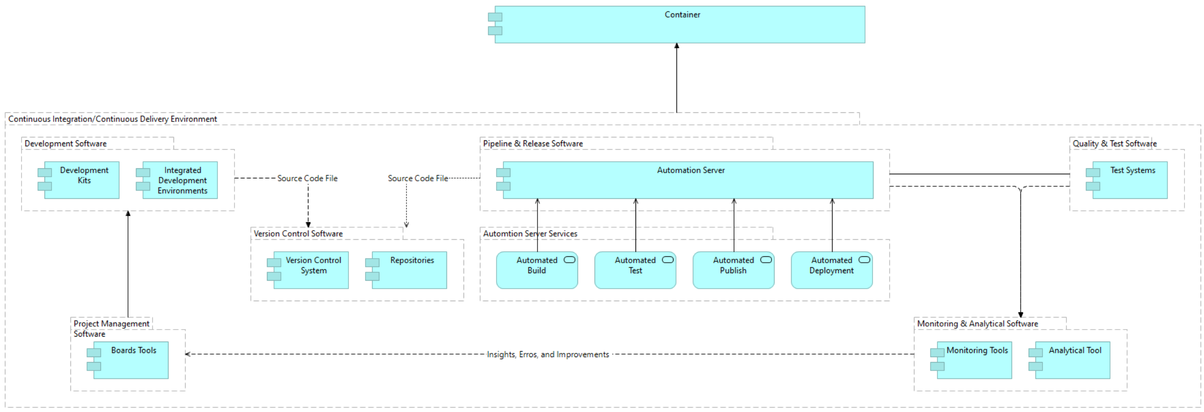

Figure 6-21 Fifth tier – Detailed view – Correlation of the tools to realize an increment within a CI/CD environment

Chapters 5.2.2 and 5.3.3 describe that the CI/CD technique is foremost used to realize and utilize IT systems and services within a MSA environment and the Explorative Layer because microservices are an enabler for fast-changing increments. Therefore, the flow shown in the model is prescriptive. It shows how an increment is built, exploited, and maintained throughout its lifecycle through the use of CI/CD. The particular steps associated with the flow can be found back in the sections “Continuous Integration,” “Continuous Delivery,” and “Continuous Monitoring” of chapter 2.2.4.

Model regarding the SDLC environments and their interrelationship with the Bimodal IT platform

Figure 6-22 presents on an abstract level how the SDLC environments are related to the layers of the Bimodal IT platform. The model illustrates each environment of the platform and indicates how and by which SDLC environment it is supported.

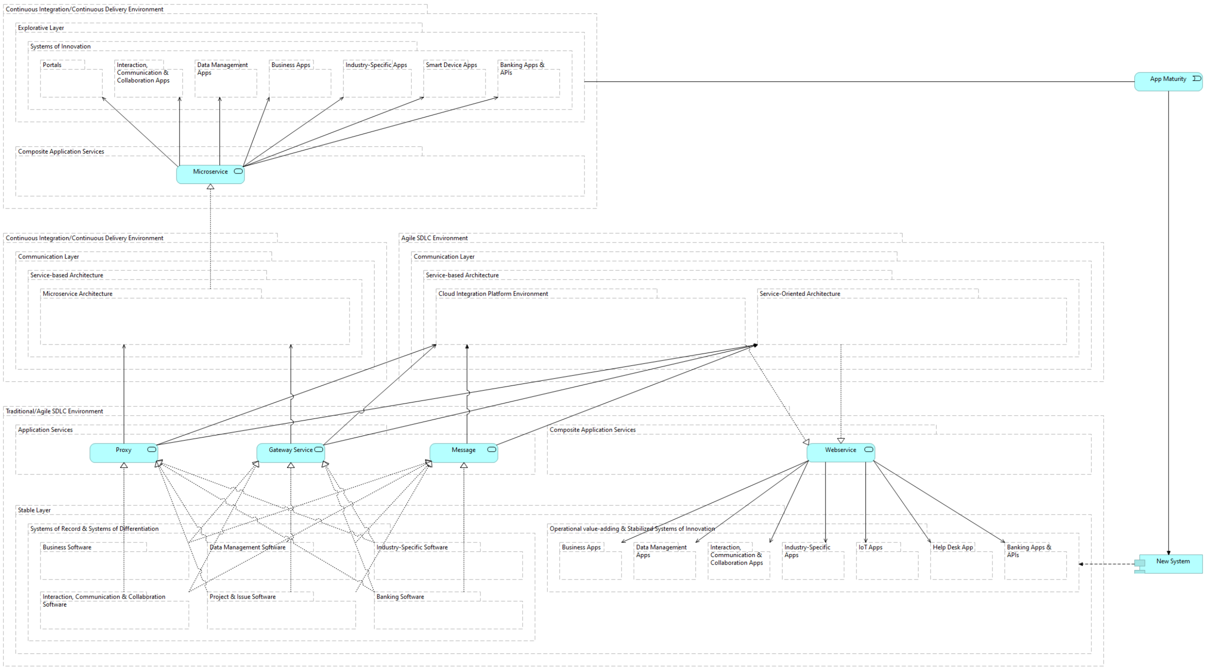

Figure 6-22 Fifth tier – Context view –SDLC environments and their interrelationship with the Bimodal IT platform

The model itself is derived and based on chapter 5.3.3.

6.3.6. Sixth tier of the platform – Deployment models and associated components

Another aspect that needs to be considered in the RA is related to hosting. In chapter 5.3.4, the gathered text data discloses that there is no significant relationship determinable between a specific deployment model and the layers within a Bimodal IT platform. However, every IT system within the platform is deployed somewhere. Therefore, hosting is a relevant subject. To cope with this aspect, a sixth tier has been added into the RA.

Model regarding an on-premise and data center environment

Chapter 5.3.4 determines that the on-premise and data center environments share the same components and capabilities to host a specific IT system. The only difference is that an on-premise environment is the responsibility of the organization itself. They are in control and own the servers. In contrast, a data center is a physical facility that offers computing and storage resources to an organization to host their IT systems and data (Cisco, 2019). To highlight the difference between the two environments within the RA, the server farm has been split up into independent servers within the on-premise model.

Appendix 7 illustrates the on-premise model, while appendix 8 portrays the data center view.

For the components regarding the on-premise and data center environments apply that they are based and retrieved from table 5-29. In contrast, all the capabilities are retrieved from table 5-30. Also, the figures visualize how the capabilities and components are integrated with each other. The models are based on the retrieved components and capabilities.

Model regarding a cloud environment

Appendix 9 presents a model regarding the cloud environment. The components of the platform are retrieved from table 5-31 and the capabilities from table 5-32. Also, the figure illustrates how the capabilities and components are integrated with each other. In addition, the view also presents through which subscription models a cloud environment can be offered. These specific subscription models are retrieved from table 5-33.

Model regarding the hosting environments and their interrelationship with the Bimodal IT platform

Figure 6-23 illustrates the hosting environments and how they are related to the layers of a Bimodal IT platform on an abstract level. The model itself is based on and derived from chapter 5.3.4.

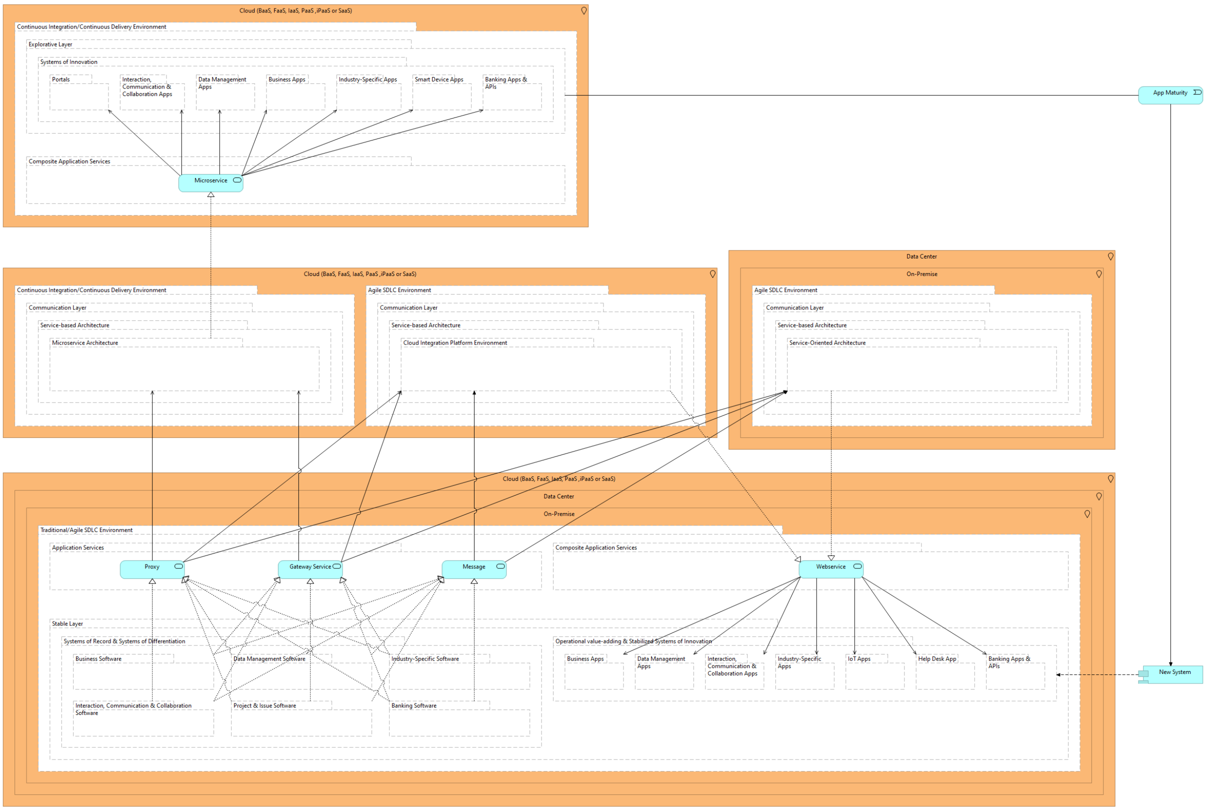

Figure 6-23 Sixth tier – Context view – Hosting environments and their interrelationship with the Bimodal IT platform

7. Expert feedback

This chapter presents the methodology for gathering the feedback and the associated results regarding the developed RA. It explains how the process is conducted and outlines the objective, approach, candidate selection, data collection, and result of the feedback.

7.1. Objective and approach for collecting the expert feedback

The objective of the expert feedback is to verify and justify the developed RA. The verification step can be performed through feedback on the completeness, accuracy, and recognizability of the platforms' core components. The feedback will also help to determine if the RA can fulfill its purpose, namely that the RA is suitable for the set-up, realization, or adoption of a Bimodal IT platform within an organization.

The approach for collecting the feedback will be based on interviews.

7.1.1. Goal of the interview and related feedback

The aim of the interview is to gather feedback and the participants' view on the developed models and views within the RA. To collect the data, the most important views of the RA will be sent to the participants. These models can help to verify if the elements are recognizable, complete and can be assessed as fundamental components of the specified environments within the platform. In essence, the goal of the feedback is to verify if the RA itself is complete, applicable, and able to fulfill its purpose. The feedback itself will also help to justify the RA.

7.2. Case and candidate selection for collecting the expert feedback

To conduct the feedback process, the study focuses on four organizations. Each of the selected organizations is treated as a separate case. Within each case, an Enterprise Architect or an Integration Architect will be asked to provide feedback on the developed RA.

7.2.1. Case selection process

The selection of the cases is done randomly. In the selection itself, however, there has been made a split between two examined organizations that were part of the multi-case study and two organizations that did not participate in the multi-case study. For the last two organizations apply that they need to be active within the consulting industry.

The distinction within the case selection process will help to collect accurate and trustworthy feedback. In addition, this approach can also help to increase the credibility and improve the RA.

Table 7-1 presents an overview of the selected cases. Due to privacy reasons and ethical considerations, the organizations and their candidates are anonymized. Each organization is characterized through a company ID, such as C1, C2, etc. And the participants are characterized via candidate 1, candidate 2, etc. The count regarding the company IDs and participants will continue from the case and candidate selection conducted in chapter 4.10.1.

7.3. Interview process for collecting the expert feedback

The interview process to collect the feedback includes the set-up of the interview conditions and questions. The interview format comprises an introduction section explaining the purpose and the problem statement of the research. This part is especially important for the organizations that did not take part in the multi-case study. The format also includes the aim of the feedback and comprises the conditions of the interview itself, such as the duration, confidentiality, the permission to record the interview, and the aim and usage of the gathered data.

7.3.1. Interview execution

The interview process started when a candidate agreed to participate in the feedback.

The first step was to plan and send over the appointment details in an email together with the information on the subject. The information sent included an attachment that contained the introduction, the problem statement, the aim of the interview and feedback, and some models of the RA. The attached models also contained a short explanation about the views and how to read them. The choice was made to send seven views, which included the context models of the second, fifth, and sixth tier and the detailed models regarding the components and capabilities of the CIP and MSA service-based environments.

One hour before the interview itself, some precautions were taken to meet the conditions as stated in the interview format. Once the interview started, a short introduction was given. During the introduction, the following elements were discussed:

- The purpose of the feedback;
- Terms and conditions of the interview;
- Duration and confidentiality;
- Permission to record the interview.

After this step, the interview started to collect the feedback.

In total, four interviews were conducted in this study to verify and justify the developed RA. The related cases and candidates are presented in table 7-1.

Company ID	Industry	Participant	Position/Role in the company	Role ID
C6	Postal	Candidate 16	Integration Architect	IA-02

C7	Products	Candidate 17	Enterprise Architect	EA-07
C10	Consulting	Candidate 18	Enterprise Architect	EA-08
C11	Consulting	Candidate 19	Integration Architect	IA-03

Table 7-1 Selected cases and their related candidates to collect the feedback on the RA

7.3.2. Interview questions

In total, six questions are prepared to collect the feedback from the candidates. Each interview comprises the same questions so that the interview is consistent and the gathered data can be analyzed and compared in a similar way. The interview is semi-structured and contains only open questions. This approach allows to collect as much feedback as possible from the participants. The interview questions to collect the feedback are stated in table 7-2.

Number	Questions to collect the feedback
1	What is your first impression when looking at the Reference Architecture?
2	What is your impression when looking at the different layers and components? Do you recognize those, or do you miss any elements?
3	Looking at the Reference Architecture and the experiences in your current or past projects, in which environments/projects do you see a match? Which environments/projects would benefit from this Reference Architecture?
4	Can the Reference Architecture be helpful or even act as a framework for the current architectural views and models? For example, to create better views and models because the Reference Architecture can help structure and visualize the IT landscape?
5	Where do you see advantages of this model? Where do you see potential challenges or drawbacks?
6	Any other remarks/thoughts you would like to share?

Table 7-2 Interview questions to collect feedback on the developed RA

7.4. Data analysis regarding the feedback

To analyze the feedback, the researcher will make use of the conventional content analysis and summative content analysis methods. The analysis itself will try to provide an answer on the recognizability and applicability of the RA. Regarding the content of the models and their elements apply that the models themselves are based on a solid analysis. Therefore, it is unnecessary to perform a verification step on every content within the RA.

7.5. Feedback result

This section describes the conducted feedback analysis and presents the results that are retrieved from the feedback interviews.

First of all, all the interviews are transcribed and coded. The responses on each question are labeled with one or more keys (#-tag). Through this approach, every response is grouped into specific themes. Subsequently, the findings are classified within the feedback data.

During the conventional content analysis, 27 unique keys are created to identify the themes and to group the feedback data. The keys are presented in appendix 10.

The third step involved the creation of the codes. The codes themselves are derived from the keys. Since the keys are the link between the codes and the feedback data, they can be used to trace the processed results back to the original feedback interview transcripts. In total, 26 codes are created. The codes are presented in appendix 11. After the identification of the codes, the codes are grouped into topics and quantified by counting how often the codes are mentioned within the feedback interviews.

For the data analysis on the feedback also applies that the related codes (key findings) are combined, grouped, and divided into five clusters. Therefore, the correlations between the clusters, topics, keys, and codes are presented in appendix 12. The clusters themselves are also shown in figure 7-1 and discussed in the upcoming paragraphs. This approach ensures that the collected content, conduced analysis, and context is clear, understandable, and readable.

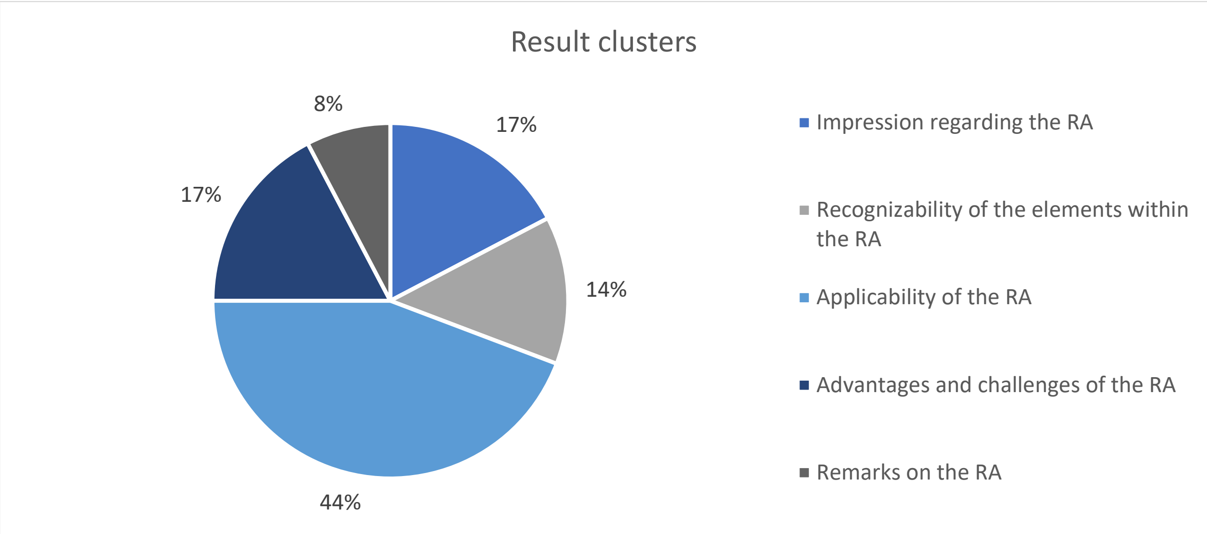

Figure 7-1 Result clusters based on the feedback data analysis

The over-all count of the 26 codes resulted in 52 responses. By plotting the responses over the five result clusters shows that nine responses can be allocated to the cluster "Impression regarding the RA," seven responses to the cluster "Recognizability of the elements within the RA," 23 responses to the cluster "Applicability of the RA," nine responses to the cluster "Advantages and challenges of the RA," and four responses to the cluster "Remarks on the RA." The distribution of the responses per cluster is also included in figure 7-1.

7.5.1. Impression regarding the Reference Architecture

Figure 7-2 complements the data analysis and presents the codes from the feedback data to discuss the opinions and thoughts of the candidates regarding the RA.

The Reference Architecture looks complete and is clear

In total, three candidates (candidates 16, 18, and 19) mention that the RA is clear. They understand the goal and purpose of the RA. Consequently, candidate 16 indicates that the RA looks complete by reflecting on the question if any essential elements or patterns are missing within the RA.

The Reference Architecture provides a proposal and guidelines to model and visualize an IT landscape

Regarding the goal of the RA, three candidates (candidates 17, 18, and 19) underline that the RA provides a good proposal on how an IT landscape could be visualized. Candidate 19 also appreciates the RA because it can showcase how all the IT systems are embedded within an architecture and disclose how they interact with each other.

In addition, candidate 18 even compared the RA with current views and indicated that the RA provides a decent guideline on modeling an IT landscape. Another conception of the candidate was that the RA could visualize a Bimodal IT environment by capturing an organizations' entire IT landscape and fitting it into the model.

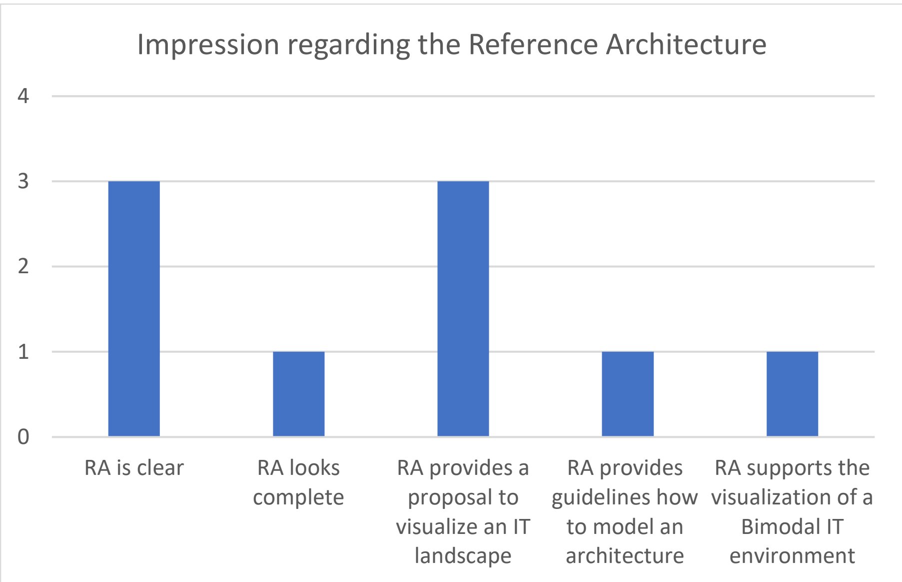

Figure 7-2 Result overview on the impressions regarding the RA

7.5.2. Recognizability of the elements within the Reference Architecture

Figure 7-3 complements the data analysis regarding the recognizability. It presents the codes to indicate if the RA is complete and fully recognizable by the candidates.

All the modeled elements are recognizable

All the candidates recognize the elements and patterns within the RA. Regarding the lack or missing components, the data shows that the candidates do not miss any crucial elements within the provided models of the RA.

It is important to explain some of the essential elements

Two candidates (candidates 16 and 18) indicate that certain components mentioned within the RA should be explained a bit more. Especially when the components are important or even essential to realize a certain pattern or environment. Both candidates provided the components Service Mesh and API Gateway as an example. The candidates discuss that many people do not know the difference between these two components and, therefore, can lead to a scenario in which a stakeholder is going to choose between one of the two components when implementing a Container Orchestration Platform. However, this should not be the case because, at the core, both components are important when establishing the communication patterns within a MSA environment.

The concept of Bimodal IT also implicates an architecture

As indicated in chapter 1.2, the concept of Bimodal IT implies more than only a business transformation. During the verification of the RA, this argument also has been recognized and acknowledged by one of the candidates. According to candidate 18, the RA likes to disclose that Bimodal IT is not only a concept but also an architecture. The candidate indicates that this angle of approach towards the concept of Bimodal IT "is a very interesting thought."

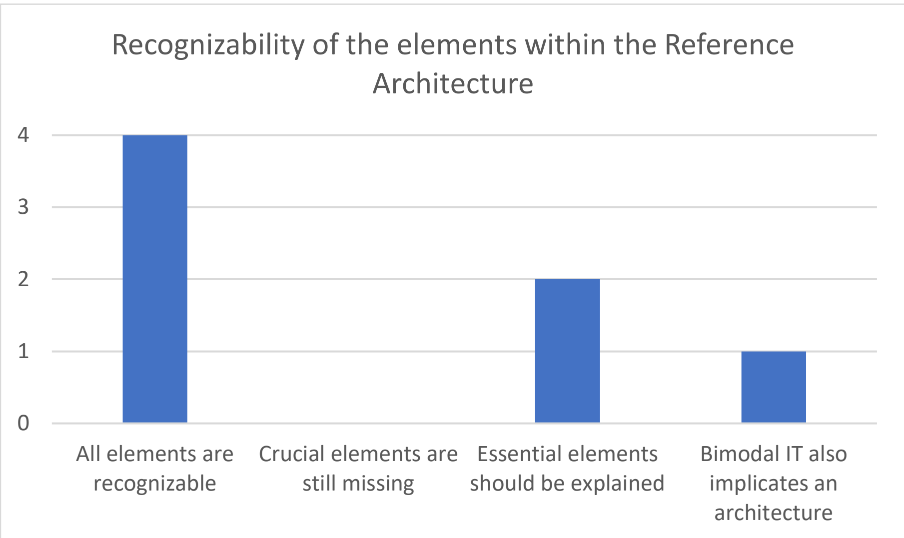

Figure 7-3 Result overview on the recognizability of the modeled elements

7.5.3. Applicability of the Reference Architecture

Figure 7-4 complements the data analysis regarding the applicability of the RA. It presents the codes from the feedback data to describe if the RA can be used to model an IT landscape.

The Reference Architecture is applicable and can help to visualize (hidden) patterns

All the candidates indicate that the RA is applicable. Candidate 17 underlines that the RA can help them to visualize the IT landscape in another way and to discover hidden patterns, while candidate 16 states that they “have created multiple Track and Trace functionalities. One API is created for the internal customer care agents, while the other two APIs and associated web apps are created for the customers to trace their parcels. In this case, the APIs and associated web apps, back-end IT systems, and integration patterns could be modeled through the Reference Architecture. Ultimately, I believe that the provided Reference Architecture can be used in any IT project to expose the communication patterns between the front-end and back-end IT systems.”

Despite the agreement that the RA can visualize all the patterns within an architecture, candidates 16 and 19 raise the concern that the integration patterns are not solely based on services. Both candidates indicate that occasionally the connection between the Stable Layer and Explorative Layer or the system categories Systems of Record and Systems of Differentiation and Operational value-adding and Stabilized Systems of Innovation are established through SFTP or point-to-point integration. The reason can be explained by the comment provided by candidate 19. The candidate indicates that in some cases, “a web app is connected directly, through another mechanism instead of services, with the back-end because services are not appropriate for certain scenarios. An example can be a Mendix App that is connected through SFTP with a DMS. Or certain critical applications that require a point-to-point integration with the back-end systems. If these connections are realized through services, the change exists that the services could expose critical data outside the organizational boundaries, which can have a big impact if it ends up on the streets, like intelligence data or personal data.”

The Reference Architecture can support in the creation of better views and in the set-up of a proper architecture

Candidates 17, 18, and 19 disclose that the RA can be used as a baseline to visualize an IT landscape. They also indicate that the RA can help them differentiate the implemented IT systems and subsequently allocate each IT system into one of the prescribed system categories, which are an integral part of the layers within a Bimodal IT platform.

Consequently, candidates 16, 17, and 19 discuss that the RA can support them in creating better views or complementing the existing views. In the case of candidate 19, the RA can be quite helpful because most of the current models lack the integration components and patterns. Whereas candidate 17 argues the following: "We already have multiple diagrams regarding our IT landscape and systems. By adding the details presented in your Reference Architecture, we can create a new dimension into our own models and views."

On top of the visualization, two candidates (candidates 18 and 19) discuss that the RA can assist them in the set-up of a proper architecture. And in the case of candidate 18, the RA can even help to set-up a Bimodal IT platform within the IT landscape of a particular organization.

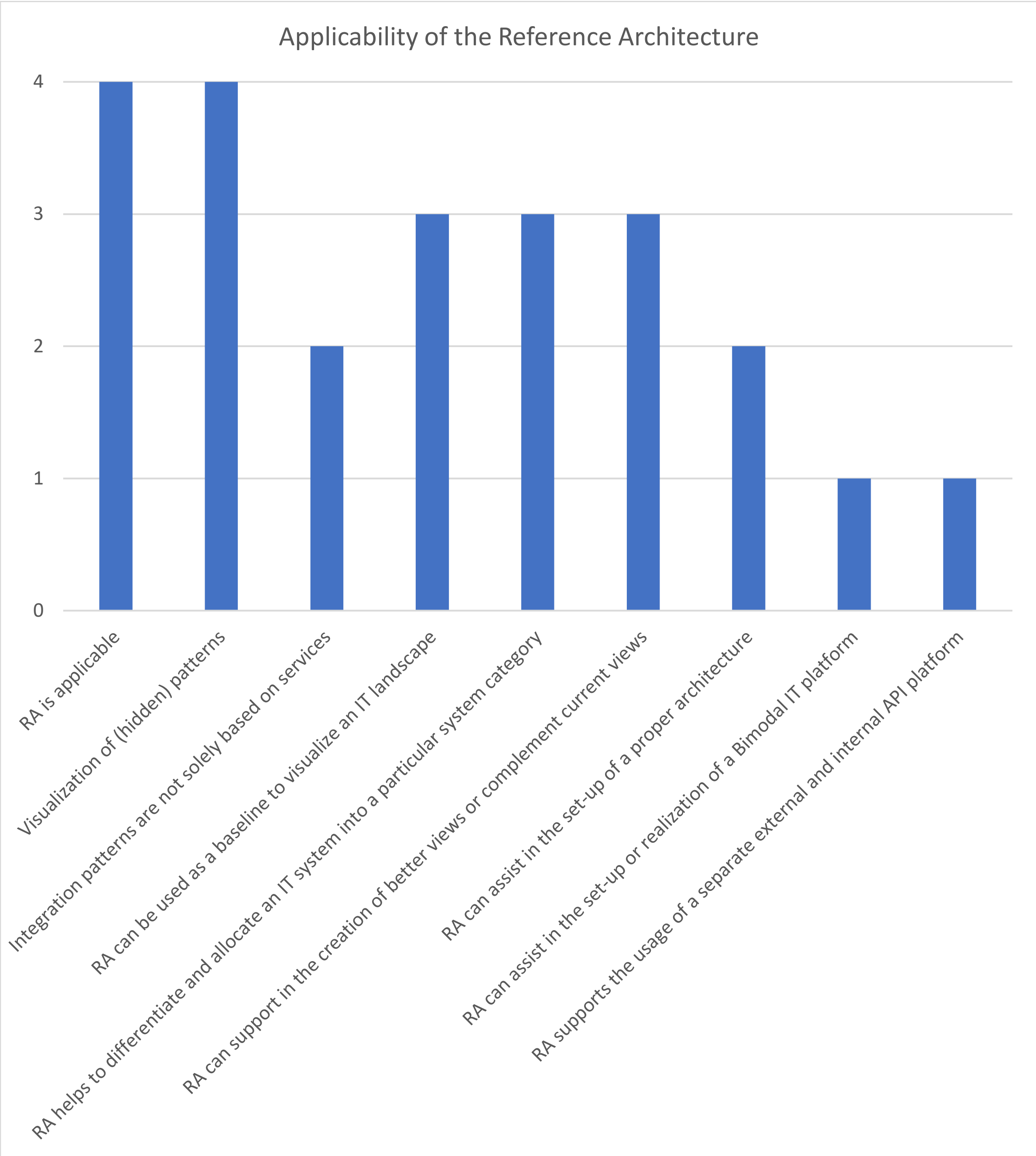

Figure 7-4 Result overview on the applicability of the RA

The Reference Architecture supports the usage of a separate external and internal API platform

Candidate 19 mentions that the RA will force organizations to rethink their integration patterns for each IT system. Also, the RA facilitates the use of an external and internal API platform within an IT landscape to establish and leverage the various connections and IT systems.

7.5.4. Advantages and challenges of the Reference Architecture

Figure 7-5 complements the data analysis regarding the advantages and challenges of the RA and presents the codes to describe the potential impact that the RA can have on an organization.

The Reference Architecture can support in the analysis, decision making, and discussions

Three candidates (candidates 16, 18, and 19) agree that the RA can be helpful in the analysis and discussions regarding the architecture. Candidate 18 argues, "the views can support in the analysis and discussion when implementing or removing a system because it can show the dependencies with other systems." According to candidate 17, the RA can even support in outsourcing decisions. The candidate indicates that the RA can help "to create different outsourcing strategies for each individual environment."

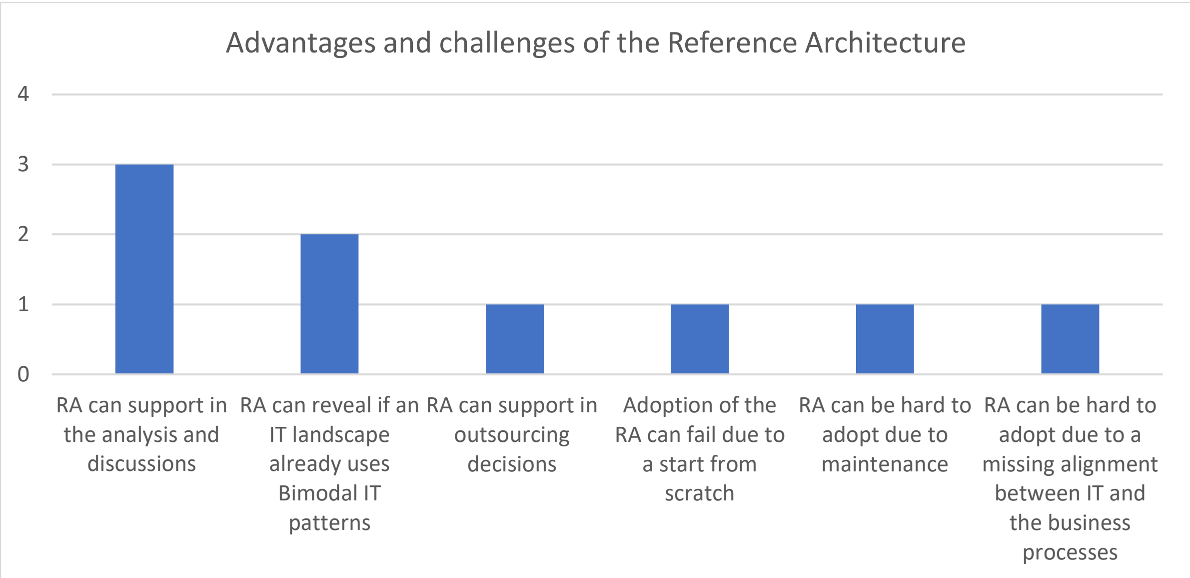

Figure 7-5 Result overview on the advantages and challenges of the RA

The Reference Architecture helps to indicate if an organization already makes use of patterns that relates to Bimodal IT

As discussed in chapter 1.3, the developed RA can be used as a reference to comply or even to realize a Bimodal IT environment. Candidates 17 and 18 support this argument. Candidate 17 indicates that the use of the RA can help them to model all the IT systems from their IT landscape and subsequently reveal if they are using patterns that can be related to a Bimodal IT environment.

Challenges regarding the adoption of the Reference Architecture

Two candidates (candidates 17 and 18) also indicate some challenges that are associated with the use or adoption of the RA. Candidate 17 discusses that they have to find an efficient way to embed the RA into their existing models without starting from scratch. And candidate 18 argues that the models need to be maintained on a frequent base because it captures and presents the entire IT landscape of an organization. Next to this concern, the candidate also sees a drawback for the RA because there is no alignment in place between the business processes and the IT landscape itself.

7.5.5. Remarks on the Reference Architecture

Figure 7-6 complements the data analysis by capturing the remarks from the candidates. It presents some additional comments from the candidates on how the RA can be improved.

Remarks

From the four candidates, only candidate 18 shared some remarks on the RA. The candidate suggests creating some additional models regarding the data flows and linking the RA models with the business processes of a particular organization. This activity can help to cope with the missing alignment and to provide a bigger picture regarding the collaboration between IT and the business.

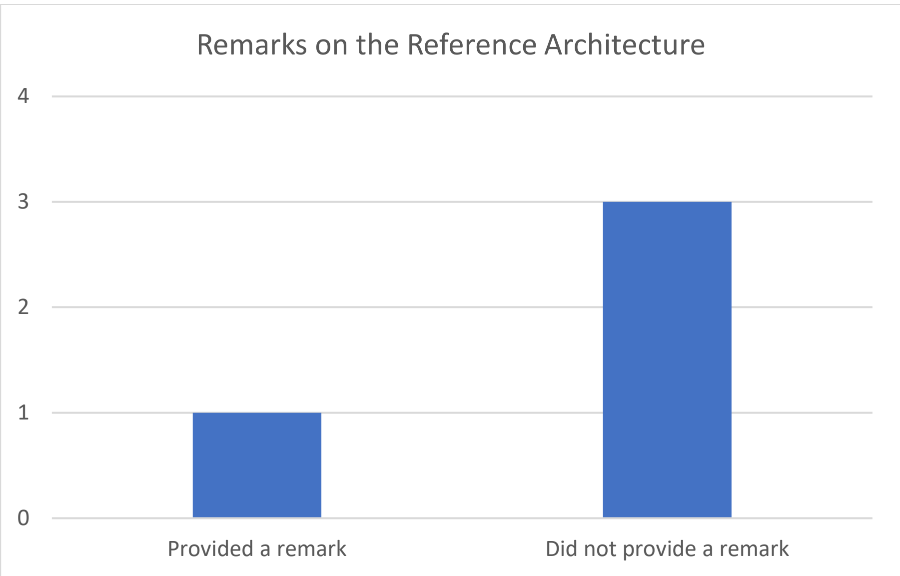

Figure 7-6 Result overview on remarks

8. Discussion

This chapter describes and discusses the developed RA and relates it to the existing literature, findings, and retrieved feedback. It presents a reflection on the RA itself and justifies why the research has been conducted.

The first section zooms into the developed RA and compares it with existing RA's. Subsequently, the RA is discussed on how it can support multiple concepts instead of focusing solely on the concept of Bimodal IT. Also, the applicability and the advantages of the RA are discussed in this section. The second sections present some hypothetical models that could complement the developed RA. The third section defines Bimodal Architecture, which is retrieved and based on the conducted research. The fourth section discusses the validity of the RA in comparison to the future. And the last section provides some best practices on how to implement and adopt the constructed RA.

8.1. Reference Architecture

8.1.1. The developed Reference Architecture in comparison to existing Reference Architectures

In essence, a RA is an abstract description of a product to describe or address a specific part of an architecture. The focus of many RA's is to provide a generic architecture with guidelines to address, describe, and implement the customer context, business architecture, data architecture, application architecture, and related technical architecture within a specific domain or to solve a particular problem (Cloutier et al., 2010; The Open Group, 2016).

The developed RA in this research has a broader scope and is different because it focuses on the entire IT landscape and specifies how the application and technical architecture of an organization should be set-up or arranged to cope with IT projects that enclose business-critical as well as digital (modular) IT systems. Conclusively, the RA addresses and explains how to arrange an architecture that delivers and supports increments that are subjected to different speeds.

Also, the developed RA in this research is built through the use of ArchiMate. ArchiMate is an open and independent modeling language used to describe, analyze, and visualize the business architecture, application architecture, and technical architecture (The Open Group, 2019). Many of the existing RA's do not follow a generic framework or are not constructed through a standard specification, such as ArchiMate or BlueDolphin.

Another important difference between the developed RA in this research and the current RA's is that the developed RA is descriptive and prescriptive. The reason is that some elements are an integral part of the Bimodal IT platform, whereas other elements are free configurable based on the organizational needs. In addition, without the three prescriptive layers, it is not possible to indicate an architecture as a Bimodal Architecture.

Due to all the mentioned reasons, the conclusion can be drawn that the developed RA within this research is different from existing RA's and can also contribute to the current body of knowledge.

8.1.2. How the Reference Architecture can assist multiple concepts in delivering agility

The study introduces several concepts that can be used to support organizations within their transformation journey. Chapters 2.1.1, 2.1.2, 2.1.3, 2.2.2, and 5.2.1 discuss that the concepts support an organization in its digitalization strategy and help to maintain and exploit the available legacy environment and its implemented IT systems. While comparing the several concepts in chapters 2.1.4, 2.2.3, and 5.3.1, it became clear that all the mentioned concepts deal with the same phenomenon. Therefore, the presumption can be made from a theoretical perspective that the concepts can share the same architecture. However, none of the concepts indicate or discuss how the underlying architecture should be arranged or set-up to support the various speeds of IT delivery.

The distinction between the concepts and the architecture is not surprising when looking at the problem from an organizational perspective. The common rule is that IT should facilitate and support the business despite its underlying architecture. This perspective also leads to another discussion, namely that Multi-Speed IT is a better approach than Bimodal IT. In the Multi-Speed approach, an IT project can be split-up into an unlimited number of modes. Consequently, for each modality or mode complies another regime and requires different integration patterns to connect the related IT systems to each other within the IT landscape to decrease negative influences or unwanted side effects.

As discussed in chapter 2.2.2, the Multi-Speed approach allows to blend various delivery approaches so that an organization can choose the proper operating and delivery method for each type of change, and at the same time, govern the strategic goals and objectives (Bayley & Shacklady, 2015; Ram, 2017). The modalities also influence how the increments within an IT project are released to production. This could be done simultaneously or separately after a specific increment is fully built, agreed upon, and accepted by the business.

From an architectural perspective, the IT landscape and its embedded IT systems are fully isolated in relation to the various concepts and modalities. Each IT system and its related increments are part of an IT landscape and, therefore, should be embedded somewhere within the existing architecture.

To help an organization become agile, the different concepts should also comprise the architectural patterns. Without considering the architecture, the concepts are hypothetically only able to change the organizational culture and prepare the related business on agility because, in many cases, the underlying IT landscape itself is not ready to cope with the different speeds of IT delivery. By using or adopting the developed RA, an organization will be able to set-up a proper architecture and support agility despite which concept or how many modalities it uses.

Within a Bimodal IT platform, visualized or realized through the RA, an IT system can be embedded within one of the two layers to cope with the required speed. Except if it comprises an IT system that implements an integration mechanism or supports in the development, as indicated in chapter 6.3.5. More specifically, an IT system can only be part of one system category at a time, as stated in chapter 6.3.3. The system categories can comprise several architectural principles but are an integral part of a layer and, therefore, comply with the associated layers' characteristics, as mentioned in table 5-16 or table 5-17. Due to the set-up of the architecture, the Bimodal IT platform itself does not discriminate how an IT project is delivered and implemented, but it helps to categorize where a specific IT system should be allocated within the architecture during its life cycle to meet the business needs, demands, and speed of delivery.

8.1.3. Potential name for the Reference Architecture

Another point of discussion can be related to the name of the developed RA. As indicated in chapter 1.5, the terms Bimodal IT, Bimodal IT platform, and Bimodal Architecture are used interchangeably within this study. However, the question arises if the terms Bimodal Architecture and Bimodal IT platform are a proper identification for the created RA within the research. The name "Three-layered Architecture to support dynamic IT landscapes" could potentially be more suitable because of the three identified layers, which are an integral part of the platform. Also, by providing the RA a different name, it could cope and be related more easily to other concepts as well, amongst the concept of Bimodal IT.

Despite the discussion regarding the name of the RA, the research will keep using the term Bimodal IT platform for the developed RA to prevent confusion. However, in future research, the mentioned name "Three-layered Architecture to support dynamic IT landscapes" will be used to identify the RA.

8.1.4. Applicability of the Reference Architecture

Commonly, a RA is used as a guideline. Also, in many cases, only specific parts of the RA are used to complement the existing views and models within an organization. However, regarding the applicability of the developed RA, the feedback is evident in chapter 7.5.3. The feedback determines that the RA itself is applicable. It can help create or complement existing views and models and support an organization with a baseline to visualize its entire IT landscape. The related question is to what extent the RA will be really adopted or implemented by an organization. As indicated by one of the candidates in chapter 7.5.4, an organization does not want to start from scratch. Therefore, an efficient way needs to be found to embed the RA or parts of it into their existing models and views. The availability of specific tools that automate or simplify the modeling process can increase the adoption and implementation of the RA.

The feedback also reveals that the modeled service-based environments within the Communication Layer are not the only integration patterns that can be used to connect the IT systems from the Stable Layer with the embedded IT systems within the Explorative Layer. This indication is important because it depends on an organization how each IT system is connected and which integration patterns are preferred. However, due to the rise of modular IT systems and the focus on decentralization, APIs are becoming more prominent to meet the demands and needs of the various stakeholders. This change also includes the need for organizations to rethink their integration patterns and API strategies to support and serve their internal and external stakeholders, which can increase the usage of two separate API platforms, as mentioned in chapters 5.3.1 and 7.5.3.

8.1.5. Advantages and added value of the Reference Architecture

As indicated in chapter 7.5.3, the RA can help to highlight and disclose patterns that are currently not covered by the current views and models. This is potentially the most significant gain and added value of the RA. Besides, the RA also provides some advantages. First of all, the RA can reveal if an IT landscape already contains or uses patterns that can be related to Bimodal IT. The second advantage is that the RA provides insight into an IT landscape and its implemented IT systems. It can even help to differentiate the IT systems between the layers and tiers by categorizing, allocating, and modeling the various IT systems within the platform. The third advantage is that it shows how the different IT systems are integrated and which integration patterns are most used or preferred. The fourth advantage can be related to the fact that the RA can support an Enterprise Architect or an Integration Architect in the discussion of why multiple service-based environments are required within an architecture. It can even help the architects to create an enterprise-wide integration pattern and associated principles, indicating how each system category should be connected within the IT landscape. The fifth advantage of the RA is that it can help to create outsourcing strategies and show which layers, system categories, or service-based environments are outsourced. As discussed in chapter 2.2.5, the bimodal IT concept differentiates six archetypes. Based on the archetypes and the RA, an organization can decide how it will apply and conduct its outsourcing strategy. And the last advantage hides in the fact that the RA can help to set-up and realize a Bimodal IT platform within an organization. This advantage is also acknowledged by one of the candidates within the feedback.

8.2. Hypothetical models and elements of the Reference Architecture

8.2.1. Service concepts and associated principles to realize and facilitate modular IT systems

Chapters 5.3.1, 5.3.3, and 6.3.3 discuss that the IT systems embedded within the Explorative Layer are mainly realized and exploited through microservices, whereas the IT systems within the system category Operational value-adding and Stabilized Systems of Innovation are established through web services.

The distinction between the two service concepts, web service versus microservice, can lead to several discussions and cause an unintended split within the development of APIs. An organization can presume that web services are not supportive or able to realize and facilitate IT systems within an Explorative Layer, which is not the case. Therefore, it is vital for an organization to understand both service concepts. First of all, a web service is platform-independent and loosely coupled, as described in chapter 2.3.3. A web service can be either based on SOAP or REST principles (Mumbaikar & Padiya, 2013). If a web service is based on REST principles, it can be implemented as a microservice, as indicated in chapter 2.3.4. A microservice can be described as a small, autonomous application that provides an independent, cohesive function or operation through the interaction of messages (Dragoni et al., 2017). It is used to offer an API to other microservices or IT systems so they can integrate with it. In the case a web service is based on SOAP principles, it uses XML to define the protocol and is sent over a transport protocol, such as SMTP, FTP, or HTTP, to communicate with other services (Mumbaikar & Padiya, 2013). The web service itself is a self-contained piece of functionality and can be used to connect with other web services, microservices, and monolithic IT systems.

Both service concepts are used to provide certain functionality and can realize a service-based architecture (SOA or MSA), as explained in chapters 2.3.3 and 2.3.4. This means that both service principles can be used to realize IT systems within the Explorative Layer and the system category Operational value-adding and Stabilized Systems of Innovation.

However, through figure 6-17 in chapter 6.3.4, the distinction can be increased between the two service concepts. Therefore, the discussion should be about the completeness of the figure and if it covers all the integration patterns that a service has to offer. To cope with both service concepts, a new model has been developed that is shown in figure 8-1. The figure shows that the Communication Layer can realize web services as well as microservices to support the development and exploitation of IT systems within the Explorative Layer and the system category Operational value-adding and Stabilized Systems of Innovation.

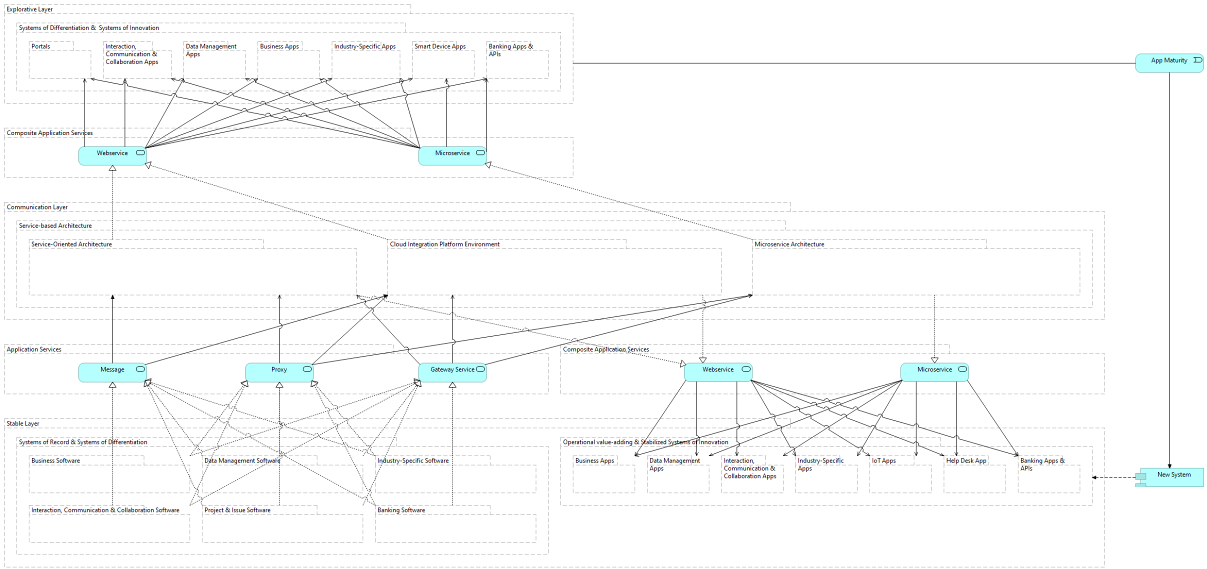

Figure 8-1 Fourth tier – Context view – Integration patterns between a Stable Layer and an Explorative Layer including the service concepts web services and microservices

8.2.2. The usefulness of multiple service-based environments within the Communication Layer

Apparently, two service principles can be used to build a web service. Based on this finding, the question could arise if all the three service-based environments, as presented in figure 6-3 in chapter 6.3.2, are required within a Bimodal IT platform because, for instance, an ESB environment could be sufficient to create and utilize both service principles. The answer to this question is that one service-based environment, such as an ESB or CIP environment, is capable enough to establish all the communication and integration patterns between and within the layers of the platform. However, it is also common for an organization to have a mix of integration patterns, such as several API types and services to connect and integrate their IT systems. Therefore, it entirely depends on an IT organization how they want to connect and integrate the various IT systems and which integration mechanisms are preferable to achieve this goal. In addition, from an architectural perspective, it is important to highlight that only three service-based environments are applicable to realize a service-based architecture within an IT landscape.

8.2.3. Deployment models regarding the Explorative Layer, CIP, and MSA environment

Figure 6-23 in chapter 6.3.6 presents the underlying deployment models of each layer within the sixth tier of the RA. The question is, however, if this model is complete.

Chapter 5.3.4 indicates that the IT systems embedded within the Explorative Layer are mainly deployed in the cloud. The same applies to the solutions related to the CIP and MSA environments. Regarding the Stable Layer, the data reveals that the associated IT systems can also be hosted in an on-premise or data center environment, next to the cloud. For the ESB environment applies that the corresponding IT systems are primarily deployed on-premise or in a data center. This finding is based on the result in chapter 5.2.4. Consequently, all these findings have led to the development of the context model presented in figure 6-23.

However, based on the researchers' experience, this does not necessarily mean that all the IT systems embedded within the Explorative Layer or the solutions used to enable the CIP and MSA service-based environments cannot be deployed in an on-premise or data center environment. The vendors also support this statement. For instance, Azure can be deployed on-premise by adopting and implementing Azure Stack (Microsoft, 2017). These deployment practices, however, are not disclosed in the findings mentioned in chapter 5.3.4.

To cope with all the possible hosting approaches for each layer, a new model has been developed that is shown in figure 8-2. This figure reflects the researchers' experience and the hosting practices provided by the vendors. It shows that the IT systems of the Explorative Layer and the solutions used to enable the CIP and MSA environments can also be hosted through other deployment models instead of the cloud.

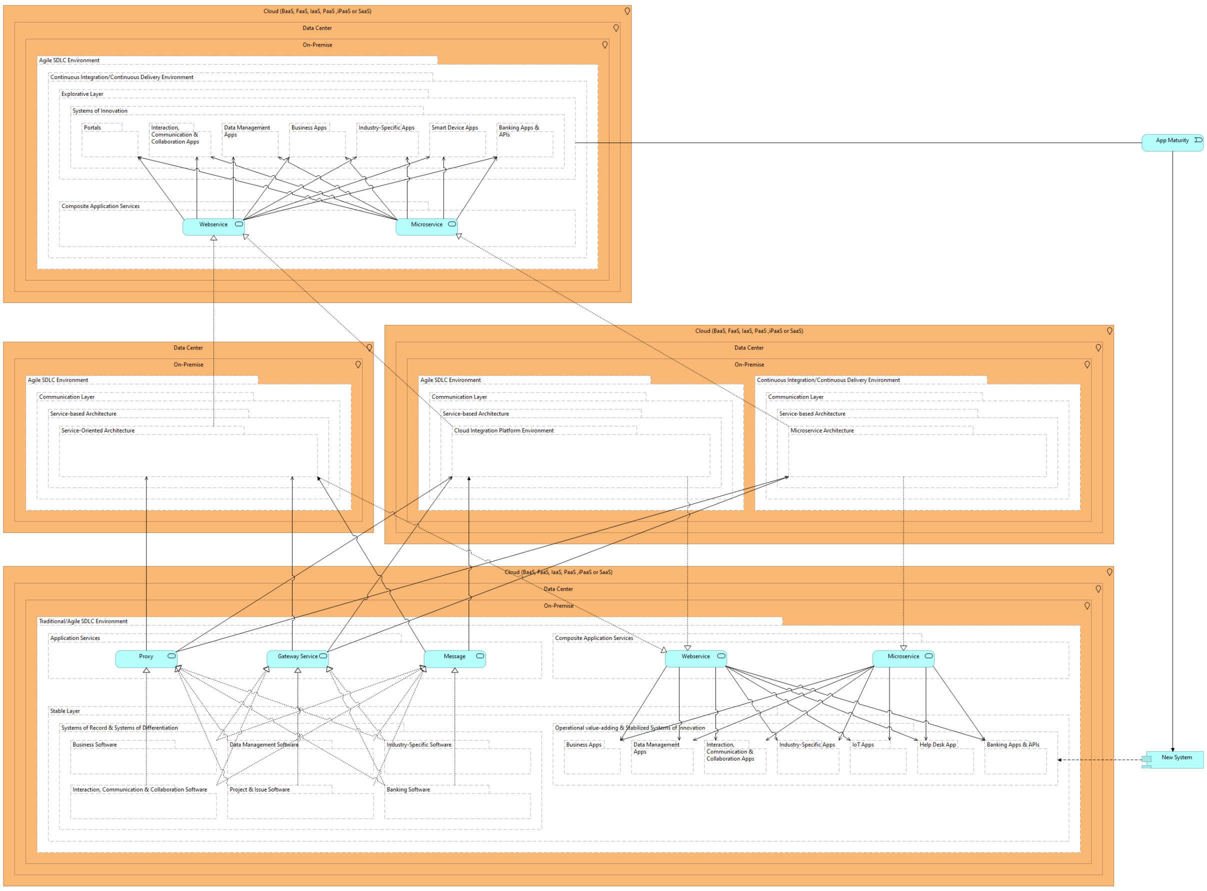

Figure 8-2 Sixth tier – Context view – Extension of the hosting environments and their interrelationship with the Bimodal IT platform

8.3. Defining the term Bimodal Architecture

As mention in chapter 2.1.1, the concept of Bimodal IT was introduced in 2013 by Gartner. Within the concept, Gartner deviates the IT delivery styles into two modes. However, Gartner does not discuss or define the architectural aspect that is associated with the concept. This also applies to the other concepts, which are based on and derived from the Bimodal IT concept.

Since a definition is missing in the existing body of knowledge, the research will define the architectural aspect. This is also in line with the objective of the research, as indicated in chapters 1.3 and 1.4, because the focus of the research is to fill the gap and to provide the design patterns of a Bimodal IT platform. Also, the definition itself is not limited to the term Bimodal Architecture and can be used in other concepts to support and define the related architectural aspect.

From an architectural point of view, the term Bimodal Architecture can be defined as follow:

> *"Bimodal Architecture is a coherent whole of the Stable Layer, Communication Layer, and Explorative Layer within an organization that supports business continuity and agility."*
>
> *"The Stable Layer defines an architecture that incorporates the Systems of Record and Systems of Differentiation (mission- and business-critical IT systems and related SaaS solutions to support business process automation) and their associated integration patterns within an organization to provide stability, reliability, and standardization to guarantee business continuity. Also, the layer comprises Operational value-adding and Stabilized Systems of Innovation to cover the value-adding business applications and (internal) APIs to leverage and improve the information need, responsiveness, and efficiency within an organization. These IT systems are modular, innovative, and fast-changing from nature."*
>
> *"The Communication Layer defines the service-based environment (ESB, CIP, and MSA environments) and presents the integration patterns between and also within the Stable Layer and Explorative Layer to enable and facilitate the connectivity, communication, integration, and collaboration between the various IT systems. It also supports in the delivery of modular IT systems and related increments. Conclusively, this layer integrates and connects the Stable Layer with the Explorative Layer and vice versa."*
>
> *"The Explorative Layer defines an architecture that incorporates the Systems of Innovation and their associated integration patterns within an organization to support*

the digitalization strategy and to improve and increase the organizational responsiveness, competitiveness, and the engagement with its stakeholders."

The definition itself is mainly based on and derived from the gathered interview data and the developed RA. It complements the concept of Bimodal IT and the other (related) concepts regarding the architectural aspect. Also, the definition does not necessarily implicate the different kinds of IT delivery styles because, for the platform itself, it does not matter how an IT project is delivered or implemented. It only cares where a specific IT system or increment is allocated within the architecture during its life cycle.

The definition can also contribute to the existing literature because the architectural aspect related to Bimodal IT is neither defined nor described before.

8.4. Future of the Reference Architecture

8.4.1. CI/CD is also becoming popular within the Stable Layer, next to the other layers

Another interesting point for the discussion can be related to CI/CD. Candidates 7, 10, and 14 mention in chapter 5.2.4 that CI/CD is also getting attention from the major software vendors. Therefore, there is an upcoming trend noticeable regarding CI/CD within the Stable Layer. However, the discussion is to which extend CI/CD can be used within the Stable Layer because it is conceivable that an organization does not want to adjust its business-critical objects on a frequent base.

On the other hand, to cope with the competition and the increased need for agility within established organizations, CI/CD can help realize and release increments faster than conventional approaches. However, it also depends on the architecture if and how CI/CD can be used in the development process regarding the IT systems embedded within the system category Systems of Record and Systems of Differentiation, especially for the mission- and business-critical IT systems. The reason is that a change can have a severe impact on several areas of a particular IT system or even multiple integration patterns when it is not done or thought thoroughly. Also, not every business-critical object needs to be subjected to rapid change or development.

For the Explorative Layer applies that it already makes use of CI/CD. CI/CD is foremost the preferred technique for the development of IT systems with microservices.

8.4.2. Perception of speed will change

At the moment, speed is translated through traditional approaches, also indicated as "slow pace," and agile approaches, referred to as "fast pace," to deliver an IT system within the IT landscape of an organization. Therefore, the speed regarding how much time it will take to deliver an IT system is completely based on the chosen approach. Other considerations, however, such as the speed at which technology and supportive development tools are changing and emerging and in which pace partners, suppliers, and vendors introduce or adopt new delivery approaches are not taken into account. Therefore, the term speed needs to put into perspective and reconsidered on an average period because the perception of speed changes continuously due to external factors.

Also, for the past five years, Low-Code Software Development (LCSD) has become an emerging paradigm within organizations. LCSD minimalizes source code development by providing a visual approach with interactive graphical interfaces, visual diagrams, and declarative languages to promote rapid application development (Abdullah Al Alamin et al., 2021; Tisi et al., 2019). One of the most significant gains of LCSD is that the related stages have a shorter time and execution span than an agile approach. Figure 8-3 presents the LCSD steps and compares it with the stages used in an agile software development environment. The inner-circle presents the steps that are required for the development of an IT system through the use of a Low-Code Development Platform (LCDP), whereas the outer-circle shows the steps to realize agile software development.

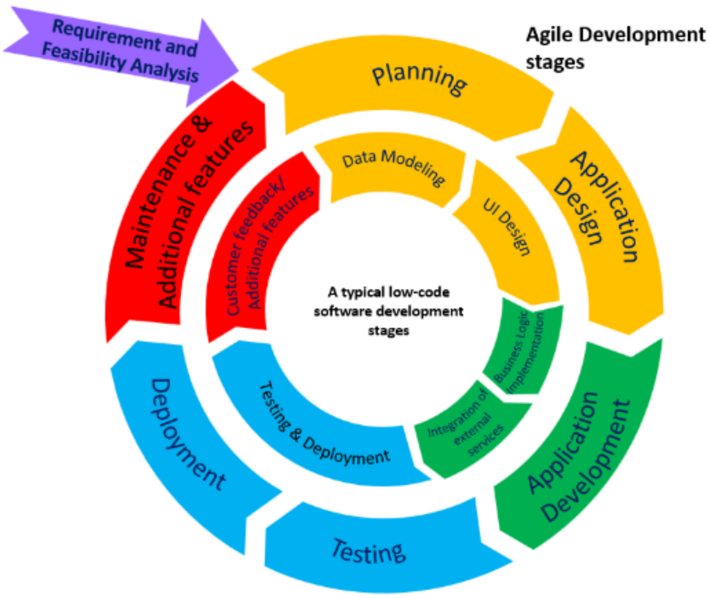

Figure 8-3 Steps to realize LCSD in relation to agile software development (Abdullah Al Alamin et al., 2021)

Within a LCDP, the back-end components are nothing more than a database and several APIs to establish the integration with other IT systems or third-party software, while the front-end comprises the business logic and UI (Abdullah Al Alamin et al., 2021). Every IT system developed on this platform can be adjusted in hours if the changes only relate to the front-end, while a complete IT system can be created within a day (Eaton, 2019). This way of development adds another dimension to the perception of speed and asks for a redefinition of the terms slow pace and fast pace.

8.4.3. Validity of the Reference Architecture

As indicated in the previous two chapters, CI/CD is becoming more dominant, and the perception of speed can change rapidly due to new emerging technologies and approaches. These changes are not only impacting the IT landscape of an organization but can also impact the models and views that are used to visualize the architecture.

When looking at the developed RA within the research, the conclusion can be drawn that the RA can still be valid and support the models and views with the upcoming trends and changes in the perception of speed. The reason is that the RA is set-up dynamically and can handle the modeling of different kinds of components. In addition, there will be a need for multiple IT environments to establish a proper architecture. In the case of a LCDP, the Stable Layer can contain the databases, whereas the UI can be embedded within the Explorative Layer. This is another approach of modeling compared to the current set-up of the RA in which different IT systems are allocated and modeled within the different layers of the platform.

Also, in chapter 5.3.5, the data retrieved from the interviews discloses that a Bimodal Architecture will stay relevant because the need for multiple IT environments within an IT landscape will never disappear. The RA can support this finding with its layers and related tiers.

8.5. Best practices regarding the implementation of the Reference Architecture

The RA itself can be used in multiple ways. First of all, it can be used to assess or orient whether a Bimodal IT platform is suitable for an organization. It can also help to identify whether an organization already uses patterns that can be associated with a Bimodal IT platform. Secondly, the RA can assist as a template to visualize an IT landscape in a unique way. Finally, the RA can support the visualization of hidden patterns, such as integration mechanisms and related patterns, to display how certain IT systems are connected and communicating with each other.

Suppose an organization decides to use the RA. In that case, the RA can assist in categorizing, allocating, and modeling the implemented IT systems into the different layers, system categories, and tiers. Subsequently, the RA can help to improve, enhance, or to construct additional models to complement the existing views and models. An organization may also decide to use the RA as a baseline and create new views and models from scratch to visualize their IT landscape. All these activities will help an organization to create guidelines in relation to the RA and to align their views and models with the patterns of a Bimodal IT platform.

Once the models and views are completed and aligned, the organization can verify and realize multiple detailed views. These views and models are intended and can be used to satisfy the needs of the business and IT projects. It can incorporate individual IT systems and how their objects are integrated with other IT systems within the architecture or highlight how certain business processes are related to the embedded IT systems within the different system categories.

Consequently, the constructed or enhanced views and models can be implemented within the organization. From this point in time, the views and models can assist in the analysis, decision-making, and discussion processes regarding the implemented IT systems and associated integration patterns within the IT landscape. It can also be used to execute or to adjust the IT strategy. The implementation step of the models can eventually lead to the realization, set-up, and adoption of a Bimodal IT platform within the architecture, which is the ultimate goal of the RA.

9. Conclusion, limitations, and future research

This chapter provides an answer to the research question and covers the conclusion, limitations, and incorporates future research regarding the adoption and implementation of a Bimodal IT platform.

9.1. Conclusion

The purpose of the research was to identify and determine the core elements that are related to the foundation and set-up of a Bimodal IT platform within the IT landscape of an organization.

Therefore, the main question throughout the research is:

What are the fundamental design patterns of a Bimodal Architecture?

The first step for answering the question was to take a comprehensive look at the concept of Bimodal IT and its associated aspects within the existing body of knowledge. However, during the literature review, it became clear that the focus of the literature is mainly related to the organizational and managerial aspects of the concept and discloses that the concept of Bimodal IT is just one of the concepts that can deal with the transformation journey of an organization.

The second step was to conduct an interview to retrieve the core elements from experts that already have experience with Bimodal IT. Through a multi-case study, 15 candidates distributed over nine companies were interviewed to collect the essential elements that are minimally required to set-up the foundation of the platform.

One of the major findings in the data supported the finding regarding the three fundamental layers, as concluded in the literature review. The main layers of the platform are a Stable Layer, an Explorative Layer, and a Communication Layer. The data also disclosed that several SDLC tools and deployment models should be taken into consideration because they are used to support the platform and, therefore, should be part of it.

The literature review regarding the identified architectural elements, combined with the gathered interview data about the elements that should be part of the platform, allowed the development of a RA to visualize, set-up, and implement a Bimodal IT platform.

In total, six tiers are constructed within the RA. The first tier presents the main layers of the platform. The second tier contains the predefined system categories and the associated service-based environments. The third tier comprises the various groups of IT systems. The fourth tier holds and allows the modeling of multiple IT systems, integration mechanisms, and related integration patterns. The fifth tier visualizes and contains the SDLC environments and related groups of IT systems to support and allow the development of the various IT systems and their associated increments. And the sixth tier comprises the deployment environments. In the RA itself, tiers one, two, and five are prescriptive to indicate and distinguish a Bimodal IT platform from other architectures.

The last step was to validate and justify the constructed RA. To fulfill this step, the RA has been validated via a feedback process. In total, four candidates were asks for their feedback on the developed RA. The validation showed that the RA is applicable and can assist in the analysis, decision-making, and discussion processes when initializing, realizing, exploiting, or utilizing an IT landscape.

Since the RA is applicable and could assist in the visualization, realization, implementation, and adoption of a Bimodal IT platform within an organization, the answer to the research question is:

The design patterns of a Bimodal IT platform comprise a Stable Layer, Communication Layer, and Explorative Layer. The Stable and Explorative layers include predefined system categories, whereas the Communication Layer can contain a maximum of three service-based environments. The platform also uses a SDLC and Deployment tier because the associated components are supportive to the platform. Based on the elements, a Reference Architecture has been developed, which can support organizations to visualize and structure a Bimodal IT platform. It can even assist in the realization, set-up, and implementation of the platform within their architecture.

Despite the answer, there is still a gap identifiable regarding the direction of enterprise architecture, software architecture, and their related demands within the future. The expectation is that Low-Code development will become more prominent. This approach allows organizations to build IT systems by anyone at any time because it is based on a visual design principle. The approach will also lead and introduce new ways to develop and exploit IT systems within the IT landscape(s) of organizations.

Due to the emergence and evolvement of new approaches, platforms, and technologies, the demand for a separate stable and fast-changing IT environment will only increase within the organizations. Therefore, there is a need for a platform or architecture that can accommodate with this demand, which is the case of the constructed RA.

To extend the answer with the practical use of the constructed RA, the outcome of this study can be concluded as follow:

The Reference Architecture accommodates and assists organizations in the set-up of a platform that can deal with multiple IT environments (stable versus fast-changing) and help them cope with the changing perceptions in agility and speed through the use of this architecture.

9.2. Limitations and future research

The concept of Bimodal IT is a broad subject that covers multiple aspects and topics within an organization. Therefore, to answer the research question, the scope was limited to the architectural aspect and solely focused on the essential elements regarding the set-up of the platform. Other architectural aspects and patterns, such as data flows, security, core infrastructural components, etc., are not considered in the research. Further research on these aspects and patterns could strengthen the constructed RA and help determine additional components, for instance, to govern the platform or align the business processes.

Another limitation of the study is that it does not mention anything regarding the team set-up. As mentioned, the Bimodal IT platform is built up through three layers. Therefore, it is conceivable that the various teams should have certain skillsets in-house to operate within the different layers. Since this aspect is not considered within the research, future research could investigate and determine which resources or skillsets are mandatory within the teams to ensure a smooth operation and enable the rapid delivery of IT systems through the platform.

Also, for determining and gathering the essential elements of the platform, the candidate selection process is conducted in a strict methodological way. This means that many roles and business members, such as business analysts, key-users, etc., were excluded when identifying and determining the elements. Due to this choice, the change exists that many members will miss or even will not be able to understand the models of the RA. Future revisions and additional viewpoints could solve this problem. However, this activity can increase the complexity of the RA and lead to multiplied views. Future research could propose alternatives to handle or mitigate these problems.

The fourth limitation relates to the focus of the research on established organizations. Due to the rise of digital offerings, services, and enhanced customer experience, every organization needs to focus more on IT agility and IT exploration. Therefore, the developed RA is not only applicable for the established organizations but could also be used by new entrants. However, further research or a revision on the RA needs to be conducted to create or tailor the models to fit their needs.

Finally, the RA is outlined as a black and white image in the research. An organization can either visualize or implement a Bimodal IT platform through the developed RA or will not able at all. This is not necessarily true. An approach could be developed in which the steps, stages, or maturity levels regarding the adoption or implementation of the platform are measured and disclosed. This approach could help to adopt and implement the platform with more ease.

References

Abdullah Al Alamin, M., Malakar, S., Uddin, G., Afroz, S., Bin Haider, T., & Iqbal, A. (2021). *An Empirical Study of Developer Discussions on Low-Code Software Development Challenges*. Retrieved from https://arxiv.org/pdf/2103.11429.pdf

Aderaldo, C. M., Mendonça, N. C., Pahl, C., & Jamshidi, P. (2017). Benchmark Requirements for Microservices Architecture Research. *Proceedings - 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering, ECASE 2017*, 8–13. https://doi.org/10.1109/ECASE.2017.4

Andersson, H., & Tuddenham, P. (2014). Reinventing IT to support digitization. *McKinsey*. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/reinventing-it-to-support-digitization

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE COMPUTER SOCIETY*, *33*(3), 42–52. https://doi.org/10.1109/MS.2016.64

Baratchi, M. (2018a). *Software Engineering - Introduction and SDLC*.

Baratchi, M. (2018b). *Software Engineering - Test Management and CI*.

Bayley, N., & Shacklady, J. (2015). Gearing Up for Growth Using Multi-speed IT. In *Accenture*. Retrieved from https://www.accenture.com/_acnmedia/accenture/conversion-assets/dotcom/documents/global/pdf/strategy_7/accenture-multi-speed-it-infographic.pdf

Bils, S. (2014). Two-speed IT: Necessary, But Not Sufficient. *Dell Technologies*. Retrieved from https://infocus.dellemc.com/scott-bils/two-speed-it-necessary-not-sufficient/

Bossert, O., Ip, C., & Jürgen, L. (2014). A two-speed architecture for the digital enterprise. *McKinsey*. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/a-two-speed-it-architecture-for-the-digital-enterprise

Bossert, O., Ip, C., & Starikova, I. (2015). Beyond agile: Reorganizing IT for faster software delivery. *McKinsey*. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/beyond-agile-reorganizing-it-for-faster-software-delivery

Brata De, R. (2014). SAP PI for Beginners. Retrieved from SAP Community Blogs website: http://scn.sap.com/docs/DOC-41766

Bughin, J., Chui, M., & Manyika, J. (2013). Ten IT-enabled business trends for the decade ahead. *McKinsey Quarterly*. Retrieved from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ten-it-enabled-business-trends-for-the-decade-ahead

Chandramouli, R., & Butcher, Z. (2020). Building Secure Microservices-based Applications Using Service-Mesh Architecture. *NIST Special Publication 800-204A*, 27. https://doi.org/10.6028/NIST.SP.800-204A

Cisco. (2019). What Is a Data Center? Retrieved March 18, 2021, from Cisco website:

https://www.cisco.com/c/en_in/solutions/data-center-virtualization/what-is-a-data-center.html

Clark, K. (2018a). *Comparing a service mesh with API management in a microservice architecture*. Retrieved from https://community.ibm.com/community/user/integration/blogs/kim-clark1/2018/11/13/comparing-a-service-mesh-with-api-management-in-a-microservice-architecture?S_TACT=105AGX15

Clark, K. (2018b). *The fate of the ESB*. Retrieved from https://developer.ibm.com/depmodels/microservices/articles/cl-lightweight-integration-1/

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., & Bone, M. (2010). The Concept of Reference Architectures. *Systems Engineering*, *13*(1), 14–27. https://doi.org/10.1002/sys.20129

Colbert, B. (2016). Think MultiModal IT...Not Bimodal. *Salesforce*. Retrieved from http://answers.salesforce.com/blog/2016/06/multimodal-versus-bimodal-it.html

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, *20*(3), 329–354. https://doi.org/10.1287/isre.1090.0236

Denning, S. (2016). Explaining Agile. *Forbes*, 1–12. Retrieved from https://www.forbes.com/sites/stevedenning/2016/09/08/explaining-agile/#68e919d8301b

Denning, S. (2020). Agile Isn ’ t New : What ’ s New Is The C-Suite Embracing It. *Forbes*. Retrieved from https://www.forbes.com/sites/stevedenning/2020/05/03/agile-isnt-new-whats-new-is-c-suite-embracing-it/#7413916e5e5e

Desmet, D., Löffler, M., & Weinberg, A. (2016). Modernizing IT for a digital era. *McKinsey*. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/modernizing-it-for-a-digital-era

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. In *Present and Ulterior Software Engineering*. https://doi.org/10.1007/978-3-319-67425-4_12

Eaton, D. (2019). Mendix Crash Course: From intro to app in just one day. Retrieved April 16, 2021, from Mendix website: https://www.mendix.com/blog/mendix-crash-course-from-intro-to-app-in-just-one-day/

El-Sheikh, E., Zimmermann, A., & Jain, L. C. (2016). Emerging Trends and Enterprise Service-Oriented in the Evolution of Architectures. In *Intelligent Systems Reference Library* (Vol. 111). https://doi.org/10.1007/978-3-319-40564-3

Elo, S., & Kyngäs, H. (2008). The qualitative content analysis process. *Journal of Advanced Nursing*, *62*(1), 107–115. https://doi.org/10.1111/j.1365-2648.2007.04569.x

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1–4.

https://doi.org/10.11648/j.ajtas.20160501.11

Fowler, M. (2006). *Continuous Integration*. Retrieved from https://moodle2019-20.ua.es/moodle/pluginfile.php/2228/mod_resource/content/2/martin-fowler-continuous-integration.pdf

Franken, O., King, C., & Daae, L. (2016). *How CIOs Can Power the Digital Transformation by Implementing Service Integration and Multi-speed IT How CIOs Can Power the Digital Transformation by Implementing Service Integration and Multi-speed IT*. Retrieved from https://www.capgemini.com/nl-nl/wp-content/uploads/sites/7/2017/07/service_integration_multi_speed_it.pdf

Gartner. (2011). Microservice. Retrieved December 2, 2019, from Gartner website: https://www.gartner.com/en/information-technology/glossary/microservice

Gartner. (2012a). Gartner Says Adopting a Pace-Layered Application Strategy Can Accelerate Innovation. *Businesswire*. Retrieved from https://www.businesswire.com/news/home/20120214006233/en/Gartner-Says-Adopting-a-Pace-Layered-Application-Strategy-Can-Accelerate-Innovation

Gartner. (2012b). Pace-layered Application Strategy. Retrieved January 9, 2021, from Gartner website: https://www.gartner.com/en/information-technology/glossary/pace-layered-application-strategy

Gartner. (2013). Bimodal. Retrieved November 17, 2019, from Gartner website: https://www.gartner.com/en/information-technology/glossary/bimodal

Gartner. (2019). Definition of Information Technology (IT) - Gartner Information Technology Glossary. Retrieved February 20, 2021, from Gartner website: https://www.gartner.com/en/information-technology/glossary/it-information-technology

Ghareeb, I., & Berchten, N. (2011). The digitisation of everything: “How organisations must adapt to changing consumer behaviour.” *Ernst & Young LLP*. Retrieved from https://www.academia.edu/23705650/The_digitisation_of_everything_How_organisations_must_adapt_to_changing_consumer_behaviour

Haffke, I., Kalgovas, B., & Benlian, A. (2017a). Options for transforming the IT function using bimodal IT. *MIS Quarterly Executive*, *16*(2), 101–120. Retrieved from https://www.semanticscholar.org/paper/Options-for-Transforming-the-IT-Function-Using-IT-Haffke-Kalgovas/14e76916528b5a99cff40254693e0ba35a3f59bd

Haffke, I., Kalgovas, B., & Benlian, A. (2017b). The Transformative Role of Bimodal IT in an Era of Digital Business. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*. https://doi.org/10.24251/hicss.2017.660

Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment : Leveraging information technology for transforming organizations. *IBM Systems Journal*, *32*(1). Retrieved from https://ieeexplore.ieee.org/abstract/document/5387096

Henthorn-Iwane, A. (2015). *Bimodal IT and Remodeling Traditional IT for Greater Agility*. 18–19. Retrieved from https://devops.com/bimodal-it-and-remodeling-traditional-it-for-greater-agility/

Hinchcliffe, D. (2016). IT leaders inundated with bimodal IT meme | ZDNet. *ZDNet*. Retrieved from https://www.zdnet.com/article/it-leaders-inundated-with-bimodal-it-meme/

Horlach, B., Drews, P., & Schirmer, I. (2016). Bimodal IT: Business-IT alignment in the age of digital transformation. *Multikonferenz Wirtschaftsinformatik (MKWI)*, *3*, 1417–1428. Retrieved from https://www.researchgate.net/profile/Paul_Drews/publication/287642679_Bimodal_IT_Business-IT_alignment_in_the_age_of_digital_transformation/links/56e1545508aee77a1600271c/Bimodal-IT-Business-IT-alignment-in-the-age-of-digital-transformation.pdf

Horlach, B., Drews, P., Schirmer, I., & Böhmann, T. (2017). Increasing the Agility of IT Delivery: Five Types of Bimodal IT Organization. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 5420–5429. https://doi.org/10.24251/hicss.2017.656

Hsieh, H.-F., & Shannon, S. E. (2005). Three Approaches to Qualitative Content Analysis. *SAGE Journals*, *15*(9), 1277–1288. https://doi.org/10.1177/1049732305276687

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.

IBM Cloud Education. (2019). What is Serverless Computing? | IBM. Retrieved July 15, 2020, from https://www.ibm.com/cloud/learn/serverless#toc-serverless-cipjgcK-

Jaramillo, D., Nguyen, D. V., & Smart, R. (2016). Leveraging microservices architecture by using Docker technology. *Conference Proceedings - IEEE SOUTHEASTCON*. https://doi.org/10.1109/SECON.2016.7506647

Katz, F. (2015). *Implementing a Bimodal IT Strategy for Mobile App Development*. Retrieved from https://appery.io/wp-content/uploads/Bimodal2106.pdf

Keller, T., Baumgartner, T., & Brucker-Kley, E. (2018). Towards a multi-modal IT. *Proceedings of the 16th International Conference E-Society*, 209–216. Retrieved from https://www.researchgate.net/publication/324602612_TOWARDS_A_MULTI-MODAL_IT

Ketterer, H., Rehberg, B., Schmid, C. N., & Kleine, D. (2016). The End of Two-Speed It. *BCG*. Retrieved from https://www.bcg.com/publications/2016/software-agile-digital-transformation-end-of-two-speed-it.aspx

Kirschner, B., & Kenney, P. (2014). *Lessons from the App Masters*. Retrieved from https://pages.apigee.com/rs/apigee/images/apigee-ebook-app-masters-pdf.pdf

Kratzke, N. (2018). A brief history of cloud application architectures. *Applied Sciences*, *8*(8). https://doi.org/10.3390/app8081368

Kruiswijk, B. (2017a). *Introduction to SOA*.

Kruiswijk, B. (2017b). *SOA Infrastructure*.

Leau, Y., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software Development Life Cycle AGILE vs Traditional Approaches. *International Conference on Information and Network Technology*, *37*. Retrieved from http://ku-fpg.github.io/files/agile-traditional.pdf

Lloyd, W., Ramesh, S., Chinthalapati, S., Ly, L., & Pallickara, S. (2018). Serverless computing: An investigation of factors influencing microservice performance. *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*. https://doi.org/10.1109/IC2E.2018.00039

Lu, Y., & Ramamurthy, R. K. (2020). Understanding the Link Between Information Technology Capability and Organizational Agility: An Empirical Examination. *MIS Quarterly*, *35*(4), 931–954. Retrieved from https://www.jstor.org/stable/41409967?seq=1

Mathiassen, L., & Pries-Heje, J. (2006). Business agility and diffusion of information technology. *European Journal of Information Systems*, *15*(2), 116–119. https://doi.org/10.1057/palgrave.ejis.3000610

Mesaglio, M., & Mingay, S. (2015). How to Achieve Enterprise Agility With a Bimodal Capability. *Gartner Research*. Retrieved from https://www.gartner.com/en/documents/3037317/how-to-achieve-enterprise-agility-with-a-bimodal-capabil

Microsoft. (2017). Azure Stack. Retrieved April 15, 2021, from Microsoft website: https://azure.microsoft.com/en-us/overview/azure-stack/

Microsoft. (2018). *Azure Integration Services*. Retrieved from https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-integration-services/Azure-Integration-Services-Whitepaper-v1-0.pdf/

Microsoft. (2019). Azure - Designing a microservices architecture. Retrieved March 14, 2021, from Microsoft website: https://docs.microsoft.com/en-us/azure/architecture/microservices/design/

Microsoft. (2020). *BizTalk*. Retrieved from https://docs.microsoft.com/en-us/biztalk/core/biztalk-server-core-documentation

Montesi, F., & Weber, J. (2016). Circuit Breakers, Discovery, and API Gateways in Microservices. *ArXiv - Computer Science*. Retrieved from https://www.semanticscholar.org/paper/Circuit-Breakers%2C-Discovery%2C-and-API-Gateways-in-Montesi-Weber/00025f8caf92b2e737027790b36c82208d9c1ac7

Morgan, R. E. (2004). Business agility and internal marketing. *European Business Review*, *16*(5), 464–472. Retrieved from https://www.emerald.com/insight/content/doi/10.1108/09555340410699811/full/html

Mumbaikar, S., & Padiya, P. (2013). Web services based on SOAP and REST principles. *International Journal of Scientific and Research Publications*, *3*(5). Retrieved from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.7936&rep=rep1&type=pdf

Namiot, D., & Sneps-Sneppe, M. (2014). On Micro-services Architecture. *International Journal of Open Information Technologies*, *2*. Retrieved from http://injoit.org/index.php/j1/article/view/139

Newmann, D. (2017). Agility Is The Key To Accelerating Digital Transformation. *Forbes*. Retrieved from https://www.forbes.com/sites/danielnewman/2017/04/18/agility-is-the-key-to-accelerating-digital-transformation/#6f9615d97277

Olsson, H. H., Alahyari, H., & Bosch, J. (2012). Climbing the “Stairway to heaven” - A mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. *Proceedings - 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2012*. https://doi.org/10.1109/SEAA.2012.54

Oosterhout, M. Van, Waarts, E., & Hillegersberg, J. Van. (2006). Change factors requiring agility and implications for IT. *European Journal of Information Systems*, (15), 132–145. https://doi.org/10.1057/palgrave.ejis.3000601

Overby, E., Bharadwaj, A., & Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, (15), 120–131. https://doi.org/10.1057/palgrave.ejis.3000600

Pahl, C. (2015). Containerization and the PaaS Cloud. *IEEE Cloud Computing*, *2*(3), 24–31. https://doi.org/10.1109/MCC.2015.51

Radcliffe, R. (2016). *Multi-speed IT : The reality of enterprise agility*. Retrieved from https://devops.com/multi-speed-stepping-stone-enterprise-agility-not-end-goal/

Ram, B. (2017). Mastering Multi-Speed IT. In *Accenture*. Retrieved from https://www.accenture.com/au-en/mastering-multi-speed-it

Reimer, T., Abraham, P., & Tan, Q. (2013). Federated identity access broker pattern for cloud computing. *Proceedings - 16th International Conference on Network-Based Information Systems, NBiS 2013*, 134–140. https://doi.org/10.1109/NBiS.2013.23

Rivera, J. (2014). Gartner Says in the Digital World CIOs Need Bimodal IT: Rock Solid IT with Ability for Fluidity. *Gartner Press Release*. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2014-10-06-gartner-says-in-the-digital-world-cios-need-bimodal-it-rock-solid-it-with-ability-for-fluidity

Roberts, N., & Grover, V. (2012). *Leveraging Information Technology Infrastructure to Facilitate a Firm’s Customer Agility and Competitive Activity: An Empirical Investigation*. *28*(4), 231–269. https://doi.org/10.2753/MIS0742-1222280409

Ross, J., Mocker, M., & Beath, C. (2018). Five Building Blocks of Digital Transformation. *MIT Center for Information Systems Research*, *18*(6), 1–4. Retrieved from https://cisr.mit.edu/publication/2018_0601_BuildingBlocks_RossMockerBeath

Roy, D., Athalye, M. S., Singh, M., & Natarajan, S. (2015). *TWO-SPEED IT*. Retrieved from https://www.infosyspublicservices.com/industries/public-sector/perspectives/Documents/two-speed-IT.pdf

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8–13. https://doi.org/10.1145/1764810.1764814

Sahil, B., Ankur, S., & Rani, U. (2017). A detailed study of Software Development Life Cycle (SDLC) Models. *International Journal Of Engineering And Computer Science*, *6*(7). Retrieved from http://www.ijecs.in/index.php/ijecs/article/view/2830

SAP. (2020). *SAP Integration Architecture Guide for Cloud and Hybrid Landscapes*. Retrieved from https://www.sap.com/documents/2020/11/400ae14b-bf7d-0010-87a3-c30de2ffd8ff.html

Saraswathi, R. (2020). Four Architecture Choices for Application Development in the Digital Age | IBM. Retrieved July 15, 2020, from IBM website: https://www.ibm.com/cloud/blog/four-architecture-choices-for-application-development

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2015). *Research methods for business students* (7th ed.). Retrieved from https://books.google.nl/books?id=vUdOCgAAQBAJ&dq=Research+methods+for+business+students+7th+edition&hl=nl&sa=X&ved=0ahUKEwjtiseIkLLjAhX_wAIHHbz1BY0Q6wEIMTAA

Serrano, N., Hernantes, J., & Gallardo, G. (2014). Service-oriented architecture and legacy systems. *IEEE Software*, *31*(5), 15–19. https://doi.org/10.1109/MS.2014.125

Sill, A. (2016). The Design and Architecture of Microservices. In *IEEE Cloud Computing* (Vol. 3). https://doi.org/10.1109/MCC.2016.111

Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management*, *33*(1), 140–152. https://doi.org/10.1016/j.ijproman.2014.03.008

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, *17*(4/2013). https://doi.org/10.12948/issn14531305/17.4.2013.06

The Open Group. (2016). Reference Architectures and Open Group Standards. Retrieved April 9, 2021, from The Open Group website: http://www.opengroup.org/iot/wp-refarchs/p2.htm

The Open Group. (2019). ArchiMate - Enterprise Architecture Modeling Language. Retrieved April 9, 2021, from The Open Group website: https://www.opengroup.org/archimate-forum/archimate-overview

Thomas, D. R. (2006). Method Notes A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*. https://doi.org/10.1177/1098214005283748

Tisi, M., Mottu, J.-M., Kolovos, D. S., De Lara, J., Guerra, E., Ruscio, D. Di, … Wimmer, M. (2019). Lowcomote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms. *CEUR*, *2405*(13). Retrieved from https://www.semanticscholar.org/paper/Lowcomote%3A-Training-the-Next-Generation-of-Experts-Tisi-Mottu/493276dc609388aa44358873b0b52cee6382753b

Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management*, *14*(3), 207–222. https://doi.org/10.1111/1467-8551.00375

Virmani, M. (2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *5th International Conference on Innovative Computing Technology, INTECH 2015*, (Intech). https://doi.org/10.1109/INTECH.2015.7173368

Xiao, Z., Wijegunaratne, I., & Qiang, X. (2016). Reflections on SOA and Microservices. *Proceedings - 4th International Conference on Enterprise Systems: Advances in Enterprise Systems, ES 2016*, 60–67. https://doi.org/10.1109/ES.2016.14

Yale, Y., Silveira, H., & Sundaram, M. (2016). A microservice based reference architecture model in the context of enterprise architecture. *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 1856–1860. https://doi.org/10.1109/IMCEC.2016.7867539

Yin, R. K. (2018). *Case study research and applications: design and methods* (6e ed.). SAGE Publications Ltd.

Zhang, Z., & Sharifi, H. (2000). A methodology for achieving agility in manufacturing organisations. *International Journal of Operations & Production Management*, *20*(4). Retrieved from https://www.emerald.com/insight/content/doi/10.1108/01443570010314818/full/pdf?title=a-methodology-for-achieving-agility-in-manufacturing-organisations

Appendix 1 – Keys

During the conventional content analysis 96 unique keys are defined within the text data. These unique keys are presented in table A.

Unique keys (#-tag)
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed
#bimodal_IT_layers_terminology
#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed
#candidate_current_role
#candidate_experience
#Communication_layer_CIP_capabilities
#Communication_layer_CIP_components
#Communication_layer_ESB_capabilities
#Communication_layer_ESB_components
#Communication_layer_MSA_capabilities
#Communication_layer_MSA_components
#Communication_layer_systems_and_software
#company_profile
#concept_bimodal_IT_agile_architecture
#concept_bimodal_IT_agile_teams
#concept_bimodal_IT_designs
#concept_bimodal_IT_estimations
#concept_bimodal_IT_future
#concept_bimodal_IT_planning
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_relevancy
#concept_bimodal_IT_roles

#concept_bimodal_IT_technical_debt
#concept_bimodal_IT_traditional_architecture
#concept_bimodal_IT_traditional_teams
#concept_digital_transformation_digital_platform
#concept_digital_transformation_operational_backbone
#concept_Pace-layerd_Application_strategy
#concept_Pace-layerd_Application_strategy_terminology
#continuous_integration_and_delivery_adoption
#continuous_integration_and_delivery_components
#continuous_integration_and_delivery_practices
#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities
#deployment_model
#deployment_model_cloud
#deployment_model_cloud_FaaS
#deployment_model_cloud_Hybrid
#deployment_model_cloud_iPaaS
#deployment_model_cloud_Paas
#deployment_model_cloud_Private
#deployment_model_cloud_Public
#deployment_model_cloud_SaaS
#deployment_model_cloud_serverless_computing
#deployment_model_data_center
#deployment_model_movement_towards_cloud
#deployment_model_on-premise
#Fast-Speed_characteristics
#Fast-Speed_systems_and_software
#Fast-Speed_systems_and_software_functionalities
#hosting_platform_cloud_components
#hosting_platform_data_center_components
#hosting_platform_on-premise_components
#impact_communication
#impact_continuous_alignment
#impact_planning
#impact_prioritization
#impact_release_management
#impact_release_strategy
#IT_capability
#IT_organization_focus
#IT_Service_Management

#IT_trends
#methodology
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_devops
#methodology_agile_framework_SAFe
#methodology_agile_framework_Scrum
#methodology_traditional_framework
#methodology_traditional_framework_waterfall
#notes_not_all_systems_can_be_deployed_in_the_cloud
#notes_no_CI/CD_pipeline_in_place
#organization_current_IT_landscape
#organization_IT_milestones
#organization_IT_roadmap
#organization_IT_strategy
#Slow-Speed_characteristics
#Slow-Speed_stabilized_systems_of_innovation
#Slow-Speed_systems_and_software
#Slow-Speed_systems_and_software_functionalities
#software_delivery_life_cycle_components
#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

Table A Unique keys via conventional content analysis

Appendix 2 – Correlation between keys and codebook tabs

Based on the keys multiple tabs are created within the codebook. The correlation of the keys and the tabs are presented in table B.

Unique keys (#-tag)	Codebook tabs
#candidate_current_role; #candidate_experience; #company_profile; #IT_capability; #IT_organization_focus	General information
#concept_Pace-layerd_Application_strategy; #concept_Pace-layerd_Application_strategy_terminology; #concept_digital_transformation_operational_backbone; #concept_digital_transformation_digital_platform; #bimodal_IT_layers_terminology; #bimodal_IT_layers_Slow-Speed; #bimodal_IT_integration_patterns_Slow-Speed; #bimodal_IT_layers_Fast-Speed; #bimodal_IT_integration_patterns_Fast-Speed; #bimodal_IT_layers_Communication_layer; #Communication_layer_systems_and_software; #bimodal_IT_layers_Communication_layer_ESB; #bimodal_IT_layers_Communication_layer_ESB_services; #bimodal_IT_layers_Communication_layer_CIP; #bimodal_IT_layers_Communication_layer_CIP_services; #bimodal_IT_layers_Communication_layer_MSA; #bimodal_IT_layers_Communication_layer_MSA_services; #bimodal_IT_integration_patterns_through_Communication_layer; #bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed; #bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed	Bimodal IT layers
#Slow-Speed_systems_and_software; #Slow-Speed_systems_and_software_functionalities; #Slow-Speed_stabilized_systems_of_innovation; #Fast-Speed_systems_and_software; #Fast-Speed_systems_and_software_functionalities	Systems & Software (Slow-Speed) (Fast-Speed)
#Slow-Speed_characteristics; #Fast-Speed_characteristics	Characteristics (Slow-Speed) (Fast-Speed)
#bimodal_IT_layers_Communication_layer_ESB & #Communication_layer_ESB_components; #Communication_layer_ESB_capabilities; #bimodal_IT_layers_Communication_layer_CIP &	Middleware components (ESB components

#Communication_layer_CIP_components; #Communication_layer_CIP_capabilities; #bimodal_IT_layers_Communication_layer_MSA & #Communication_layer_MSA_components; #Communication_layer_MSA_capabilities	& capabilities) (CIP components & capabilities) (MSA components & capabilities)
#methodology; #methodology_agile_framework; #methodology_agile_framework_continuous_integration_and_delivery; #methodology_agile_framework_devops; #methodology_agile_framework_SAFe; #methodology_agile_framework_Scrum; #methodology_traditional_framework; #methodology_traditional_framework_waterfall	SDLC methodology
#software_delivery_life_cycle_components; #software_delivery_life_cycle_systems_and_software; #software_delivery_life_cycle_systems_and_software_functionalities	SDLC tools & components (analysis)
#impact_release_management; #impact_release_strategy; #impact_continuous_alignment; #impact_communication; #impact_prioritization; #impact_planning	Impact
#continuous_integration_and_delivery_adoption	CI/CD approach
#continuous_integration_and_delivery_components; #continuous_integration_and_delivery_systems_and_software; #continuous_integration_and_delivery_systems_and_software_functionalities	CI/CD tools & components (analysis)
#continuous_integration_and_delivery_practices	CI/CD practices (analysis)
#deployment_model; #deployment_model_cloud; #deployment_model_cloud_Private; #deployment_model_cloud_Hybrid; #deployment_model_cloud_Public; #deployment_model_cloud_SaaS; #deployment_model_cloud_PaaS; #deployment_model_cloud_iPaaS; #deployment_model_cloud_FaaS; #deployment_model_cloud_serverless_computing; #deployment_model_on-premise; #deployment_model_data_center; #hosting_platform_on-premise_components; #hosting_platform_data_center_components;	Deployment models (on-premise & data center) (cloud)

#hosting_platform_cloud_components; #deployment_model_movement_towards_cloud	
#concept_bimodal_IT_future; #concept_bimodal_IT_relevancy; #IT_trends	Bimodal future
#organization_IT_milestones; #organization_current_IT_landscape; #organization_IT_roadmap; #organization_IT_strategy; #concept_bimodal_IT_agile_teams; #concept_bimodal_IT_traditional_teams; #concept_bimodal_IT_roles; #concept_bimodal_IT_designs; #concept_bimodal_IT_estimations; #concept_bimodal_IT_planning; #concept_bimodal_IT_prioritization; #concept_bimodal_IT_agile_architecture; #concept_bimodal_IT_traditional_architecture; #IT_Service_Management; #concept_bimodal_IT_technical_debt	Roadmap, Roles & Processes

Table B Correlation of keys with codebook tabs

Universiteit Leiden

Appendix 3 – High-level and sub-level codes

Table C presents the created high-level and sub-level codes to conduct the data analysis.

Code ID	High-level code	Subcode ID	Subcode
1	Discusses additional concepts next to Bimodal IT		
2	Uses typologies of other concepts to describe systems		
3	Deviated terminology for Slow-Speed and Fast-Speed		
4	Deviated terminology for the Communication layer		
5	Bimodal IT comprises three layers		
6	Stable Layer comprises Stabilized digital systems & software		
7	Discusses a separate internal and external API platform		
8	Integration between core systems and Stabilized digital systems & software		
9	Integration through services (within a layer)		
10	Integration through other technologies, like EAI, EDIFACT, Proxies, IFlows, etc.		
11	Integration between Stable Layer and Explorative Layer		
12	Integration through services (Stable Layer and Explorative Layer)		
13	Integration through other technologies (Stable Layer and Explorative Layer)		
14	Stable Layer facilitates Explorative Layer		
15	App movement (from Explorative Layer to Stable Layer)		

16	API movement (from Explorative Layer to Stable Layer)		
17	Discusses specific systems & software embedded in the Stable Layer		
18	Discusses Stabilized digital systems & software		
19	Discusses specific systems & software embedded in the Explorative Layer		
20	Stable Layer characteristics	20.1	Stable
		20.2	Reliable
		20.3	Robust
		20.4	Secure
		20.5	Predictable
		20.6	Supports and stores business relevant objects and data
		20.7	It does not change a lot once deployed in production
		20.8	Comprises fewer errors
		20.9	Traditional and agile way of software delivery
		20.10	Characterized by long release cycles
		20.11	Solid test processes
		20.12	Problem-solution design
		20.13	Focus on standardization
		20.14	Delivers operational value
		20.15	Implementation costs are high
		20.16	Use of ITIL processes
		20.17	High availability and fault tolerance
		20.18	The Stable domain is a facilitator for the Explorative domain
		20.19	Highly influenced by third parties

		20.20	Core systems are not easily replaceable due to their importance
21	Explorative Layer characteristics	21.1	Less stable
		21.2	Flexible
		21.3	Fast-changing
		21.4	Supports and stores minimal or no business relevant objects and data
		21.5	Errors are already factored in
		21.6	Only an agile way of development is supported
		21.7	Solutions are centered around interaction and experience
		21.8	New technology-oriented
		21.9	Trend to have only a development and production environment
		21.10	Delivers business value
		21.11	Rapid delivery oriented. Often and frequent deployments
		21.12	24/7 availability. It can be achieved through an off-line capability
		21.13	Requirements and planning changes continuously
		21.14	Features are small and independent
		21.15	Focus on innovation, experimentation, and exploration
		21.16	Compromises modern systems and applications (cloud-based)
		21.17	Continuous integration/deployment in place to increase time-to-market
22	Did not mention any ESB components		

23	Discusses all ESB components	23.1	Biztalk - Message Builder (comprising an Adapter Builder, Service Builder, and a Mediation Flow Builder)
		23.2	Biztalk - Integration Engine (comprising an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)
		23.3	Biztalk - Integration Controller/Integration Gateway (comprising a Configuration Manager, Web Service Manager, Access Controller, Exception Handler, Event Handler, Mediation Flow Manager, System Landscape/Tenant Manager, Auditor, and a Logger)
		23.4	Biztalk - Service Repository
		23.5	Biztalk - Business Activity Monitoring
		23.6	Biztalk - Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL)
		23.7	SAP PI/PO - Integration Builder (comprising an Enterprise Service Repository and an Integration Directory)
		23.8	SAP PI/PO - Integration Server (comprising an Adapter Engine, Integration Engine, and a Business Process Engine)
		23.9	SAP PI/PO - System Landscape (comprising a System Landscape/Tenant Manager)

		23.10	SAP PI/PO - Configuration and Monitoring (comprising a Configuration Manager, Web Service Manager, Mediation Flow Manager, Event Handler, Adapter Manager, Access Controller, Exception Handler, Logger, and an Auditor)
		23.11	SAP PI/PO - Business Activity Monitoring
		23.12	SAP PI/PO - Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL)
24	Mentions a part of the ESB components		
25	Uses Biztalk as an ESB		
26	Uses Oracle Service Bus as an ESB		
27	Uses SAP PI/PO as an ESB		
28	Did not discuss any ESB capabilities		
29	Mentions all ESB capabilities	29.1	Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Message Tracking, Throttling, and Exception Management)
		29.2	Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)
		29.3	Security (provides Authentication, Authorization,

			Encryption, Certification, and Message Security)
		29.4	Transportation (allows HTTP, HTTPS, SOAP, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)
30	Mentions a part of the ESB capabilities		
31	Did not mention any CIP components		
32	Discusses all CIP components	32.1	Azure Integration Services - Cloud Integration Service Message and Event Hub (comprises a Service Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Message Queues, Exception handler, Logger, and an Auditor)
		32.2	Azure Integration Services - Cloud Integration Service Connectivity Hub (comprises a Repository for Adapter Discovery, Adapter Builder, Adapter Configuration Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor)
		32.3	Azure Integration Services - Cloud Integration Service Integration Engine (comprises an Adapter Engine, Orchestration Engine,

		Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)
	32.4	Azure Integration Services - API Management Service (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager)
	32.5	Azure Integration Services - BPM Service (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)
	32.6	Azure Integration Services - Gateway Service (comprises a Traffic Manager, Exception Handler, Policy Manager, Version Manager, Call Handler, Logger, and an Auditor)
	32.7	SAP Cloud Platform - Cloud Integration Service Message and Event Hub (comprises an Integration Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Enterprise Service Repository, Message Queues, Exception handler, Logger, and an Auditor)
	32.8	SAP Cloud Platform - Cloud Integration Service Connectivity Hub (comprises a Repository for Adapter Discovery, Adapter Builder, Adapter Configuration

			Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor
		32.9	SAP Cloud Platform - Cloud Integration Service Integration Engine (comprises an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine)
		32.10	SAP Cloud Platform - API Management Service (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager)
		32.11	SAP Cloud Platform - BPM Service (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)
		32.12	SAP Cloud Platform - Gateway Service (comprises a Traffic Manager, Exception Handler, Policy Manager, Version Manager, Call Handler, Logger, and an Auditor)
33	Mentions a part of the CIP components		
34	Uses Azure Integration Services as a CIP		
35	Uses SAP Cloud Platform Integration as a CIP		
36	Did not discuss any CIP capabilities		
37	Mentions all CIP capabilities	37.1	Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration

			Management, Deployment, Load Balancing, Service Registry, Service Discovery, Call handling, Message Tracking, Throttling, Rate Limits, Traffic Management, Exception Management, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts)
		37.2	Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages)
		37.3	Security (provides Authentication, Authorization, Encryption, Certification, and Message Security)
		37.4	Transportation (allows HTTP, HTTPS, SOAP, OData, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)
38	Mentions a part of the CIP capabilities		
39	Did not mention any MSA components		
40	Discusses all MSA components	40.1	Nodes
		40.2	Pods (*represents the running containers)*
		40.3	Scheduler
		40.4	Controllers
		40.5	Key-value Database
		40.6	API Gateway
		40.7	API Manager
		40.8	API server

		40.9	Resource Manager
		40.10	Container Orchestration Engine
		40.11	Storage
		40.12	Queues
		40.13	Container Registry
		40.14	Container Repository (optional)
		40.15	Network Manager
		40.16	Infrastructure Security Manager
		40.17	Provisioning Manager
		40.18	Logs
41	Mentions a part of the MSA components		
42	Uses Azure as a MSA		
43	Uses another platform as a MSA rather than Azure		
44	Did not discuss any MSA capabilities		
45	Mentions all MSA capabilities	45.1	Container Orchestration Platform (provides Container Orchestration, Container clustering, Image Discovery, Container Security, Network settings, Key Vault, Load Balancing, Scheduling, Rollouts, Rollbacks, Self-Healing, and Workflow Management)
		45.2	API Management (provides Service Discovery, Messaging, Routing, Call handling, Transformation, Rate Limits, Traffic Management, Throttling, Caching, Circuit Breaker, Failover, Microservice Security, Configuration Management, Networking/IP, Analytics, API Version Management, Monitoring,

			Logging, and Policies and Contracts)
		45.3	Container (comprises microservices)
46	Mentions a part of the MSA capabilities		
47	Usage of traditional methodologies in the Explorative domain		
48	Usage of agile methodologies in the Explorative domain		
49	Usage of agile methodologies in an ESB environment		
50	Usage of agile methodologies in a CIP environment		
51	Usage of agile methodologies in a MSA environment		
52	Usage of traditional methodologies in the Stable domain		
53	Usage of agile methodologies in the Stable domain		
54	Usage of traditional methodologies in an ESB environment		
55	Usage of traditional methodologies in a CIP environment		
56	Usage of traditional methodologies in a MSA environment		
57	Mentions that CI/CD is upcoming within the Stable domain		
58	Discusses SDLC tools & components	58.1	A tool/system to manage the planning
		58.2	A tool/system to manage the requirements
		58.3	A tool/system that supports Project Management and/or IT Service Management activities

		58.4	A tool/system for tickets
		58.5	Functional and technical design tools
		58.6	Collaboration tools
		58.7	A Document Management System
		58.8	Version Control systems/Repositories
		58.9	Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, software, and APIs
		58.10	Automation server (build, test, publish and deploy automation)
		58.11	Middleware tools
		58.12	Software Quality Assurance tools. This category also includes test tools
		58.13	Release and Deployment tools
		58.14	Monitoring and Analytics tools
59	Indicates that release management has an impact		
60	Indicates that prioritization has an impact		
61	Indicates that continuous alignment and communication are a necessity		
62	Mentions that CI/CD is used in the Stable domain		
63	Mentions that CI/CD is used in the Explorative domain		
64	Mentions that CI/CD is used in an ESB environment		
65	Mentions that CI/CD is used in a CIP environment		
66	Mentions that CI/CD is used in a MSA environment		

67	Discusses CI/CD tools & components	67.1	Use of local IDEs or DKs
		67.2	Completed source code are submitted to a Repository
		67.3	Board functionality. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug Tracking
		67.4	Version Control systems/Repositories
		67.5	Artifact Repository. To store, publish, and share software packages securely
		67.6	Pipeline functionality. This functionality is an automation server and is used to automate the build, test, and publish activities
		67.7	Release functionality. To automate the deployment step
		67.8	Monitoring and analysis functionalities. These tools are used for monitoring, analysis, and visualization
		67.9	Alert functionality. To notify a developer whether the build succeeded or failed
		67.10	A tool to automate and test the various API calls
68	Mentions CI/CD practices	68.1	A CI/CD strategy needs to be in place
		68.2	A recovery strategy should be in place
		68.3	A proper branching structure should be in place
		68.4	Increments should be small. This minimizes the branches

		68.5	Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages
		68.6	Only required features and functionalities should be build and released
		68.7	The developer should test the increment rigorously on his/her local environment
		68.8	Within the pipeline, the proper tests should be configured and kicked off
		68.9	A verification step should be in place before each deployment
		68.10	Configurations and passwords are easily accessible and always up-to-date
		68.11	Use of a Version Control System
		68.12	Use of an Automation Server that can automatically build, test, publish and deploy the various increments
		68.13	Use of a Release Manager tool to deploy the application on any environment
		68.14	Provide KPIs to measure the pipeline executions
		68.15	Monitoring and analytics need to be in place to measure and analyze the solutions
		68.16	All the affected layers must confirm to an agile approach. Hereby, DevOps is the preferred method
69	Mentions that there is no relationship between Bimodal IT and hosting		

70	Discusses that there is a relationship between Bimodal IT and hosting
71	There are practices in place for hosting different kinds of systems
72	There are no common practices in place regarding to hosting
73	Discusses that the systems & software of the Explorative domain are hosted on-premise
74	Discusses that the systems & software of the Explorative domain are hosted in a data center
75	Discusses that the systems & software of the Explorative domain are hosted in the cloud
76	Discusses that the systems & software of the Stable domain are hosted on-premise
77	Discusses that the systems & software of the Stable domain are hosted in a data center
78	Discusses that the systems & software of the Stable domain are hosted in the cloud
79	Discusses that the ESB tools & components are hosted on-premise
80	Discusses that the ESB tools & components are hosted in a data center
81	Discusses that the ESB tools & components are hosted in the cloud

82	Discusses that the CIP tools & components are hosted on-premise		
83	Discusses that the CIP tools & components are hosted in a data center		
84	Discusses that the CIP tools & components are hosted in the cloud		
85	Discusses that the MSA tools & components are hosted on-premise		
86	Discusses that the MSA tools & components are hosted in a data center		
87	Discusses that the MSA tools & components are hosted in the cloud		
88	Mentions a trend towards the cloud		
89	Mentions the components of an on-premise hosting platform	89.1	Server Farm (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)
		89.2	VPN or (Reverse) Proxy
		89.3	Load balancing
		89.4	Data access
		89.5	File access
		89.6	Data cache
		89.7	Web cache
		89.8	Process scheduling
90	Mentions the components of a data center hosting platform	90.1	Server Farm (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)
		90.2	VPN or (Reverse) Proxy
		90.3	Load balancing
		90.4	Data access
		90.5	File access
		90.6	Data cache

		90.7	Web cache
		90.8	Process scheduling
91	Mentions the components of a cloud hosting platform	91.1	Physical servers located in a hosting center (comprising application servers, database servers, file servers, exchange servers, web servers, etc.)
		91.2	Containerization technology
		91.3	Images (creation and discovery)
		91.4	Event-driven serverless compute platform
		91.5	Cloud solution is offered through a Public, Private, or Hybrid model
		91.6	Software as a Service (SaaS)
		91.7	Platform as a Service (Paas)
		91.8	Integration Platform as a Service (iPaaS)
		91.9	Function as a Service (FaaS)
		91.10	Back-end as a Service (BaaS)
		91.11	Infrastructure as a Service (IaaS)
92	Bimodal IT will stay relevant in the future		
93	Bimodal IT is not relevant		
94	Other concepts are more suitable than Bimodal IT		
95	Need for a multi-disciplinary team set-up as per the agile manifesto		
96	Business should change itself to commit to the agile way of working		
97	Agile architecture differs from a Traditional architecture		
98	Mentions that the priority is set by a PO within the Explorative domain		

99	Mentions that the priority is set by a PO within the Stable domain
100	Mentions that the priority is set by the business within the Explorative domain
101	Mentions that the priority is set by the business within the Stable domain
102	IT Service Management does not distinguish between the Stable and Explorative domain
103	Discusses a high technical debt in the Explorative domain
104	Discusses a low technical debt in the Explorative domain
105	Discusses a high technical debt in the Communication Layer
106	Discusses a low technical debt in the Communication Layer
107	Discusses a high technical debt in the Stable domain
108	Discusses a low technical debt in the Stable domain

Table C High-level codes and associated subcodes

Appendix 4 – Correlation between codebook tabs, topics, and codes

Table D shows the correlation between the codebook tabs, the topics and the related codes.

Codebook tabs	Topics and associated high-level codes and subcodes
General information	Roles: - Enterprise Architect (EA) - Integration Architect (IA) - Solution Architect (SA) - Project Manager (PM) - Developer (DEV) - Integration and Technical Lead (ITL) - Innovation Lead (INL) Industry: - Utility - Banking - Consumer Goods - Consulting - Postal - Products - Municipality - Hosting
Bimodal IT layers	Pace-layered Application Strategy and Digital Transformation can be used to achieve agility: - Discusses additional concepts next to Bimodal IT - Uses typologies of other concepts to describe systems Terminology of the layers: - Deviated terminology for Slow-Speed and Fast-Speed - Deviated terminology for the Communication layer Bimodal components, domains, systems & software, and integration patterns: - Bimodal IT comprises three layers - Stable Layer comprises Stabilized digital systems & software - Discusses a separate internal and external API platform - Integration between core systems and Stabilized digital systems & software - Integration through services (within a layer)

	- Integration through other technologies, like EAI, EDIFACT, Proxies, IFlows, etc. - Integration between Stable Layer and Explorative Layer - Integration through services (Stable Layer and Explorative Layer) - Integration through other technologies (Stable Layer and Explorative Layer) Stable Layer is a facilitator for the Explorative domain: - Stable Layer facilitates Explorative Layer Software movement from the Explorative Layer to the Stable Layer: - App movement (from Explorative Layer to Stable Layer) - API movement (from Explorative Layer to Stable Layer)
Systems & Software (Slow-Speed) (Fast-Speed)	Bimodal components, domains, systems & software, and integration patterns: - Discusses specific systems & software embedded in the Stable Layer - Discusses Stabilized digital systems & software - Discusses specific systems & software embedded in the Explorative Layer
Characteristics (Slow-Speed) (Fast-Speed)	Layer characteristics: - Stable Layer characteristics - Explorative Layer characteristics Stable Layer characteristics (subcodes): - Stable - Reliable - Robust - Secure - Predictable - Supports and stores business relevant objects and data - It does not change a lot once deployed in production - Comprises fewer errors - Traditional and agile way of software delivery - Characterized by long release cycles - Solid test processes - Problem-solution design - Focus on standardization

	- Delivers operational value - Implementation costs are high - Use of ITIL processes - High availability and fault tolerance - The Stable domain is a facilitator for the Explorative domain - Highly influenced by third parties - Core systems are not easily replaceable due to their importance Explorative Layer characteristics (subcodes): - Less stable - Flexible - Fast-changing - Supports and stores minimal or no business relevant objects and data - Errors are already factored in - Only an agile way of development is supported - Solutions are centered around interaction and experience - New technology-oriented - Trend to have only a development and production environment - Delivers business value - Rapid delivery oriented. Often and frequent deployments - 24/7 availability. It can be achieved through an off-line capability - Requirements and planning changes continuously - Features are small and independent - Focus on innovation, experimentation, and exploration - Compromises modern systems and applications (cloud-based) - Continuous integration/deployment in place to increase time-to-market
Middleware components	ESB components & capabilities: - Did not mention any ESB components - Discusses all ESB components - Mentions a part of the ESB components - Uses Biztalk as an ESB - Uses Oracle Service Bus as an ESB - Uses SAP PI/PO as an ESB - Did not discuss any ESB capabilities - Mentions all ESB capabilities - Mentions a part of the ESB capabilities CIP components & capabilities:

	- Did not mention any CIP components - Discusses all CIP components - Mentions a part of the CIP components - Uses Azure Integration Services as a CIP - Uses SAP Cloud Platform Integration as a CIP - Did not discuss any CIP capabilities - Mentions all CIP capabilities - Mentions a part of the CIP capabilities MSA components & capabilities: - Did not mention any MSA components - Discusses all MSA components - Mentions a part of the MSA components - Uses Azure as a MSA - Uses another platform as a MSA rather than Azure - Did not discuss any MSA capabilities - Mentions all MSA capabilities - Mentions a part of the MSA capabilities
ESB components (analysis)	ESB components (subcodes): - Biztalk -- Message Builder (comprising an Adapter Builder, Service Builder, and a Mediation Flow Builder) -- Integration Engine (comprising an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine) -- Integration Controller/Integration Gateway (comprising a Configuration Manager, Web Service Manager, Access Controller, Exception Handler, Event Handler, Mediation Flow Manager, System Landscape/Tenant Manager, Auditor, and a Logger) -- Service Repository -- Business Activity Monitoring* -- Business Process Manager and Process Automation (comprising a Workflow Builder, Rule Builder, and BPEL) - SAP PI/PO -- Integration Builder (comprising an Enterprise Service Repository and an Integration Directory) -- Integration Server (comprising an Adapter Engine, Integration Engine, and a Business Process Engine) -- System Landscape (comprising a System Landscape/Tenant Manager)

	-- Configuration and Monitoring (comprising a Configuration Manager, Web Service Manager, Mediation Flow Manager, Event Handler, Adapter Manager, Access Controller, Exception Handler, Logger, and an Auditor) * Business Activity Monitoring & Business Process Manager and Process Automation contains the same components as Biztalk
ESB capabilities (analysis)	ESB capabilities (subcodes): - Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Message Tracking, Throttling, and Exception Management) - Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages) - Security (provides Authentication, Authorization, Encryption, Certification, and Message Security) - Transportation (allows HTTP, HTTPS, SOAP, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)
CIP components (analysis)	CIP components (subcodes): - Azure Integration Services -- Cloud Integration Service - Message and Event Hub (comprises a Service Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Message Queues, Exception handler, Logger, and an Auditor) -- Cloud Integration Service - Connectivity Hub* (comprises a Repository for Adapter Discovery, Adapter Builder, Adapter Configuration Manager, Adapter Security Manager, Adapter Event Handler, Workflow Builder, Access Controller, Logger, and an Auditor) -- Cloud Integration Service - Integration Engine* (comprises an Adapter Engine, Orchestration Engine, Transformation Engine, Routing Engine, Rule Engine, and a Publisher and Subscriber Engine) -- API Management Service* (comprises an API Repository, API Designer, API Configuration Manager, and an API Test Manager) -- BPM Service* (comprises a Business Process Manager, Workflow Builder, and a Business Activity Monitor)

	-- Gateway Service (comprises a Traffic Manager, Exception Handler, Policy Manager, Version Manager, Call Handler, Logger, and an Auditor) - SCP -- Cloud Integration Service - Message and Event Hub (comprises an Integration Repository, Integration/Message Builder, System Landscape/Tenant Manager, Transaction Handler, Configuration Manager, Security Manager, Access Controller, Event Handler, Enterprise Service Repository, Message Queues, Exception handler, Logger, and an Auditor) * Cloud Integration Service - Connectivity Hub; Cloud Integration Service - Integration Engine; API Management Service; BPM Service & Gateway Service contains the same components as Azure Integration Services
CIP capabilities (analysis)	CIP capabilities (subcodes): - Operations and Management (provides Statistics, Statuses, Alerts, Failover, Configuration Management, Deployment, Load Balancing, Service Registry, Service Discovery, Call handling, Message Tracking, Throttling, Rate Limits, Traffic Management, Exception Management, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts) - Mediation (provides Transformation, Protocol Translation, Caching, Message Enrichment, Dynamic Routing, Message Validation, Reliable Messaging, and allows Pass-Through of Messages) - Security (provides Authentication, Authorization, Encryption, Certification, and Message Security) - Transportation (allows HTTP, HTTPS, SOAP, OData, REST, XML, FTP, SFTP, SMTP, Flat File, Database Adapters, 3rd Party Adapters, and Custom Adapters)
MSA components (analysis)	MSA components (subcodes): - Nodes - Pods (*represents the running containers)* - Scheduler - Controllers - Key-value Database - API Gateway - API Manager - API server - Resource Manager - Container Orchestration Engine

	- Storage - Queues - Container Registry - Container Repository (optional) - Network Manager - Infrastructure Security Manager - Provisioning Manager - Logs
MSA capabilities (analysis)	MSA capabilities (subcodes): - Container Orchestration Platform (provides Container Orchestration, Container clustering, Image Discovery, Container Security, Network settings, Key Vault, Load Balancing, Scheduling, Rollouts, Rollbacks, Self-Healing, and Workflow Management) - API Management (provides Service Discovery, Messaging, Routing, Call handling, Transformation, Rate Limits, Traffic Management, Throttling, Caching, Circuit Breaker, Failover, Microservice Security, Configuration Management, Networking/IP, Analytics, API Version Management, Monitoring, Logging, and Policies and Contracts) - Container (comprises microservices)
SDLC methodology	Use of agile methodologies in the layers: - Usage of traditional methodologies in the Explorative domain - Usage of agile methodologies in the Explorative domain - Usage of agile methodologies in an ESB environment - Usage of agile methodologies in a CIP environment - Usage of agile methodologies in a MSA environment Use of traditional methodologies in the layers: - Usage of traditional methodologies in the Stable domain - Usage of agile methodologies in the Stable domain - Usage of traditional methodologies in an ESB environment - Usage of traditional methodologies in a CIP environment - Usage of traditional methodologies in a MSA environment CI/CD is upcoming within the Stable domain: - Mentions that CI/CD is upcoming within the Stable domain
SDLC tools & components (analysis)	Discusses SDLC tools & components SDLC tools & components (subcodes): - A tool/system to manage the planning - A tool/system to manage the requirements

	- A tool/system that supports Project Management and/or IT Service Management activities - A tool/system for tickets - Functional and technical design tools - Collaboration tools - A Document Management System - Version Control systems/Repositories - Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, software, and APIs - Automation server (build, test, publish and deploy automation) - Middleware tools - Software Quality Assurance tools. This category also includes test tools - Release and Deployment tools - Monitoring and Analytics tools
Impact	Impact of release management and prioritization on a Bimodal IT environment: - Indicates that release management has an impact - Indicates that prioritization has an impact Need for continuous alignment: - Indicates that continuous alignment and communication are a necessity
CI/CD approach	Use of agile methodologies in the layers: - Mentions that CI/CD is used in the Stable domain - Mentions that CI/CD is used in the Explorative domain - Mentions that CI/CD is used in an ESB environment - Mentions that CI/CD is used in a CIP environment - Mentions that CI/CD is used in a MSA environment
CI/CD tools & components (analysis)	Discusses CI/CD tools & components CI/CD tools & components (subcodes): - Use of local IDEs or DKs - Completed source code are submitted to a Repository - Board functionality. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug Tracking - Version Control systems/Repositories - Artifact Repository. To store, publish, and share software packages securely

	- Pipeline functionality. This functionality is an automation server and is used to automate the build, test, and publish activities - Release functionality. To automate the deployment step - Monitoring and analysis functionalities. These tools are used for monitoring, analysis, and visualization - Alert functionality. To notify a developer whether the build succeeded or failed - A tool to automate and test the various API calls
CI/CD practices (analysis)	Mentions CI/CD practices CI/CD practices (subcodes): - A CI/CD strategy needs to be in place - A recovery strategy should be in place - A proper branching structure should be in place - Increments should be small. This minimizes the branches - Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages - Only required features and functionalities should be build and released - The developer should test the increment rigorously on his/her local environment - Within the pipeline, the proper tests should be configured and kicked off - A verification step should be in place before each deployment - Configurations and passwords are easily accessible and always up-to-date - Use of a Version Control System - Use of an Automation Server that can automatically build, test, publish and deploy the various increments - Use of a Release Manager tool to deploy the application on any environment - Provide KPIs to measure the pipeline executions - Monitoring and analytics need to be in place to measure and analyze the solutions - All the affected layers must confirm to an agile approach. Hereby, DevOps is the preferred method
Deployment models	Determination of a relationship between Bimodal IT and hosting: - Mentions that there is no relationship between Bimodal IT and hosting - Discusses that there is a relationship between Bimodal IT and hosting

	Hosting practices: - There are practices in place for hosting different kinds of systems - There are no common practices in place regarding to hosting Available deployment models for software: - Discusses that the systems & software of the Explorative domain are hosted on-premise - Discusses that the systems & software of the Explorative domain are hosted in a data center - Discusses that the systems & software of the Explorative domain are hosted in the cloud - Discusses that the systems & software of the Stable domain are hosted on-premise - Discusses that the systems & software of the Stable domain are hosted in a data center - Discusses that the systems & software of the Stable domain are hosted in the cloud - Discusses that the ESB tools & components are hosted on-premise - Discusses that the ESB tools & components are hosted in a data center - Discusses that the ESB tools & components are hosted in the cloud - Discusses that the CIP tools & components are hosted on-premise - Discusses that the CIP tools & components are hosted in a data center - Discusses that the CIP tools & components are hosted in the cloud - Discusses that the MSA tools & components are hosted on-premise - Discusses that the MSA tools & components are hosted in a data center - Discusses that the MSA tools & components are hosted in the cloud Trend towards the cloud: - Mentions a trend towards the cloud

On-premise & data center hosting	Components and capabilities of the hosting environments: - Mentions the components of an on-premise hosting platform - Mentions the components of a data center hosting platform Components of an on-premise and data center environment (subcodes): - Server Farm (comprising application servers, database servers, file servers, exchange servers, web servers, etc.) - VPN or (Reverse) Proxy Capabilities of an on-premise or data center environment (subcodes): - Load balancing - Data access - File access - Data cache - Web cache - Process scheduling
Cloud hosting	Components and capabilities of the hosting environments: - Mentions the components of a cloud hosting platform Components of a cloud environment (subcodes): - Physical servers located in a hosting center (comprising application servers, database servers, file servers, exchange servers, web servers, etc.) - Containerization technology Capabilities of a cloud environment (subcodes): - Images (creation and discovery) - Event-driven serverless compute platform - Cloud solution is offered through a Public, Private, or Hybrid model Subscription methods of a cloud environment (subcodes): - Software as a Service (SaaS) - Platform as a Service (Paas) - Integration Platform as a Service (iPaaS) - Function as a Service (FaaS) - Back-end as a Service (BaaS) - Infrastructure as a Service (IaaS)
Bimodal future	Bimodal IT relevancy:

	- Bimodal IT will stay relevant in the future - Bimodal IT is not relevant - Other concepts are more suitable than Bimodal IT
Roadmap, Roles & Processes	Agile commitment through multi-disciplinary teams and change of business: - Need for a multi-disciplinary team set-up as per the agile manifesto - Business should change itself to commit to the agile way of working Agile architecture versus Traditional architecture: - Agile architecture differs from a Traditional architecture Prioritization within the layers: - Mentions that the priority is set by a PO within the Explorative domain - Mentions that the priority is set by a PO within the Stable domain - Mentions that the priority is set by the business within the Explorative domain - Mentions that the priority is set by the business within the Stable domain IT Service Management: - IT Service Management does not distinguish between the Stable and Explorative domain Technical debt: - Discusses a high technical debt in the Explorative domain - Discusses a low technical debt in the Explorative domain - Discusses a high technical debt in the Communication Layer - Discusses a low technical debt in the Communication Layer - Discusses a high technical debt in the Stable domain - Discusses a low technical debt in the Stable domain

Table D Correlation between the codebook tabs, topics, and codes

Appendix 5 – Correlation between codebook tabs, topics, and result clusters

Table E presents the correlation between the codebook tabs, related topics, and associated clusters.

Clusters	Codebook tabs	Topics
Bimodal Environment	Bimodal IT layers Systems & Software Slow-Speed systems & software Fast-Speed systems & software Characteristics Slow-Speed characteristics Fast-Speed characteristics	- Pace-layered Application Strategy and Digital Transformation can be used to achieve agility - Terminology of the layers - Bimodal components, domains, systems & software, and integration patterns - Stable Layer is a facilitator for the Explorative domain - Software movement from the Explorative Layer to the Stable Layer - Layer characteristics
Middleware	Middleware components ESB components analysis ESB capabilities analysis CIP components analysis CIP capabilities analysis MSA components analysis MSA capabilities analysis	- ESB components & capabilities - CIP components & capabilities - MSA components & capabilities
SDLC methodologies & approaches	SDLC methodology SDLC tools & components SDLC tools & components analysis Impact CI/CD approach CI/CD tools & components CI/CD tools & components analysis CI/CD practices CI/CD practices analysis	- Use of traditional methodologies in the layers - Use of agile methodologies in the layers - CI/CD is upcoming within the Stable domain - CI/CD practices - SDLC tools & components - CI/CD tools & components - Impact of release management and

		prioritization on a Bimodal IT environment - Need for continuous alignment
Deployment	Deployment models On-premise & Data center hosting Cloud hosting	- Determination of a relationship between Bimodal IT and hosting - Hosting practices - Available deployment models for software - Components and capabilities of the hosting environments - Trend towards the cloud
Bimodal relevancy, adoption & implications	Bimodal future Roadmap, Roles & Processes	- Bimodal IT relevancy - Agile commitment through multi-disciplinary teams and change of business - Agile architecture versus Traditional architecture - Prioritization within the layers - IT Service Management - Technical debt

Table E Correlation between the codebook tabs, topics, and associated result clusters

Appendix 6 – Deployment of a Container Orchestration Platform

Figure A presents the infrastructure and how a Container Orchestration Platform can be deployed within a MSA environment.

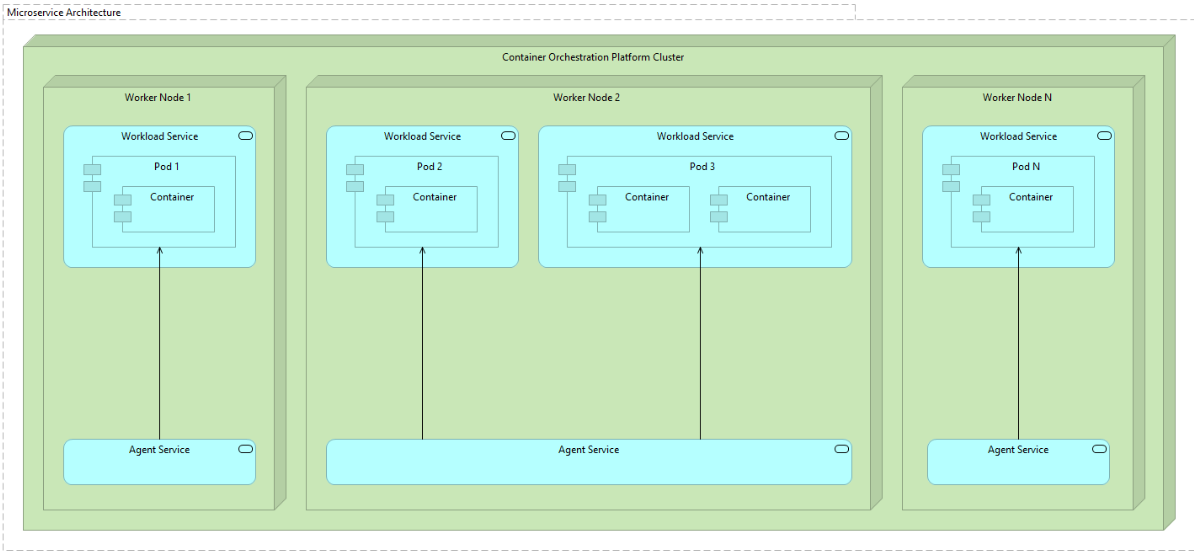

Figure A Fourth tier – Detailed view – Deployment model of a Container Orchestration Platform solution

Appendix 7 – On-premise environment

Figure B presents the components and capabilities of an on-premise deployment model. It also shows how the components and capabilities are cooperating and integrated.

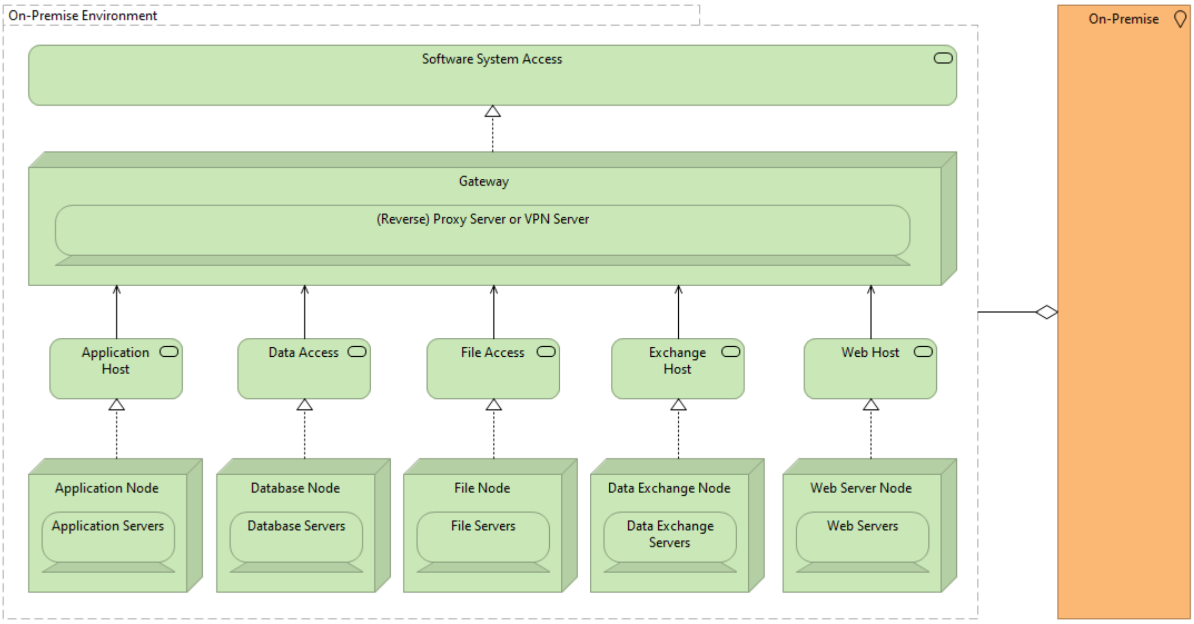

Figure B Sixth tier – Detailed view – Infrastructure of an on-premise environment

Appendix 8 – Data center environment

Figure C presents the components and capabilities of a data center deployment model. It also shows how the components and capabilities are cooperating and integrated.

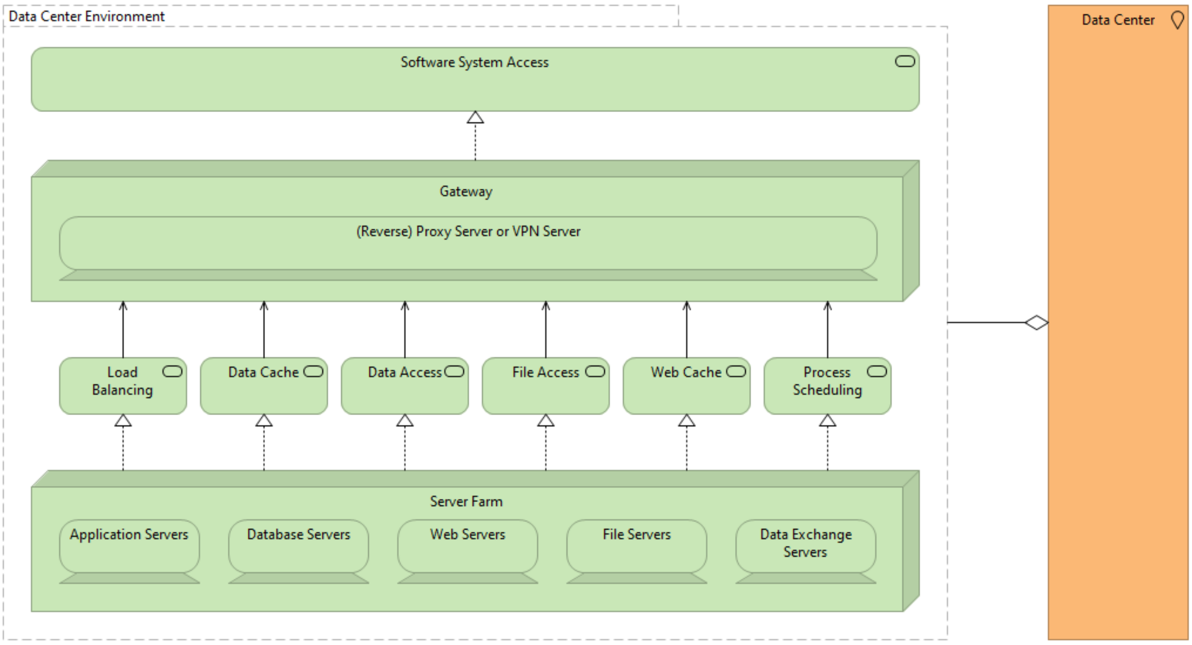

Figure C Sixth tier – Detailed view – Infrastructure of a data center environment

Appendix 9 – Cloud environment

Figure D presents the components and capabilities of a cloud deployment model. The model also shows how the components and capabilities are cooperating and integrated. In addition, it also portrays the subscription models that are associated with the offering of a cloud platform.

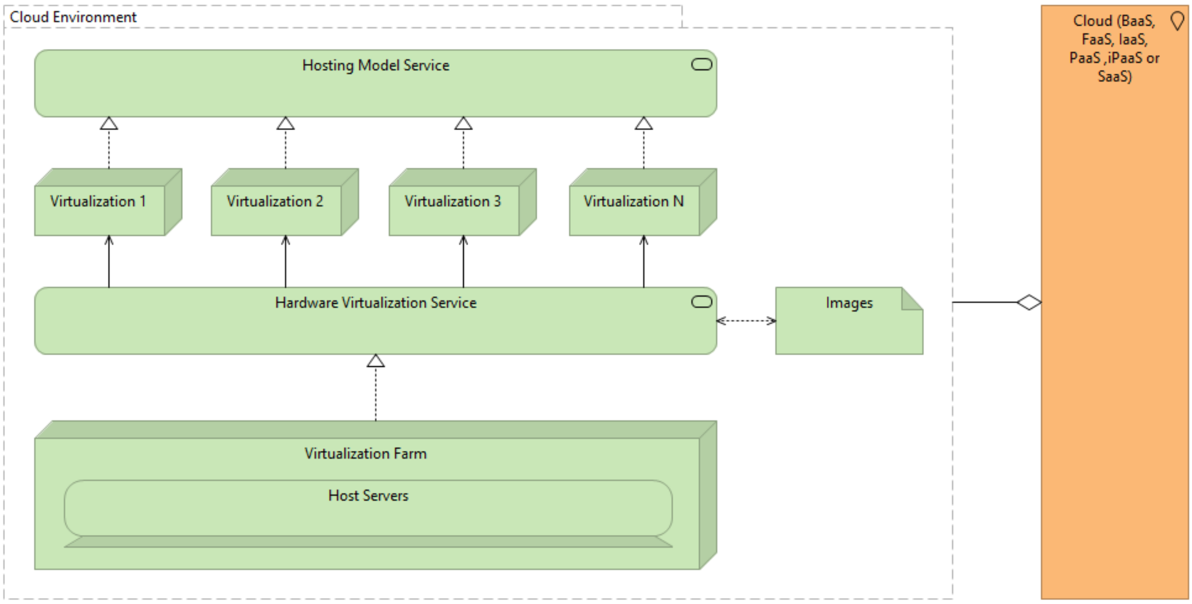

Figure D Sixth tier – Detailed view – Infrastructure of a cloud environment

Appendix 10 – Feedback keys

During the conventional content analysis 27 unique keys are defined within the feedback data. These unique keys are presented in table F.

Unique keys (#-tag)
#RA_is_clear
#RA_looks_complete
#RA_is_applicable
#integration_patterns_are_beyond_services
#recognizes_all_elements
#does_not_miss_any_critical_elements
#prefers_explanation_of_essential_components
#visualization_of_patterns
#RA_is_applicable_in_a_project
#RA_is_applicable_on_a_enterprise-wide_level
#RA_can_support_in_analysis_and_discussions
#RA_can_help_to_create_better_views_or_complete_existing_views
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture
#disclose_the_usage_of_Bimodal_IT_patterns
#RA_helps_to_distuigish_between_system_categories
#RA_can_help_to_create_outsourcing_strategie_for_each_system_category
#concern_regarding_adoption_of_RA_due_to_the_creation_of_multiple_views
#RA_can_be_used_as_a_baseline
#RA_visualizes_a_Bimodal_IT_environment
#Bimodal_IT_is_also_an_architecture_rather_than_only_a_concept
#detailed_views_required_within_the_RA_to_link_processes
#RA_can_help_to_set-up_an_architecture
#RA_can_help_to_set-up_a_Bimodal_IT_platform
#concern_regarding_adoption_of_RA_due_to_maintenance_of_RA
#concern_regarding_adoption_of_RA_due_to_IT_business_alignment
#remarks_to_improve_the_RA
#RA_supports_a_seperate_external_and_internal_API_platform

Table F Unique keys via conventional content analysis regarding the feedback data

Appendix 11 – Codes regarding the feedback data analysis

Table G presents the created codes to conduct the data analysis regarding the retrieved feedback.

Code ID	Code
1	RA is clear
2	RA looks complete
3	RA is applicable
4	Integration patterns are not solely based on services
5	All elements are recognizable
6	Crucial elements are still missing
7	Essential elements should be explained
8	Visualization of (hidden) patterns
9	RA can support in the analysis and discussions
10	RA can support in the creation of better views or complement current views
11	RA provides a proposal to visualize an IT landscape
12	RA provides guidelines how to model an architecture
13	RA can reveal if an IT landscape already uses Bimodal IT patterns
14	RA helps to differentiate and allocate an IT system into a particular system category
15	RA can support in outsourcing decisions
16	Adoption of the RA can fail due to a start from scratch
17	RA can be used as a baseline to visualize an IT landscape
18	RA supports the visualization of a Bimodal IT environment
19	Bimodal IT also implicates an architecture
20	RA can assist in the set-up of a proper architecture
21	RA can assist in the set-up or realization of a Bimodal IT platform
22	RA can be hard to adopt due to maintenance
23	RA can be hard to adopt due to a missing alignment between IT and the business processes
24	RA supports the usage of a separate external and internal API platform
25	Provided a remark
26	Did not provide a remark

Table G Codes to conduct the feedback data analysis

Appendix 12 – Correlation between feedback keys, codes, topics, and result clusters

Table H presents the correlation between the feedback keys, derived codes, related topics, and associated clusters.

Clusters	Topics	Keys	Codes
Impression regarding the RA	Impression	#RA_is_clear #RA_looks_complete #RA_provides_a_proposal_or_guideline_to_visualize_an_architecture #RA_visualizes_a_Bimodal_IT_environment	- RA is clear - RA looks complete - RA provides a proposal to visualize an IT landscape - RA provides guidelines how to model an architecture - RA supports the visualization of a Bimodal IT environment
Recognizability of the elements within the RA	Recognizability	#recognizes_all_elements #does_not_miss_any_critical_elements #prefers_explanation_of_essential_components #Bimodal_IT_is_also_an_architecture_rather_than_only_a_concept	- All elements are recognizable - Crucial elements are still missing - Essential elements should be explained - Bimodal IT also implicates an architecture
Applicability of the RA	Applicability	#RA_is_applicable #integration_patterns_are_beyond_services #visualization_of_patterns #RA_is_applicable_in_a_project #RA_is_applicable_on_a_enterprise-wide_level #RA_can_help_to_create_better_views_or_complete_existing_views #RA_helps_to_distuigish_between_system_categories #RA_can_be_used_as_a_baseline #RA_can_help_to_set-up_an_architecture	- RA is applicable - Integration patterns are not solely based on services - Visualization of (hidden) patterns - RA can support in the creation of better views or complement current views - RA helps to differentiate and allocate an IT system into a particular system category - RA can be used as a baseline to visualize an IT landscape - RA can assist in the set-up of a proper architecture - RA can assist in the set-up or realization of a Bimodal IT platform

		#RA_can_help_to_set-up_a_Bimodal_IT_platform #RA_supports_a_seperate_external_and_internal_API_platform	- RA supports the usage of a separate external and internal API platform
Advantages and challenges of the RA	Advantage Drawbacks	#RA_can_support_in_analysis_and_discussions #disclose_the_usage_of_Bimodal_IT_patterns #RA_can_help_to_create_outsourcing_strategie_for_each_system_category #concern_regarding_adoption_of_RA_due_to_the_creation_of_multiple_views #concern_regarding_adoption_of_RA_due_to_maintenance_of_RA #concern_regarding_adoption_of_RA_due_to_IT_business_alignment	- RA can support in the analysis and discussions - RA can reveal if an IT landscape already uses Bimodal IT patterns - RA can support in outsourcing decisions - Adoption of the RA can fail due to a start from scratch - RA can be hard to adopt due to maintenance - RA can be hard to adopt due to a missing alignment between IT and the business processes
Remarks on the RA	Remarks		- Provided a remark - Did not provide a remark

Table H Correlation between the keys, codes, topics, and associated result clusters

Appendix 13 – Expert interviews

Candidate 1 – DEV-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I have four years of working experience. I am working as a medior consultant/developer {DEV-01) at a consulting company {Consulting-01} that is centered around Microsoft products. My main focus is on microservice and API (back-end) developments. Currently, I am located at a client as a senior software developer.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The company's goal and focus are to deliver innovative services and solutions to enterprises worldwide using the Microsoft platform. There are around 39.000 professionals in over 80 locations in 24 countries. My work domain is related to Azure and Dynamics 365 products.

#IT_organization_focus
#IT_capability
#candidate_current_role

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I see Bimodal Architecture as an infrastructure and software platform that is divided into two different layers, a Digitized and a Digital layer. Both layers are communicating with each other via services. This can be microservices or web services.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_MSA_services

The Digitized (back-end) layer and associated services are related to commodity. The incorporated systems store and manage business-critical data and supports the core business processes. Consequently, the layer concentrates on efficiency and excellence and forms the Operational Backbone of an organization.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_characteristics
#concept_digital_transformation_operational_backbone

The Digital layer realizes the Digital Platform by focusing on innovative and explorative solutions to enhance the user experience. To achieve this goal, new features must be released early and often. The systems are supported via agile development approaches, preferable CI/CD to increase customer intimacy and improve the time-to-market.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics
#concept_digital_transformation_digital_platform
#methodology_agile_framework_continuous_integration_and_delivery

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The main components of a Bimodal Architecture are:

- Digital layer. This layer focuses on experimentation, exploration, and innovation. The associated systems and applications are continuously in development and can change rapidly based on the volatile requirements;
- Digitized layer. This layer comprises the core systems and solutions that create operational value. The associated systems support the critical business processes and processes that hardly change. When these systems are compromised, it can jeopardize the business continuity;
- Communication layer. This layer is essential within the Bimodal environment because the incorporated middleware mechanisms connect, integrates, and executes the communication within and between the Digitized and Digital layers through services. This layer makes it possible to facilitate and exploit the Digital layer. However, note that the underlying service-based architecture itself is stable.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, it is recognizable. The presented models are reflecting the layers, as mentioned in question 1 and 2. However, I have two comments regarding the presented models. The first comment is related to the typology of the layers. The second comment is related to the systems and solutions responsible for creating the operational value.

#bimodal_IT_layers_terminology

The typologies “Slow-Speed” and “Fast-Speed” are retrieved from the Two-Speed IT approach defined by McKinsey. The approach implies a Fast-Speed customer-centric front-end that runs alongside a Slow-Speed transactional back-end. This conceptualization is partly true. The reason is that some of the Microsoft solutions are customer-centric but are part of the Digitized environment, such as Dynamics 365 CRM, Power Platform, and Mecoms. While Power Apps, Sitecore, SharePoint, Office 365, Commerce, etc., are part of the Digital environment. As mentioned in question 1, the Digitized platform supports and handles the core business objects and data. In contrast, the Digital environment is mainly focused on exploration and innovation and is established through the Digitized layer. Therefore, the typology of the Two-Speed IT concept is misleading and can create confusion within an organization.

#bimodal_IT_layers_terminology
#Slow-Speed_systems_and_software
#bimodal_IT_layers_Slow-Speed
#Slow-Speed_characteristics
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics

The second comment is related to the systems that are creating operational value. Take, for example, sensors or certain power apps that are especially created to register the time spent on activities or costs. These solutions are modern and innovative, but at the same time, they provide major insights within the operational activities that can help to control or to make better decisions. Therefore, these systems and solutions are part of the Digitized environment instead of the Digital layer. Also, these systems do not change a lot once they are available in production.

#Slow-Speed_stabilized_systems_of_innovation

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

Yes. The following systems can be found within the Digitized layer:

- Business Central;

- Dynamics 365 CRM;
- Power Platform;
- Mecoms;
- Power BI;
- SAP S/4HANA;
- Oracle E-Business Suite;
- Sensor APIs;
- Time and Expense App;
- Booking and Reservation App;
- Mileage Tracking App;
- Other internal APIs. To create and leverage operational value.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Systems that can be found within the Digital layer are:

- Power Apps;
- Sitecore;
- SharePoint;
- Office 365;
- Commerce;
- Teams;
- Skype;
- Web Apps;
- iOS Apps;
- Android Apps;
- Other external APIs. To provide additional services and offerings to various stakeholders outside the organization.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of the Digitized layer are:

- Stable and reliable;
- Security is an important and ongoing aspect;
- Supports the main business processes, objects, and data;
- Does not change a lot once deployed into production;
- Less error-prone;
- Makes use of agile and traditional methodologies to develop a system or solution;
- Delivers operational value;
- Solid test processes;
- Problem-solution design;

- Facilitator for the Digital layer by exposing services.

#Slow-Speed_characteristics

The characteristics of the Digital layer are:

- Less stable;
- Flexible. The solutions can change rapidly and often;
- Stores and contains minimal or no critical data;
- Error-prone is high due to the experimental and explorative nature;
- Supports only an Agile way of development;
- Rapid delivery oriented. Deployment happens more often;
- Continuous integration/deployment in place to increase time-to-market;
- The solutions are centered around interaction and experience;
- Focus on experimentation, exploration, and innovation;
- New technology-oriented;
- Trend to have only a development and production environment;
- Requirements and planning changes constantly.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Communication layer can comprise a service bus, a microservice environment, or even a cloud integration platform. The cloud integration platform is a lightweight integration mechanism to integrate systems running on any deployment model.

The (technical) components of the service bus (Biztalk) are:

- Message Builder;
    - Set-up of Adapters;
    - Creation of Services;
    - Set-up of Mediation Flows;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;

- Integration Controller/Integration Gateway;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Event Handler;
    - Mediation flow Manager;
    - System Landscape/Tenant Manager;
    - Auditor;
    - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;

    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

Also, Microsoft provides a solid, reliable, and scalable platform for the integration of on-premise and cloud-based solutions. This platform, named Azure Integration Services, brings the API Management, Logic Apps, Service Bus (including the integration with the Biztalk Server), and Event Grid functionalities together. The Service Bus is used to connect the on-premise and cloud-based systems and services through highly secured messages. The Event Grid ensures the connection between Azure and third-party services through a fully managed event-routing service based on a publish-subscribe model. Logic Apps is used to automate various workflows. And the API Manager is used to publish APIs. These APIs can be used by internal and external developers when connecting to back-end systems and services.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of the cloud integration platform are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Mediation Flows;
        - Integration/Message Builder;
            - Set-up of Mediation Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;

        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;
            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;

- API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Translation;

- Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

The best practice to run microservices is through the use of container technology. These containers can be managed through a container orchestration platform. The container orchestration platform is deployed as a cluster and consists of a set of nodes. These nodes are hosting pods, which represent a set of running containers.

#bimodal_IT_layers_Communication_layer_MSA

The container orchestration platform comprises the following (technical) components:

- Scheduler;
- Controllers;
- Key-value Database;
- API server;
- Several Nodes;
    - Hosts several Pods;
- Resource Manager;

- Container Orchestration engine.

Other components in the infrastructure are:

- Storage;
- Queues;
- Container registry;
- Container repository (optional);
- Network;
- Provisioning;
- Logs;
- Infrastructure Security.

#Communication_layer_MSA_components

The microservice environment realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Failover;
    - Microservice Security;
    - Configuration Management;
    - Networking/IP;
    - Analytics;

  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Container;
  - Microservices.

#Communication_layer_MSA_capabilities

Interviewer: First of all, thank you for the extensive explanation regarding the service bus and the microservice platform at its components. I also got to know that Azure supports serverless computing. Can you maybe briefly explain this from a deployment model perspective?

Serverless computing is the ability to run custom code on-demand and at scale in the cloud.

#deployment_model_cloud_serverless_computing

For the cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology, like Azure Functions or Azure Storage Account, and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are offered as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- or Hybrid.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

For the Digital layer applies that the applications and systems are mainly delivered through CI/CD. In some cases, another agile approach is used to deliver an increment or a solution.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery

The Digitized layer uses a mix of SDLC approaches to deliver the relevant applications, systems, and solutions. Foremost, Waterfall is used to develop and deliver the core systems. The systems and solutions generating operational value are delivered through an agile approach, like Scrum.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_traditional_framework_waterfall
#methodology_agile_framework
#methodology_agile_framework_Scrum

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The potential impact of a Bimodal Architecture environment is related to release management. The various teams need to be continuously aligned. This continuous alignment will help manage and oversee the various deployments, reduce interoperability issues, and keep the several teams informed.

#impact_release_management
#impact_continuous_alignment

9: Do you use different methodologies across the different layers?

Yes. The systems within the Digital layer are mainly delivered through CI/CD. The same applies to the related microservices. Other back-end and API developments in the Communication layer are developed and released in an agile way (mainly Scrum).

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#bimodal_IT_layers_Communication_layer
#methodology_agile_framework_Scrum

The systems and solutions within the Digitized layer apply that they are developed and delivered through a mix of traditional and agile SDLC approaches.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

Microsoft has released Azure DevOps to enable CI/CD. CI/CD is mainly used within the digital and microservice environments, such as Azure.

#continuous_integration_and_delivery_systems_and_software
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

CI/CD ensures early and often releases. Therefore, it is very suitable within the Digital layer. Generally, the Digital layers' systems and solutions do not impact the underlying and often critical business processes.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery

10b: Which layers are affected by CI/CD?

The Digital layer and the microservice environment, which is part of the Communication layer.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

Every build starts locally. A developer can use various IDEs to perform the development. Once the code is ready, it is submitted into a repository. From this point, Azure DevOps comes into play.

#continuous_integration_and_delivery_systems_and_software

Azure DevOps comprises the following components:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, and Work items;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;

- Azure Pipeline. This functionality has an integration with Jenkins. This functionality is used to automate the build (via Maven, Gradle, Visual Studio, etc.), test (via Selenium, FitNesse, JUnit, etc.), and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. Azure DevOps can use Azure Functions, Azure Service Fabric, Azure Kubernetes Service, or Docker containers as a deployment technology;
- Azure monitor. This tool is used for monitoring and analysis.

Also, Postman is used to test and mock the APIs. Postman provides the following functionality:

- Newman. This functionality is used to automate and test the various API calls.

#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

In the case of CI/CD, the following configuration practices are applicable:

- It is important to have a CI/CD strategy in place;
- A recovery strategy should be in place;
- Increments should be small to minimize the branches;
- The developer should test the increment rigorously on his/her local environment;
- Only required features and functionalities should be build and released;
- Within the pipeline, the proper tests should be configured and kicked off;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an automation server that can automatically build, test, publish and deploy the increment;
- A verification step should be in place before each deployment;
- Configurations and passwords should be easily accessible and always up-to-date;
- Monitoring, analytics, and visualization need to be in place.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

Various tools and components are used to develop and deliver systems, applications, and solutions.

- A tool to manage the planning and requirements of the various artifacts;
- A project, service, and ticket management system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;

- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

We are using the following tools to realize the various systems, solutions, increments, and APIs:

- ServiceNow and Azure Boards;
- For specifications and collaboration:
    - Office 365;
- SharePoint as DMS;
- Git as code repository;
- Development tools:
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Eclipse (.NET and C/C++ developments);
    - Azure DevOps;
    - Xcode 12;
    - Swift UI;
    - Android Studio;
- Automation server:
    - Jenkins;
- Test tools:
    - Azure Test Plans (part of Azure DevOps to perform manual and load tests);
- Deployment tools:
    - App Service;
    - Azure Functions (serverless computing);
    - Container Instances, such as Docker;
    - Azure Service Fabric;
    - Azure Kubernetes Service;
- Monitoring tools:
    - Application Insights;
    - Azure Alerts.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

For so far, I know there is no relationship between the deployment model and the layers within a Bimodal Architecture. The systems of the Digital layer are commonly deployed in the cloud. The majority of the systems within the Digitized layer are hosted on-premise or in a data center. The reason for choosing this deployment model is that an organization wants to control the systems because most of these systems support critical business processes and are storing the associated data. However, since the last couple of years, there is a trend noticeable towards the cloud. The Digitized layer systems are deployed in a private or hybrid cloud, so the organizations can take advantage of the cloud and still control their core systems.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud
#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#deployment_model_cloud_Private
#deployment_model_cloud_Hybrid

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

There is a movement taking place towards the cloud. However, some systems will never move towards the cloud due to their importance for a particular organization or unit.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

That is depending on the IT strategy and IT choices of the client.

16: What is the future roadmap of the IT landscape of your organization?

Not applicable. We can only advise, provide guidance and help to shape the IT landscape of our clients.

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Bimodal Architecture will stay relevant as a concept. However, the focus will move from rigid systems and applications to more modular systems and solutions within the cloud. However, there will be a clear split between critical, core systems and customer-facing digital solutions. Also, due to the increase in test automation, the builds will be more reliable, which can eventually speed-up the development and delivery of the increments in both layers.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

In the Digitized layer, the planning and initial designs are mainly conducted by the business and architect. Once the phase is finished, different IT organization teams are involved in developing and delivering the solution. This can be very time-consuming and compromise the requirements over time.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_roles
#concept_bimodal_IT_planning
#concept_bimodal_IT_designs
#concept_bimodal_IT_traditional_teams

For the Digital layer applies that there is a cross-disciplinary team available for each domain/capability to deliver the increments in a short period. The disciplinary teams can be formed with the cooperation of several third parties.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_agile_teams

This also means that an agile architecture differs from a traditional architecture. In practice, it means that a sub-architecture or another viewpoint needs to be created to facilitate the iterative way of working.

#concept_bimodal_IT_traditional_architecture
#concept_bimodal_IT_agile_architecture

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

In the Digitized layer, the priority setting is conducted by the business and follows the processes of IT Service Management. Before a system can be built or adjusted, a detailed impact analysis must be delivered by the IT organization or third-party to determine the IT projects' risks and costs. Once a steering community approves the estimation, the IT project can start with the systems' development and delivery.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_traditional_teams
#IT_Service_Management
#concept_bimodal_IT_estimations

In the Digital layer, the priority is set by the PO. The costs of the associated disciplinary teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

All systems and solutions have some technical debt. However, in the Digital layer, mistakes and problems can be reversed very fast. Mostly, within another sprint. Also, there are enough opportunities to remove the technical debt in future sprints.

The systems and solutions within the Digitized layer apply that they can have a high technical debt due to the systems' complexity and robustness.

#concept_bimodal_IT_technical_debt

Candidate 2 – EA-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I have fifteen years of working experience in the IT domain. I work for a consulting company {Consulting-01} that implements and delivers Microsoft products to their clients. I work for this company since the start of my career in 2005. In the main time, I have fulfilled multiple roles and executed several projects for various clients. For the last five years, I am fulfilling the role of an Enterprise Architect {EA-01} and help different clients to adopt digitalization during their transformation journey.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The company's goal is to deliver innovative services and solutions of the Microsoft platform to enterprises worldwide. There are currently around 39.000 professionals in over 80 locations in 24 countries to help their clients transform.

#IT_organization_focus
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal Architecture is a concept that is introduced by Gartner in 2013. Gartner identifies two modes, in which one mode is focused on the existing legacy systems, while the other mode is focused on digitalization. Rather than Gartner's description, I see Bimodal Architecture as a cohesive architecture that comprises three layers. There is a stable architecture layer. In this layer, the information systems support the business processes and provide stability, predictability, and metrics to improve operations. Also, there is an explorative architecture layer. This layer comprises fast-moving applications and applications that enhance the experiences (mobile applications, portals, and web apps). This layer can also be called the "Digital layer." And there is a layer in between to support the communication and connectivity. This layer is essential because it allows the Stable layer to expose its functionalities as services and APIs and enforces the messages into functionalities and features to facilitate the Explorative layer.

#bimodal_IT_layers_Slow-Speed

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_MSA_services

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

As already discussed, Bimodal Architecture is an IT platform that exists of three layers:

- The Stable layer comprises systems that provide, support, and improves "stability," "predictability," "robustness," "operational value," "business processes," and stores "critical data." This layer includes systems, such as SAP S/4HANA, Dynamics 365, Salesforce, Oracle eBS, etc.
- The Explorative layer comprises systems that support, improves, and enhances "interaction," "experience," "experimentation," and deals with fast-changing requirements. This layer includes systems, such as Sitecore, Power Apps, SharePoint, E-commerce, Office 365, Android Apps, iOS Apps, etc.
- The Communication layer is a service-based environment. This layer connects, integrates, and executes the communication within and between the Stable and Explorative layer through services. This can be web services or microservices.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_MSA

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can link the presented models to the mentioned layers.

There is, however, one point that I like to discuss. In the first layer of your first model, you mention the information systems that support the business processes and operations. In the second layer, you have mentioned the Communication layer, and the third layer represents the Explorative layer. The second model shows the integration patterns between the three layers. Until now, the models are straightforward.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed

#bimodal_IT_layers_Communication_layer

However, you miss the fact that the Stable layer can also comprise certain apps and innovative solutions. It does not solely include information systems. Currently, you are only telling one part of the story.

#Slow-Speed_stabilized_systems_of_innovation

Also, the service-based environment of the Communication layer should be present in all layers. For example, a service bus can be used to connect several information systems internally within the Stable layer while also being used to expose an internal API to realize and facilitate an App that can improve the organizational operations. Consequently, the same service bus can expose an external API to enable a customer App or an App for businesses.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB

The same applies to the Explorative layer. We have scenarios in which iOS apps are integrated with portal functionalities to provide a full customer experience.

#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_MSA

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The Stable layer includes the following systems and software:

- Dynamics 365 Business Central;
- Dynamics 365 Finance & Operations;
- Dynamics 365 Customer Service;
- Dynamics 365 Field Service;
- Dynamics 365 Human Resources;
- Dynamics 365 Supply Chain Management;
- Dynamics 365 CRM;
- Power Platform;
- Power BI;
- SAP S/4HANA;
- SAP C4C;
- Salesforce;
- Oracle eBS;

- COBOL solutions;
- TeamCenter;
- Internal APIs;
- Web Apps, Mobile Apps, and extensions for the internal organization.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The Explorative layer includes the following systems and software:

- Power Apps;
- Sitecore;
- SharePoint;
- E-Commerce;
- Office 365;
- Teams;
- Web Apps;
- iOS Apps;
- Android Apps;
- Windows Phone Apps;
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of the Stable layer are:

- Stable, reliable, and predictable;
- Robust;
- Security is a necessity;
- Standardization is an important aspect;
- Supports business processes, objects, and data;
- Does not change a lot once deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Failover strategy in place to guarantee business continuity and availability;
- Makes use of agile and traditional approaches to develop and deliver a solution/system;
- Solid test processes;
- Problem-solution design;
- Implementation costs are high.

#Slow-Speed_characteristics

The characteristics of the Explorative layer are:

- Delivers business value at each release or iteration;
- Flexible. A solution can change rapidly and often;
- 24/7 Available;
- Stores and contains minimal or no critical data;
- Errors are incorporated due to the experimental and explorative nature;
- Only an Agile way of development is supported;
- Rapid delivery oriented. Often and frequent deployments;
- Focus on innovation, experimentation, and exploration;
- Continuous integration/deployment in place to increase time-to-market;
- The solutions are mainly centered around interaction and experience;
- Requirements and planning changes constantly;
- Trend to have only a development and production environment.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Communication layer comprises a service-based environment. This can be a service bus (Biztalk Server), a microservice environment (Azure), or even both.

#bimodal_IT_layers_Communication_layer_ESB
#Communication_layer_ESB_components
#bimodal_IT_layers_Communication_layer_MSA
#Communication_layer_MSA_components

The (technical) components of the service bus are:

- Message Builder;
    - Set-up of Adapters;
    - Creation of Services;
    - Set-up of Mediation Flows;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;

    - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Event Handler;
    - Mediation flow Manager;
    - System Landscape/Tenant Manager;
    - Auditor;
    - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;

- Dynamic Routing;
- Message Validation;
- Reliable Messaging;
- Pass-Through of Messages;

- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - XML;
  - FTP and SFTP;
  - SMTP;
  - Flat File;
  - Database Adapter;
  - 3rd Party Adapter;
  - Custom Adapter.

#Communication_layer_ESB_capabilities

As an alternative for the Biztalk Server, Azure also provides a single platform to integrate and connect on-premise and cloud-based solutions. This platform is called Azure Integration Services and combines the API Management, Logic Apps, Service Bus, and Event Grid functionalities.

The Service Bus is used to connect the on-premise and cloud-based systems and services through highly secured messages. The Event Grid ensures the connection between Azure and third-party services through a fully managed event routing service based on a publish-subscribe model. Logic Apps is used to create workflows and support automation. And the API Manager is used to publish and manage APIs. These APIs can be used by internal and external developers when connecting systems, applications, and services.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of the cloud integration platform are:

- Cloud Integration Service;
  - Message and Event Hub;
    - Repository for Service Discovery and pre-build Mediation Flows;
    - Integration/Message Builder;

Universiteit Leiden

- Set-up of Mediation Flows;
- Editing of Flows;

- System Landscape/Tenant Manager;
- Transaction Handler;
- Configuration Manager;
- Security Manager;
  - Set-up of Message Security;
  - Set-up of Encryption;
- Access Controller;
  - Set-up of Authorization;
- Event Handler;
  - Set-up of Events (Event Producer and Event Listener);
- Message Queues;
- Exception handler;
- Logger;
- Auditor;

- Connectivity Hub;
  - Repository for Adapter Discovery;
  - Adapter Builder;
    - Set-up of Adapters;
  - Adapter Configuration Manager;
  - Adapter Security Manager;
    - Set-up of Authentication;
  - Adapter Event Handler;
  - Workflow Builder;
    - Set-up of Workflows;
    - Editing of Workflows;
  - Access Controller;
    - Identity Access Manager;
    - Set-up of Authorization;
  - Logger;
  - Auditor;
- Integration Engine;
  - Adapter Engine;
  - Orchestration Engine;
  - Transformation Engine;
  - Routing Engine;
  - Rule Engine;
  - Publisher and Subscriber Engine;

- API Management Service;
  - Repository for API Discovery;

    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;

- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

The best practice to run microservices is through the use of container technology. These containers are managed through a container orchestration platform. The container orchestration platform is deployed as a cluster and consists of a set of nodes. These nodes are hosting pods, which represent a set of running containers.

#bimodal_IT_layers_Communication_layer_MSA

The container orchestration platform comprises the following (technical) components:

- Scheduler;
- Controllers;
- Key-value Database;
- API server;

- Several Nodes;
  - Hosts several Pods;
- Resource Manager;
- Container Orchestration engine.

Other components in the infrastructure are:

- Storage;
- Queues;
- Container registry;
- Container repository (optional);
- Network;
- Provisioning;
- Logs;
- Infrastructure Security.

#Communication_layer_MSA_components

The microservice environment realizes the following capabilities and functionalities:

- Container Orchestration Platform;
  - Container Orchestration;
  - Container clustering;
  - Image Discovery;
  - Container Security;
  - Network settings;
  - Key Vault;
  - Load Balancing;
  - Scheduling;
  - Rollouts and Rollbacks;
  - Self-Healing;
  - Workflow Management;
- API Management;
  - Service Discovery;
  - Messaging;
  - Routing (of services);
  - Call handling;
  - Transformation;
  - Rate Limits;
  - Traffic Management;
  - Throttling;
  - Caching;
  - Circuit Breaker;
  - Failover;

Universiteit Leiden

- Microservice Security;
- Configuration Management;
- Networking/IP;
- Analytics;
- API Version Management;
- Monitoring and Logging;
- Policies and Contracts;

- Container;
    - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

For establishing systems and software within the Explorative layer, it is common to use CI/CD or another agile methodology, like Scrum. Because, with these approaches, we can anticipate on the fast-changing requirements.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum

For the Stable layer, a mix of approaches can be used to deliver the systems and software. Commonly, the Waterfall methodology is used to design, build, and test software. Once the systems and software are put into production, an agile method is used to maintain and exploit it further.

For complex and big IT projects, it is even common to use a project management approach, like PRINCE2.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

I see two major implications when we talk regarding software delivery within a Bimodal environment.

The first impact is related to release management. During the release of a feature or increment, all three layers need to be considered. For example, when we talk about developing a new API, the functionality must first be exposed from an information system. The messages need to be mapped and transformed within the Communication layer. Subsequently, we need to invoke the message in the new App, to use the respective object and related data. Every development for this API needs to be released at once to ensure a working feature or increment. Therefore, it is a necessity to keep the various teams aligned. To achieve this, there are mechanisms in place to encourage daily communications between cross teams and weekly alignment sessions with multiple teams and stakeholders.

#impact_release_management
#impact_continuous_alignment
#impact_communication

The second impact is related to priority setting. Suppose an App of the Explorative layer needs to be adjusted because of a new attribute. In this scenario, there is a reasonable chance that also the corresponding API needs to be changed. Therefore, the priority should be set for both layers and their associated teams and should not only be applied on the App.

#impact_prioritization

By considering these two points, a lot of problems and issues can be mitigated.

9: Do you use different methodologies across the different layers?

Yes. The increments of the Explorative and Communication layers are generally delivered through CI/CD. However, a small amount of the systems and software are developed and delivered using an agile method, such as Scrum or XP.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery

The systems and software of the Stable layer are delivered through a mix of SDLC approaches.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

We have Azure DevOps in place to ensure and enable CI/CD. CI/CD is used to realize and facilitate systems and software within the Explorative layer and to establish microservices within the Communication layer.

#continuous_integration_and_delivery_systems_and_software
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

Through the Azure DevOps Pipeline functionality, increments can be automatically built and tested. Once these activities are completed, the Azure DevOps Release functionality ensures that the increments are automatically released and deployed towards the configured environments.

#continuous_integration_and_delivery_systems_and_software_functionalities

10b: Which layers are affected by CI/CD?

As already discussed, CI/CD is used to realize and facilitate systems and software within the Explorative layer, and it enables microservices within the Communication layer.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The development of increments starts locally in Visual Studio. Once the coding is finished, the source code is submitted into a Repository, in our case Git.

After this, Azure DevOps comes into play.

#continuous_integration_and_delivery_systems_and_software

The components of the Azure DevOps tool are:

- Azure Boards. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug tracking;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. Azure Pipeline is an out-of-the-box automation server. However, this functionality can also be integrated with Jenkins and other third-party automation servers. The Azure Pipeline functionality is used to automate the build, test, and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. Azure DevOps can use Azure Functions, Azure Service Fabric, Azure Kubernetes Service, or Docker containers as a deployment technology;
- Azure Monitor (Application Insights and Azure Alerts). This tool is used for monitoring and analysis.

Microsoft teams is used to notify the developer whether the build succeeded or failed.

To test and mock the APIs, Postman is used. Postman provides the following functionality:

- Newman. This functionality is used to automate and test the various API calls.

#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

To enable and ensure CI/CD, the following practices are essential:

- It is crucial to have a CI/CD strategy in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- Only required features and functionalities should be build and released;
- A verification step should be in place before every deployment;
- It needs to be ensured that configurations and passwords are easily accessible and always up-to-date;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

There are several tools and components required to develop and deliver the systems and software within the Stable and Explorative layers:

- A tool to manage the planning and requirements of the various artifacts;
- A project, service, and ticket management system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The components are embedded in the following systems and applications:

- ServiceNow and BMC Remedy is used for the creation of tickets and to support IT Service Management activities;
- Tracecloud for requirements management;
- SharePoint to store all documentation;
- For the development and configuration of increments, systems, and software, the following tools are used:
    - SAP uses ABAP as their development platform;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - JavaScript, PHP, and HTML5;
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Azure DevOps;
    - Microsoft Intune is used to configure and maintain mobile devices (mobile device management (MDM) and mobile application management (MAM));
- We are using several Test Management Tools;
    - Azure Test Plan;
    - Micro Focus UFT one and ALM / Quality Center for testing SAP, Azure, and Java software;
- For the deployment, we are using the following tools:
    - App Service;
    - Azure Functions (serverless computing);
    - Container Instances, such as Docker;
    - Azure Service Fabric;
    - Azure Kubernetes Service;
    - CHARM is used in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for SAP IFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
    - Apple Business Manager is used to deploy iOS devices;
- Monitoring tools:
    - Application Insights;
    - Azure Alerts;
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

From my perspective, there is no relationship determinable between the deployment/hosting models and the layers. In practice, however, we notice that the systems and software of the Stable layer are deployed into an on-premise or in a data center environment. For the Explorative layer applies that the systems and software are mainly deployed into the cloud.

#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

Interviewer: Thank you for the answer. Can you also elaborate a little bit on the technical infrastructure of the mentioned deployment models?

Yes, of course. For the cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology, like Azure Functions or Azure Storage Account, and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- Hybrid.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

In an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc.

For security reasons, the Server Farm servers and services can be reached and executed via a private network established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

I also like to add that recent technological developments have changed the deployment strategies of organizations. Nowadays, organizations are choosing more and more to host their systems and software in the cloud.

#deployment_model_movement_towards_cloud

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

As already discussed, the cloud is becoming the common deployment practice for every system.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

This depends on the client.

16: What is the future roadmap of the IT landscape of your organization?

This also depends on the client.

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I foresee a bright future for the concept. The reason is that the IT landscapes of our clients are becoming more complex. IoT, machine learning, artificial intelligence are upcoming trends and change how customers, suppliers, and partners experience the offered products and services. Therefore, these trends demand a different IT architecture with a clear split between the Stable and the Explorative layers and associated integration patterns.

#concept_bimodal_IT_future
#IT_trends

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The main impact is on the team set-up. Within the Stable layer, the business process owner, business analyst, and architect conduct the plan and design processes. Once these phases are performed, different teams are involved in each phase of the development process.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_roles
#concept_bimodal_IT_planning
#concept_bimodal_IT_designs
#concept_bimodal_IT_traditional_teams

For the Explorative layer applies that there is a stable disciplinary team available for each domain/capability to deliver the increments in a short period. The disciplinary teams can be formed with the cooperation of several third parties.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_agile_teams

In addition, an agile architecture differs from a traditional architecture. The main question that needs to be asked is: does the current architecture provide the proper procedures, principles, and guidelines to adjust a system in an agile way? If the answer is no, then a sub-architecture or another viewpoint needs to be designed to facilitate the iterative way of working.

#concept_bimodal_IT_traditional_architecture
#concept_bimodal_IT_agile_architecture

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The business conducts the priority setting for the systems and software within the Stable layer. Before a system can be built or adjusted, a detailed impact analysis must be delivered by the IT organization or third-party to determine the IT projects' risks and costs. Once a steering community approves the estimations, the IT project can start with the systems' development and delivery.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_estimations

In the Explorative layer, the priority is set by the PO. The costs of the associated disciplinary teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

The systems and software within the Explorative layer can have some technical debt. However, mistakes and problems can be reversed fast. Mostly, within another sprint. Also, there are enough opportunities to remove the technical debt in future sprints.

For the systems and software of the Stable layer applies that they can have a high technical debt due to the complexity and robustness of the systems and software.

#concept_bimodal_IT_technical_debt

Candidate 3 – ITL-01

General questions:

1: Personal information: years of experience, and current function/position/role within the company or project.

I have twelve years of experience in SAP and associated connectivity, such as IS-U, CRM, SD, CS, Basis, and middleware. Currently, I am fulfilling the role of an Integration and Technical Lead {ITL-01} within the SAP domain, which is part of the internal IT organization.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

Our organization operates within the Utility industry {Utility-01) and is responsible for the exploitation and maintenance of a part of the grid within The Netherlands. The IT capability exists of 52 Scrum /BizDevOps teams (each team has 7-8 members) in which all domains are represented (from Architects to API developers).

#company_profile
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

For me, Bimodal Architecture is a platform that is divided into three IT platforms. The first platform focuses on the organizations' information systems to standardize and optimize the business processes, like SAP ECC (System of Record) and SAP C4C (System of Differentiation). This platform also comprises the stabilized engagement applications (Systems of Innovation), like the Stedin Safety App, sensor APIs, and other internal APIs. The second platform is a service-based environment that connects the different core systems, establishes the interaction with the stabilized engagement systems, and realizes the interfaces between the core systems and the fast-changing applications by transmitting and receiving various services/APIs. This platform also establishes and supports the communication outside the organization, like with EDSN, Mijnaansluiting.nl, and other third parties. The third and last platform comprises the fast-changing applications (Systems of Innovation), like the customer portal, web apps, and other digital offerings.

#bimodal_IT_layers_Slow-Speed

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer
#concept_Pace-layerd_Application_strategy

Interviewer: That's an interesting answer. Can you maybe explain the first platform and especially the part of the stabilized engagement applications? Because my assumption is/was that the systems of engagement/interaction are always part of the fast-changing platform.

Well, your assumption is partly correct. Let's take the Stedin Safety App as an example.

One of our organizational goals is to work safely. The App supports this goal by providing information to the engineers regarding how to work safely with and around energy and gas constructions and components. The App includes regulation and environmental guidelines, work instructions, etc., to create awareness, avoid hazards and accidents.

As per the example, the App and its features are very stable. When the App is down or not available due to an error or maintenance, it will not impact or jeopardize anything. In the worst case, it will cause some inconvenience. Therefore, there is no need to change or deploy a new version of this App frequently. Simultaneously, we cannot mark this application as a core system because it does not support or impact any (core) business processes. Conclusively, we can say that the App is fulfilling its objective by delivering a significant contribution in creating awareness regarding safety. Therefore, this App is part of our stabilized engagement platform instead of the fast-changing platform.

#IT_organization_focus
#Slow-Speed_stabilized_systems_of_innovation

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

As already mentioned in question 1, the main components of the first platform are:

- Core Systems, like SAP ECC, SAP IS-U, and SAP C4C;
- SAP connections. We use the internal SAP interoperability mechanisms, like RFC, EDI (IDocs), and adapters to connect to the various SAP systems, like SAP IS-U and SAP ECC;
- Proxies. This mechanism is used to create services and to communicate with the "outside world," stabilized engagement systems, and other core systems within the same IT platform;
- SAP Enterprise Services. These out-of-the-box services are also used to communicate with the "outside world," stabilized engagement systems, and other core systems within the same IT platform;

- Specific middleware technologies, like SAP PI/PO and SAP Cloud Platform (SCP). The integration between the SAP systems, SAP SaaS solutions, and non-SAP systems are established through SCP and associated IFlows;
- Stabilized engagement applications and associated (internal) APIs, like the sensor APIs, Smart Meter Reading API, and Stedin Safety App.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Communication_layer_ESB
#Communication_layer_ESB_components

The second platform comprises services buses, like SAP PI/PO, Mule ESB, etc., or a cloud integration platform, like SCP. It can also comprise a microservice environment, like Azure.

#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_MSA

Interviewer: What is the difference between the mentioned tools? Why are there different service-based platforms within the second layer/platform while they are performing essentially the same thing?

Before the migration to the cloud, we had the SAP systems running in a data center. To establish the connectivity between the multiple systems and software, we used SAP PI/PO to transform the IDocs and SOAP messages into JSON messages and vice versa. Through this system, we could connect to other core systems, like ServiceNow, send and receive services from and to Azure, and even establish connections with our chain partners, like EDSN and Mijnaansluiting.nl. Also, SAP PI/PO was used to realize and facilitate the Smart Meter Reading API and Stedin Safety App. Since the migration of the SAP landscape into the cloud, we are using SCP to realize and facilitate the integration and connections between the SAP systems and the other on-premise and cloud solutions.

Last but not least, the Azure platform is used to create and maintain the APIs for the fast-changing systems, like the customer portal and the Volt Station App. This platform is also used to integrate the several fast-changing applications with each other.

#Slow-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#Communication_layer_ESB_components
#bimodal_IT_layers_Communication_layer_CIP
#Communication_layer_CIP_components
#bimodal_IT_layers_Communication_layer_MSA

#Communication_layer_MSA_components
#Fast-Speed_systems_and_software

Interviewer: Thank you for the clarification. Can you maybe elaborate on or explain the components or functions of the SAP PI/PO system, SCP, and Azure platform?

All the middleware solutions are used to send, receive, transform, orchestrate, and map the various services and APIs. These platforms also contain queueing, monitoring, and alert functionalities to notify the functional maintenance team if something happens or goes wrong.

#Communication_layer_ESB_capabilities
#Communication_layer_CIP_capabilities
#Communication_layer_MSA_capabilities

We also use SCP to send and receive messages from EDSN and other chain partners. This is important since our organization does not own all the data, like meter readings and consumption data.

#Communication_layer_CIP_components

The third platform consists of applications that are installed on the Though Books. These applications and systems are used by the engineers within the field. Also, the customer and employee portals are part of this platform. The reason is that this layer is different than the first layer because the applications within this platform need to be fast adjustable, support rapid and frequent deployments, and always be available (24/7).

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#Fast-Speed_characteristics

Presentation of the models

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can recognize the models. As mentioned in question 1 and 2, we are already separating the platform into three layers. So, we have a Slow-Speed layer, Fast-Speed layer, and a Communication layer in place within our IT landscape.

#bimodal_IT_layers_Slow-Speed

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The Slow-Speed layer comprises:

- SAP ECC;
  - Financial Management (Accounting, Controlling, Costing, and Treasury);
  - Sales Administration;
  - Invoice Management;
  - Procurement;
  - SRM;
  - Sourcing;
  - Manufacturing (Production Management, Order Management; and Quality Management);
  - HRM (Payroll, Recruitment, and Staffing);
  - PLM (Material Management, Product Configuration Management, and Quality Management);
- SAP IS-U;
  - Device Management;
  - Work Management;
  - Contract Management for moving in/out;
  - Consumption Management;
  - Billing and Invoice Management;
  - Customer Service;
- SAP C4C (CRM);
  - Account and Contact Management;
  - Interaction Center;
  - Service Management;
  - Sales Management;
  - Case Management;
- SAP Ariba (SCM);
  - Transport Management;
  - Inventory Management;
  - Warehouse Management;
- Work Order System from Oracle;
- GIS;

  - Field Operation Management;
  - Data Management;
  - Asset Management;
- A specific module for interruptions within our network and infrastructure;
- ServiceNow for IT Service Management and Project Management activities;
- And we have Workday (for the HR department) on the roadmap;
- Smart Meter Reading API;
- Stedin Safety App;
- Sensor APIs;
- And other internal APIs and (Web) Apps.

#Slow-Speed_systems_and_software

The Fast-Speed layer comprises:

- Customer portal;
- Employee portal;
- SharePoint;
- Office 365;
- A specific App for the Voltage Stations (Volt Station App);
- Specific Toughbook Apps;
- A Time registration App specifically for engineers;
- Several web Apps;
- Mobile Apps;
  - iOS;
  - Android;
- External APIs and Apps to facilitate and support our digital offerings.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The most common characteristics of the Slow-Speed layer are:

- Stability;
- Secure;
- Robust;
- Reliable;
- A minimum of errors is allowed;
- Supports business continuity, business processes, and business goals;

- Stores and manages business-critical data. However, this criterion does not apply to the Stabilized engagement applications;
- Delivers operational value;
- Use of ITIL processes;
- High Availability and fault tolerance needs to be in place;
- The systems do not change a lot once they are deployed into production;
- Next to agile approaches, also traditional methodologies are involved in the development of a system;
- Systems are not easily replaceable due to their importance;
- Facilitates systems of the Fast-Speed layer through the use of services.

#Slow-Speed_characteristics

The most common characteristics of the Fast-Speed layer are:

- 24/7 Available;
- Flexible and easily adaptable. Ease of making quick changes or replace features with newer versions;
- Rapid delivery oriented. Often and frequent deployments;
- Has the possibility to contain Business Critical Data, but this depends on the App;
- Requirements and planning changes constantly;
- Only an Agile way of development is supported;
- Small and independent increments;
- Delivers business value at each release or iteration and ends at a given moment.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Communication layer comprises a service-based environment. This can be a service bus, like SAP PI/PO, a cloud integration platform, like SCP, or even include a microservice environment, like Azure.

SCP consists of a cloud integration server that acts as a service broker and leverages APIs through an API manager.

The Cloud Integration functionality provides the ability to integrate SAP, non-SAP, cloud, and on-premise applications and processes through messages. API management provides access to simple, scalable, and secure digital assets through APIs and lets internal and external developers consume it. The Open Connectors functionality provides pre-built connectors and adapters to establish seamless connectivity between SAP and non-SAP applications. And the Integration Advisor accelerates the development of business-oriented interfaces, mappings, runtime artifacts and reduces integration efforts.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of the cloud integration platform are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;

            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;

- Alerts;
- Failover;
- Configuration Management;
- Deployment;
- Load Balancing;
- Service Registry;
- Service Discovery;
- Call handling;
- Message Tracking and Throttling;
- Rate Limits;
- Traffic Management;
- Exception Management;
- Analytics;
- API Version Management;
- Monitoring and Logging;
- Policies and Contracts;

- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - OData;
  - XML;
  - FTP and SFTP;
  - Flat File;
  - SMTP;
  - Database Adapter;
  - 3rd Party Adapters;
  - Custom Adapters.

#Communication_layer_CIP_capabilities

The Azure platform comprises the following components:

- Scheduler;
- Controllers;
- API server;
- Resource Manager;
- Container Orchestration engine;
- Storage;
- Queues;
- Logs;
- Container registry;
- Network settings;
- Infrastructure Security.

#Communication_layer_MSA_components

Interviewer: Thank you for the extensive description of the components. However, I am also curious at the components of the ESB. Can you maybe also provide the components of SAP PI/PO to make the list complete?

The (technical) components of SAP PI/PO are:

- Integration Builder;
  - Enterprise Service Repository;
    - Creation and Discovery of Services;
  - Integration Directory;
    - Set-up of Mediation Flows;
- Integration Server;
  - Adapter Engine;
    - Transformations;
  - Integration Engine;
    - Orchestration;
    - Routing;
    - Publish and Subscriptions;
  - Business Process Engine;
    - Rules;
- System Landscape;
  - System Landscape/Tenant Manager;
- Configuration and Monitoring;
  - Configuration Manager;

    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Mediation flow Manager;
    - Event Handler;
    - Adapter Manager;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Logger;
    - Auditor;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;

- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

The management has adopted SAFe as an approach to reach organizational agility. Consequently, the business domains and the teams have adopted the Scrum and BizDevOps approach. However, we are still highly focusing on IT, and the team set-up does not reflect a proper mix between developers, operational resources, and business resources. Therefore, we are still relying on several SDLC approaches, ranging from traditional to agile frameworks to deliver the various IT projects.

#methodology_agile_framework
#methodology_agile_framework_SAFe
#methodology_agile_framework_devops
#methodology_traditional_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The Bimodal environment has the most significant impact on the release strategy. The implementation of a core system is mainly performed via the Waterfall method. However, once the system is live, we need to switch towards an agile approach to align the releases between the Slow-Speed and Fast-Speed environments.

#impact_release_management
#bimodal_IT_layers_Slow-Speed

#methodology_traditional_framework_waterfall
#methodology_agile_framework

9: Do you use different methodologies across the different layers?

Yes. The Slow-Speed domain uses a mix of delivery approaches, ranging from traditional, such as the Waterfall method, to agile approaches, such as Scrum and DevOps. Whereas the systems within the Fast-Speed layer and microservices are mainly delivered through CI/CD or another agile approach, like Scrum.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA_services
#methodology_agile_framework_continuous_integration_and_delivery

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

The best practice to enable CI/CD is through DevOps and using an associated tool, like Azure DevOps. In addition, CI/CD is used to develop and deliver systems and software within the Fast-Speed layer or in the case of microservices.

#continuous_integration_and_delivery_systems_and_software
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

The Fast-Speed domain and the Azure platform.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

We are using Azure DevOps to realize and facilitate CI/CD. Azure DevOps contains the following functionalities:

- Azure Boards. To create, manage and maintain the Backlog, User Stories, Sprints, and Work items;

- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. The Pipeline is integrated with Jenkins. This functionality is used to automate the build (via Maven), test (via Selenium), and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. Azure Kubernetes Service (AKS) is used to enable and execute the deployments;
- Azure monitor. This tool is used for monitoring and analysis.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following configuration practices are applicable for CI/CD:

- A CI/CD strategy needs to be in place;
- Minimization of the branches by keeping the increments small;
- The affected layers must confirm to an agile approach. Preferable DevOps;
- The use of a Version Control System;
- The use of a CI server that can automatically build, test, publish and deploy software;
- Within the Pipeline, the proper tests should be configured and kicked off;
- A verification step needs to be in place before each release moment;
- Configurations and passwords are accessible and easy maintainable;
- Monitoring, analytics, and visualization need to be in place.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

We have the following tools and components in place to develop and deliver the various systems and software:

- A project and ticket management system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Middleware tools;
- Test management tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

To realize and facilitate the Slow-Speed and Fast-Speed systems, we are making use of the following systems and applications:

- ServiceNow for planning, creation of tickets, project management, IT Service Management activities;
- Teams as collaboration tool;
- SharePoint to store all documentation;
- Office 365 for functional specs and requirements;
- We are using the following development kits:
  - SAP NetWeaver and Workbench for ABAP developments;
  - SAP Implementation Guide to configure the SAP system;
  - SDK (SAP Development Kit);
  - JDK (Java Development Kit);
  - Visual Studio (.NET developments);
  - Eclipse for SCP;
- We are using the following middleware tools:
  - SCP;
  - Azure Repos and Azure Pipeline (integration with Jenkins);
- We are using the following Test Management Tool;
  - Panaya Test Dynamix;
- And for the deployments we are using:
  - SAP Solution Manager to deploy SAP transports;
  - Eclipse to create and deploy the IFlows;
  - Azure Release;
  - And the Java Deployment Toolkit to deploy Java software.

Also, we are making use of Postman to test and mock the various services and APIs (this tool is used to test the web services as well as the microservices).

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

Not really. We have already deployed SAP into the cloud. From an organizational point of view, the management has adopted a cloud vision. This means that most of the systems and software are already moved or are moving towards the cloud (in a short period from now). Also, I have to make one remark. Not all systems and software can be moved to the cloud due to their importance. However, we try to keep this at a minimum.

#Slow-Speed_systems_and_software
#deployment_model_cloud
#notes_not_all_systems_can_be_deployed_in_the_cloud

Interviewer: That is interesting. I wonder how are the SAP systems deployed in the cloud referring to a SaaS, PaaS, or an iPaaS environment. Can you maybe explain how this is done?

Yes, SAP ECC is deployed as a PaaS environment, while SAP C4C is subscripted as a SaaS solution. If we look at SCP, then this environment is deployed as an iPaaS environment. In Azure's case, I don't know for sure if this environment is subscripted as an iPaaS, PaaS, or even as a FaaS environment because I can imagine that we only want to pay if we use the environment. From my perspective, we have only the deployment models SaaS, PaaS, and iPaaS in scope. However, you can also verify this with one of the Architects.

#Slow-Speed_systems_and_software
#deployment_model_cloud_SaaS
#deployment_model_cloud_PaaS
#bimodal_IT_layers_Communication_layer_CIP
#deployment_model_cloud_iPaaS
#bimodal_IT_layers_Communication_layer_MSA
#deployment_model_cloud_PaaS
#deployment_model_cloud_iPaaS
#deployment_model_cloud_FaaS

Interviewer: Thank you for the explanation. Can you maybe also elaborate on the technical infrastructure of the mentioned deployment models?

Yes. An on-premise or data center environment uses a Server Farm to realize and facilitate the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can only be reached and executed via a private network established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

For the cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- Or Hybrid.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

We still have some systems in our IT landscape that are deployed on-premise or in a data center. But as already mentioned, the vision of the management is to move towards the cloud. This applies to almost all the systems and software.

#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud
#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

It started with the adoption of Scrum and BizDevOps within the organization. Due to the new CEO and his vision towards the cloud (cloud-only policy), we have already moved the SAP landscape and ESB into the cloud. Currently, we are busy integrating the ESB and API platform. Once this activity is completed, we need to start with the migration of the other core systems and software to the cloud.

#organization_current_IT_landscape
#organization_IT_strategy
#methodology_agile_framework
#deployment_model_movement_towards_cloud

16: What is the future roadmap of the IT landscape of your organization?

The future roadmap is:

- The movement of the core systems towards the cloud or replace the systems by an out-of-the-box SaaS or PaaS solution;

- The adoption of the Digital Operating Model from McKinsey.

#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I believe that Bimodal Architecture will stay relevant in the future. As you already have indicated in your model, there is a need for three separate IT platforms. A Slow-Speed layer that comprises the core systems and stabilized engagement systems. A Fast-Speed layer to facilitate the digital offerings, products, and services. And a Communication layer to establish and ensure the communication and integration between the core systems and fast-changing systems.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy
#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer

However, it is essential to note that the core systems will move more and more towards the cloud as a SaaS, PaaS, or IPaaS solution instead of being hosted in a data center or on-premise. In addition, the maintenance and development processes will change towards an agile approach because the systems will only be available within the cloud.

#methodology_agile_framework
#deployment_model_movement_towards_cloud

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

The respondent did not answer the question due to time.

Candidate 4 – EA-02

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I started my career immediately after graduation from the University at the age of 23. That is almost thirty-five years ago. The first fifteen years of my career, I have been working within the financial sector. After that, I switched towards education and lectured several courses in Business Administration and Information Systems. Also, during this period, I had a lot of contact with the Dutch Ministries.

Consequently, I was asked by one of the Dutch Ministries to help them to transform and adopt a digital platform. From that time onwards, I have fulfilled multiple roles within the IT domain, such as IT strategy advisor, IT project manager, Program Manager, and Enterprise Architect. Four years ago, I joined my current employer within the utility sector {Utility-01}, as an Enterprise Architect {EA-02}.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The IT organization counts approximately 400 members. The IT organization is divided into two capabilities, namely:

- IT;
- OT.

The OT capability is responsible for the continuation and execution of the (core) business processes. This means that the OT is responsible for exploiting and maintaining the core system/Systems of Record, which were primarily hosted in a data-center and on-premise. In contrast, the IT capability is focusing on the digital platform and follows a cloud-first strategy.

I also like to highlight that we have a classic IT organization. The IT organizations' focus is on delivering IT services instead of building up multiple digital business units together with the other organizational units.

From a management perspective, we have already adopted the SAFe methodology. But this is primarily business-driven. This means that the related DevOps teams are highly focused on IT. Also, the team set-up does not reflect a proper mix between the developers, operational resources, and business resources. Therefore, the organization is still looking and experimenting with a digital operation model by setting up BizDevOps teams. But as already said, this is still in an explorative phase.

#IT_capability
#IT_organization_focus

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I am already familiar with the concept of Gartner. Garten differentiates between 2 Modes. Mode 1 is static, and Mode 2 is dynamic. To be frank, I disagree with this concept because the concept misses some crucial insights and components. The Pace-layered Application Strategy from Gartner is a more appropriate approach to use. This approach can be described using a cruise ship metaphor.

A cruise ship has three layers:

- An engine room. The engine room needs to be stable and guarantee operation. If not, it will jeopardize the whole operation and become an obstacle for the continuation. Therefore, this environment needs to be robust and reliable (core systems/Systems of Record);
- A cabin deck. This deck is mainly standardized, but it needs to be configurable/flexible until a certain point (Systems of Differentiation);
- And a recreation deck. This deck is used for engagement, interaction, and customer intimacy. However, this deck can also be split into a stable environment and a fast-changing environment. Take, for example, the swimming pool, which is used for engagement and relaxation. The swimming pool does not change until the end of its lifetime (Stabilized Systems of Innovation and Engagement). Simultaneously, other parts of the deck can be adjusted rapidly based on its surroundings, audience, etc. The same applies to the theater room, which can be used as a movie theater, cabaret theater, conference room, etc. (Explorative Systems of Innovation and Engagement).

#concept_Pace-layerd_Application_strategy
#bimodal_IT_layers_Slow-Speed
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

Within a Bimodal Architecture environment, we can identify three layers. The first layer is related to Mode 1. The second layer is related to Mode 2. And the third layer is related to a Communication layer to establish the integration and connectivity.

Mode 1 comprises:

- Core systems (Systems of Record) to enable and ensure business process standardization, efficiency, and excellence;
- Systems and applications that are extending specific business processes, improving or increasing efficiency (Systems of Differentiation);
- Systems and applications that are adding or delivering operational value (Stabilized Systems of Innovation and Stabilized Systems of Engagement);
- Various integration patterns, including RFC's, proxies, EDI standards, adapters, APIs, and services, are used to connect and integrate the multiple systems and applications.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Mode 2 comprises:

- Digital systems and applications (mobile and intuitive) to increase, improve, and enhance engagement, interaction, brand recognition, commitment, and enable social channels (Systems of Engagement);
- Systems and applications to experiment and explore new technologies, like certain IoTs (Systems of Innovation);
- Various integration patterns, including web services and microservices to connect and integrate the different digital systems and applications.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

The Communication layer comprises:

- A service-oriented environment, such as a service bus, a microservice environment, or even both to establish and promote loose coupling and interoperability via services;
- Middleware systems and tools to create, adjust, monitor, and manage the various services (APIs).

The service-oriented environment(s) and middleware systems enable the integration, connectivity, and interoperability between Mode 1 and Mode 2. These environment(s), systems, and tools are also used internally within Mode 1 and Mode 2 to establish the various integration patterns between the (digital) systems and applications.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_integration_patterns_through_Communication_layer

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can link our IT landscape to the presented models. However, as already discussed in question 1 and 2, the Systems of Innovation and Systems of Engagement needs to be split up! Otherwise, the presented models will partly fulfill their goal/purpose.

Also, the description of the Architecture layers is discussable.

#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_terminology

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The system/applications are divided as follow:

- Mode 1 systems. Systems of Record that have a tight dependency with vendors are mainly closed and mostly involve legacy systems. Examples are:
    - SAP;
    - IBM;
    - Cobolt systems;
    - Oracle;
    - Clicksoftware;
    - Workday;
    - Microsoft Windows;
- Systems of Differentiation. These systems are part of Mode 1 because these systems are developed by the same vendors to help leverage the existing business processes (makes it flexible to a certain point) to support some of the user and customer requirements that cannot meet by the Systems of Record;
    - SAP C4C;
    - SAP Ariba;
    - Microsoft Dynamics 365 CRM;
    - Microsoft Mecoms;
    - Microsoft Power BI;
- Stabilized Systems of Innovation and Stabilized Systems of Engagement. These systems can also be embedded into Mode 1. Note that these systems will never expose functionalities as a service, and they will never become or replace a System of Record;
    - Stedin Safety App;
    - SAP Fiori;

    - Microsoft Outlook App;
    - Chatbots;
    - Report Builders;
    - Ariba Mobile App;
    - Smart Meter Reading API;
    - Sensor APIs;
    - Other internal web and mobile Apps;
- Systems of Integration. Middleware technologies, like Enterprise Service Buses and Microservices Platforms;
    - Biztalk;
    - IBM Websphere;
    - Oracle Service Bus (BEA AquaLogic);
    - SAP PI/PO;
    - SAP Cloud Platform (SCP);
    - Azure;
    - Amazon AWS;
    - Kubernetes;
    - Docker Swarm;
    - API Manager;
- Mode 2 systems. Systems of Innovation and Systems of Engagement;
    - Portals (Sitecore);
    - SharePoint;
    - Power Apps;
    - Specific Toughbook Apps;
    - SAP Customer Experience (CX) App;
    - Web Apps;
    - Mobile Apps;
    - Mendix Apps;
    - And external APIs.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#Communication_layer_systems_and_software
#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of Mode 1 are:

- Stable;
- Robust;
- Reliable;
- High availability is essential to prevent data losses and to ensure business continuity;
- Delivers operational value;
- Traditional delivery of IT. The agile way of working is not always possible;

- Does not change a lot once deployed into production;
- Facilitator for Mode 2;
- Solid test processes;
- Use of ITIL processes;
- Core systems are not easily replaceable due to their importance;
- Problem-solution design.

#Slow-Speed_characteristics

The characteristics of Mode 2 are:

- Makes only use of Agile frameworks;
- Fast-changing;
- Less stable;
- Often and frequent deployments are essential;
- 24/7 Availability. Exception on this criterion is when the associated back-end system is down, and there is no queue or storage in place to capture the data;
- Focus on innovation and exploration;
- The creation of business value is a must. Preferable after each iteration;
- Increments are small and independent;
- Apps can store business-critical data, but that is not a requirement on itself;
- The upcoming trend is only to have a development and production environment;
- Requirements and planning changes continuously.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

We are using SCP to set-up the integration and connectivity with other systems. The reason is that our SAP landscape is running in the cloud.

#bimodal_IT_layers_Communication_layer_CIP

Essential components of SCP are:

- Cloud Integration;
  - Message and Event Broker;
    - Repository for Service Discovery;
    - Integration Builder;

- Transaction Handler;
- Configuration Manager;
- Security Manager;
  - Message Security;
  - Encryption;
  - Authorization;
- Event Handler;
- Enterprise Service Repository;
- Queues;
- Exception handler;
- Logger;

- Connectivity Hub;
  - Repository for Adapter Discovery;
  - Adapter Builder;
  - Adapter Configuration Manager;
  - Adapter Security Manager;
    - Authentication;
    - Access and Identity Manager;
- Integration Engine;
  - Adapter Engine;
  - Orchestration Engine;
  - Transformation Engine;
  - Routing Engine;
  - Rule Engine;
  - Publisher and Subscriber Engine;

- API Management;
  - Repository for API Discovery;
  - API Designer;
  - API Test Manager;
- BPM;
  - Business Process Manager;
  - Workflow Builder;
  - Business Activity Monitor;
- Gateway.

#Communication_layer_CIP_components

Essential components of the microservice environment, in our case Azure, are:

- Nodes and Pods;
- Resource Manager;

- Container Orchestration engine;
- Scheduler;
- Controllers;
- Key-value Database;
- API server;
- Storage;
- Queues;
- Logs;
- Container registry;
- Networking;
- Provisioning;
- Infrastructure Security.

#Communication_layer_MSA_components

Interviewer: Thank you for indicating the components. Can you maybe also provide the components of SAP PI/PO to make the list complete?

The components of SAP PI/PO are:

- Integration Builder;
    - Enterprise Service Repository;
    - Integration Flow Manager;
- Integration Server;
    - Integration Engine;
        - Orchestration;
        - Transformations;
        - Dynamic Routing;
        - Publish and Subscriptions;
    - Business Process Engine;
        - Rules;
- System Landscape;
    - System Landscape/Tenant Manager;
- Configuration and Monitoring;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Adapter Manager;
    - Access controller;
        - Authorization;

  - Exception handler;
  - Monitoring;
  - Reporting;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflows;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

Before I answer this question, I like to indicate that some trajectories do not discriminate between the different layers and their associated SDLC methodologies. Let's look at tendering, migrations (also to the cloud), or governmental changes. You will notice that these trajectories are bounded to certain rules, incorporates a certain level of complexity, can be time-consuming, or contain specific deadlines. For these trajectories, it is advisable to use a project-based or a traditional approach instead of an agile approach.

#methodology_traditional_framework

Interviewer: This is an exciting point. Does this mean that using an agile approach is prohibited in these trajectories or even jeopardizes the project? I mean, in practice, the opposite is happening. More and more organizations are moving from a traditional approach towards an agile one.

Yes, that is true. And of course, some of the activities within these trajectories can be done in an agile way. But, keep in mind that it is impossible to execute these trajectories only in an agile way because these trajectories are bounded to some characteristics that are not supported or even jeopardize the principles of agile.

Coming back on the question. The Systems of Record are mainly delivered through the Waterfall methodology. Once the systems are deployed into production, agile methodologies are used to maintain and utilize the systems. In contrast, the Systems of Differentiation and the Stabilized Systems of Innovation and Engagement are mainly delivered using an agile method, such as Scrum, XP, or Kanban.

#Slow-Speed_systems_and_software
#methodology_traditional_framework_waterfall
#methodology_agile_framework
#Slow-Speed_stabilized_systems_of_innovation
#methodology_agile_framework

For the Mode 2 environment applies that the majority of the increments are delivered through CI/CD. If CI/CD is not possible, another agile method is used to deliver the increment.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The first impact is related to release management. Because each Mode has its own development cycle, we need to align when to release a specific functionality that is impacting both modes. If this is not done thoughtfully, the user and functional requirements could be compromised and deliver us many integration and connectivity issues.

#impact_release_management
#impact_continuous_alignment

The second impact is related to prioritization. When an API needs an adjustment or enhancement, all the related IT systems need to be identified and investigated. After this, the right priority needs to be determined and set for each development. This goes mostly wrong in practice because the priority is set by a PO who cannot oversee all adjustments and developments.

#impact_prioritization

9: Do you use different methodologies across the different layers?

The following phases are endured in case of an implementation and exploitation of a core system in Mode 1:

- Phase 1: The core system is implemented via a project (PRINCE2) or through a traditional approach, such as Waterfall;
- Phase 2: New functionalities/features are implemented or enhanced via an agile approach to ensure the release moments' speed and alignment.

For the other systems and applications in Mode 1 apply that they are developed and delivered through an agile approach. Note that there is no CI/CD pipeline in place.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#notes_no_CI/CD_pipeline_in_place

Mode 2 systems are mainly implemented through CI/CD or use another agile framework, like Scrum.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

The best practice to enable CI/CD is through the DevOps approach. In our case, we use DevOps as well as the Scrum approach to facilitate CI/CD.

In our IT landscape, CI/CD is applied in Mode 2 and within the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

Mode 2 and the Communication layer because we are making us of a Microservice platform. Also, note that the SOAP messages (web services) are delivered through another agile approach, such as Scrum.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption
#bimodal_IT_layers_Communication_layer_ESB_services
#methodology_agile_framework

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The development of increments starts locally by a developer. The developer can choose several Integrated Development Environments (IDEs) or Development Kit (DKs) to execute the coding. Once the code is realized, the source code is submitted into a Repository. After this, Azure DevOps comes into play for building and deploying the software packages (increments).

#software_delivery_life_cycle_components
#continuous_integration_and_delivery_systems_and_software

Azure DevOps provides the following functionalities:

- Azure Boards. To create, manage and maintain the Backlog, User Stories, Sprints, and Work items;

- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. This functionality has an integration with Jenkins. This functionality is used to automate the build (via Maven), test (via Selenium), and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. We are using the services of Azure Kubernetes Service (AKS) to enable and perform the deployments;
- Azure monitor. This tool is used for monitoring and analysis.

Subsequently, we use Postman to test and mock the various APIs.

#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following conditions (configuration practices) are applicable in the case of CI/CD:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- Configurations and passwords are easily accessible and up-to-date;
- A verification step should be in place before each deployment;
- Use of a Version Control System needs to be encouraged;
- Use of an Automation Server that can automatically build, test, publish and deploy the various increments;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

Various tools and components are used to develop and deliver the systems, applications, APIs, and services. These components and tools are:

- A tool to manage the planning and requirements;
- A project and IT Service Management system;
- A ticketing system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;

- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

To realize and facilitate the Mode 1 systems, we are making use of the following tools:

- ServiceNow for the creation of tickets;
- For functional and technical specifications:
    - Office 365;
- For collaboration, we are using Teams;
- SharePoint to store all documentation;
- We have the following development kits in place:
    - SAP NetWeaver and Workbench for ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Eclipse for SCP, .NET and C/C++ developments;
    - iOS development is done through Xcode 12 and Swift UI;
    - Android development is done through Android Studio;
- We are using the following Test Management Tool;
    - Panaya Test Dynamix;
- And for the deployments, we are using the following tools:
    - SAP Solution Manager to deploy SAP transports;
    - Eclipse is used to create and deploy the IFlows;
    - Azure Release;
    - And the Java Deployment Toolkit to deploy Java software.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

No, there is no formal relationship. The Mode 2 systems are mainly deployed into the cloud.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

We are using a mixed deployment strategy for the Stabilized Systems of Innovation and Stabilized Systems of Engagement. Some of these systems are deployed on-premise, such as the Microsoft Outlook App. In contrast, other systems are hosted in the cloud. For the Systems of Differentiation, we prefer SaaS solutions. Finally, the Systems of Record are deployed as much as possible in the cloud.

#Slow-Speed_stabilized_systems_of_innovation
#deployment_model_on-premise
#deployment_model_cloud
#bimodal_IT_layers_Slow-Speed
#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud
#deployment_model_cloud_SaaS

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

Currently, there are no common deployment practices available for each layer. However, we are entering into an era in which all systems are moving towards the cloud.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

The current IT landscape is established due to the fusion of multiple utility companies. The fusion itself is a result of the split between the grid and the suppliers.

#organization_IT_milestones
#organization_current_IT_landscape

16: What is the future roadmap of the IT landscape of your organization?

The vision of the upcoming 3 - 5 years is to adopt a Cloud and API first strategy. As mentioned in question 1, the organization has already adopted SAFe. The management wants to incorporate agility on each level within the company.

#organization_IT_strategy
#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

The original concept is nice, but it will not work in practice. The Pace-layered methodology is more suitable. See for the clarification also the answer provided in question 1.

#concept_bimodal_IT_future
#concept_Pace-layerd_Application_strategy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The IT organization is already working in Scrum/BizDevOps teams (each team has 7-8 members). From that perspective, there is no impact on the Bimodal Architecture environment. However, the teams are still highly focused on IT, and the SAFe methodology is primarily business-driven. This needs to change to make full use of the teams' knowledge and capacity.

#methodology_agile_framework_Scrum
#methodology_agile_framework_SAFe
#concept_bimodal_IT_agile_teams

Also, a sub-architecture or another viewpoint is needed to facilitate the iterative way of working.

#concept_bimodal_IT_agile_architecture

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

In the case of priorities, a PO is in the lead to determine these. Also, the costs of the teams are fixed and based on time and material. So, there is no impact on the Bimodal Architecture environment.

#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

Mode 2 systems and applications can have some technical debt. The reason is that for each offering or service, an App is built to invoke one or more APIs. However, if there are any issues or problems, it can be fixed very fast. Mostly, within another sprint.

For the Mode 1 systems and applications apply that they can have a high technical debt due to the systems' complexity and robustness. This is mostly the case for core systems/Systems of Record.

#concept_bimodal_IT_technical_debt

Candidate 5 – SA-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I have eleven years of working experience. I am working as a Solution Architect {SA-01} at a global bank {Banking-01}. My focus area as a Solution Architect spans from web development to microservices and APIs in Azure (.NET, C-Sharp, and Visual Basic).

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

I am working for the business unit ABF (Asset Based Financing). This unit counts 850 people and is active in the whole of Europe. The IT domain is responsible for all the agglomerated solutions that are used to support and enable corporate financing, operational asset leasing, stock and receivables financing, and credit management.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal Architecture is an architectural style that consists of a coherent IT platform with three layers. One layer aims for the execution of excellence. The associated systems are mainly standardized, support the business processes, and ensures operational performance. The layer also includes modern systems that deliver operational value. This layer is the Operational Backbone of the organization and helps the company to digitize. Another layer aims at the digital offerings. Through this layer, the organization tries to increase and enhance the user and customer experiences. The associated systems are generally built and modified through CI/CD to support the user and customer needs. Mostly, it involves new technologies and banking products to enable diversity within the product portfolio and offerings towards the customers. This layer realizes the Digital Platform and supports the digitalization strategy of the company. The third and last layer is used to connect the Operational Backbone with the Digital Platform through services. The layer consists of a service-based environment in which the Operational Backbone's functionalities are exposed as services. Subsequently, these services are used to ingrate the different systems internally. This also applies to the modern systems that generate operational value and realize and facilitate the Digital Platform. This layer can be referred to as the "Service Layer."

#bimodal_IT_layers_Slow-Speed
#concept_digital_transformation_operational_backbone
#bimodal_IT_layers_Fast-Speed
#concept_digital_transformation_digital_platform
#methodology_agile_framework_continuous_integration_and_delivery
#bimodal_IT_layers_Communication_layer
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

As mentioned in question 1, Bimodal Architecture is an IT platform that exists of three layers:

- The Operational Backbone consists of systems that can be marked as "stable" and "predictable." This layer includes systems, such as SAP, Dynamics 365, Avaloq Core, and COBOL systems.
- The Digital Platform consists of systems that can be marked as "explorative" and "digital." This layer includes systems, such as Sitecore, Power Apps, SharePoint, E-commerce, Office 365, and the Mobile Banking App.
- The Service layer is a service-based environment, such as Azure or Biztalk. This layer is essential because it connects, integrates, and executes the communication between and within the Operational Backbone and the Digital Platform through services. Without this layer, it is impossible to realize the digital platform or establish the connection between the stable systems and the modern systems that deliver operational value.

#bimodal_IT_layers_Slow-Speed
#concept_digital_transformation_operational_backbone
#Slow-Speed_systems_and_software
#bimodal_IT_layers_Fast-Speed
#concept_digital_transformation_digital_platform
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

The presented models look familiar. The layers and their connections are showing similarities with the current set-up of the IT landscape. The functionalities of Dynamics 365 are exposed as JSON messages (REST protocol). In contrast, the SAP functionalities are exposed via proxies and SOAP messages, which are eventually converted into REST messages (JSON) within the Service Layer.

#Slow-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

I also believe that the various SAP systems communicate internally via RFCs, EDIs, and adapters, but you can verify this with my colleague.

#bimodal_IT_integration_patterns_Slow-Speed

The exposed services of the Operational Backbone layer are sent to the service-based environment, which includes an ESB and a microservice environment, where the messages are validated, transformed, enriched, and routed towards their destinations, such as Sitecore or the Banking App. Simultaneously, the service-based environment also allows messages to be sent between systems within a particular layer, such as SAP and Dynamics 365 CRM. This means that the service-based environment does not only connect the Operational Backbone environment with the Digital Platform, but it also handles the (internal) communication between the systems within each layer.

#bimodal_IT_layers_Communication_layer
#Communication_layer_ESB_capabilities
#Communication_layer_CIP_capabilities
#Communication_layer_MSA_capabilities
#bimodal_IT_integration_patterns_through_Communication_layer

Most of the systems in the Digital Platform are built in .NET or Java. The services are providing the objects and data to support the applications and features within this environment.

#Fast-Speed_systems_and_software

I notice only one difference between my description of a Bimodal Architecture and your models. As already indicated, not all modern systems are part of the Digital Platform. Let's take the payment requests system Tikkie as an example since this App is relatively new. When we started with the development of this App and its features, it was part of the Digital Platform. However, when the App reached its purpose, namely making payment requests fast and easy for our customers, the App and its corresponding APIs stabilized. We also noticed that the App started generating operational value through a simplified payment process. This was the moment to make the App and the related APIs part of the Operational Backbone instead of the Digital Platform. This is a scenario that happens often. However, the problem is that the particular App cannot be embedded into the Operation Backbone layer because your models do not support this scenario.

#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed
#Slow-Speed_stabilized_systems_of_innovation

Interviewer: Can you maybe explain operational value and why some modern systems are part of the "Operational Backbone" layer instead of the "Digital Platform"?

The goal of operational value is to improve and optimize a company's operations by simplifying, automizing, and measuring the business processes. IT innovations and smart solutions, such as machine learning, augmented reality, or AI are perfect examples to achieve this. By generating and sending back the metrics and other useful data, we can improve our products and services and make the organization lean. Therefore, these particular systems should be stable, contain no or minimal errors, and should not be changed rapidly or often. This contradicts the Digital Platform characteristics, in which the systems are subjected to a fast-changing environment and market.

#Slow-Speed_stabilized_systems_of_innovation

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The systems and software that typically can be found within the Operational Backbone environment are:

- Dynamics 365 Business Central;
- Dynamics 365 Finance & Operations;
- Dynamics 365 Customer Service;
- Dynamics 365 Field Service;
- Dynamics 365 Human Resources;
- Dynamics 365 Supply Chain Management;
- Dynamics 365 CRM;
- Power Platform;
- Aquarius;
- Power BI;
- SAP S/4HANA;
- Avaloq Core;

- COBOL solutions and systems;
- Foreign Exchange (FX) Trade API;
- Tikkie API and associated App;
- Other internal APIs.

The following technologies are also part of the Operational Backbone environment:

- Machine Learning;
- Augmented Reality;
- And AI.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Systems that typically can be found within the Digital Platform layer are:

- Power Apps;
- Sitecore;
- SharePoint;
- E-commerce;
- Office 365;
- Teams;
- Web Apps;
- External APIs;
- iOS Apps and Android Apps;
    - Mobile Banking App;
    - Agent App;
    - Grip App;
    - Retirement App.

Also, note that most of the systems, applications, and solutions are developed in .NET, C-Sharp, or Java.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of the Operational Backbone layer are:

- Stable;
- Reliable;
- Robust;
- Security is an essential and ongoing aspect;
- Standardization is an important aspect;
- Supports business processes and related business-critical objects and data;

- Does not change a lot once deployed into production;
- Less error-prone;
- Delivers operational value;
- Next to agile approaches, also traditional methodologies can be involved in the development of a solution/system;
- Solid test processes;
- Problem-solution design.

#Slow-Speed_characteristics

The characteristics of the Digital Platform are:

- Delivers business value at each release or iteration;
- Flexible. The solution can change rapidly and often;
- Stores and contains minimal or no critical data;
- Highly error-prone due to the experimental and explorative nature;
- Only an Agile way of development is supported;
- Rapid delivery oriented. Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market;
- The solutions are mainly centered around user and customer experience;
- New technology-oriented;
- Trend to have only a development and production environment.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

Azure has released a single platform for the integration of on-premise and cloud-based solutions. This platform, called Azure Integration Services, combines the API Management, Logic Apps, Service Bus, and Event Grid functionalities.

The Service Bus is used to connect the on-premise and cloud-based systems and services through highly secured messages. The Event Grid ensures the connection between Azure and third-party services through a fully managed event routing service, which is based on a publish-subscribe model. Logic Apps is used to automate various workflows. And the API Manager is used to publish APIs. These APIs can be used by internal and external developers when connecting to back-end systems and services.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of the cloud integration platform are:

- Cloud Integration Service;
  - Message and Event Hub;
    - Repository for Service Discovery and pre-build Mediation Flows;
    - Integration/Message Builder;
      - Set-up of Mediation Flows;
      - Editing of Flows;
    - System Landscape/Tenant Manager;
    - Transaction Handler;
    - Configuration Manager;
    - Security Manager;
      - Set-up of Message Security;
      - Set-up of Encryption;
    - Access Controller;
      - Set-up of Authorization;
    - Event Handler;
      - Set-up of Events (Event Producer and Event Listener);
    - Message Queues;
    - Exception handler;
    - Logger;
    - Auditor;
  - Connectivity Hub;
    - Repository for Adapter Discovery;
    - Adapter Builder;
      - Set-up of Adapters;
    - Adapter Configuration Manager;
    - Adapter Security Manager;
      - Set-up of Authentication;
    - Adapter Event Handler;
    - Workflow Builder;
      - Set-up of Workflows;
      - Editing of Workflows;
    - Access Controller;
      - Identity Access Manager;
      - Set-up of Authorization;
    - Logger;
    - Auditor;
  - Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;

                - Routing Engine;
                - Rule Engine;
                - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The cloud integration platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;

- Traffic Management;
- Exception Management;
- Analytics;
- API Version Management;
- Monitoring and Logging;
- Policies and Contracts;

- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

Currently, a lot of organizations are not using the cloud integration platform yet. They are using a service broker, such as Biztalk and WCF. In that case, the service bus (Biztalk) comprises the following components:

- Message Builder;
    - Set-up of Adapters;
    - Creation of Services;
    - Set-up of Mediation Flows;

- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Event Handler;
    - Mediation flow Manager;
    - System Landscape/Tenant Manager;
    - Auditor;
    - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;

    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

The best practice to run microservices is through the use of container technology. These containers can be managed through a container orchestration platform. The container orchestration platform is deployed as a cluster and consists of a set of nodes. These nodes are hosting pods, which represent a set of running containers.

The container orchestration platform comprises the following (technical) components:

- Scheduler;
- Controllers;
- Key-value Database;
- API server;
- Several Nodes;

Universiteit Leiden

    - Hosts several Pods;
- Resource Manager;
- Container Orchestration engine.

Other components in the infrastructure are:

- Storage;
- Queues;
- Container registry;
- Container repository (optional);
- Network;
- Provisioning;
- Logs;
- Infrastructure Security.

#Communication_layer_MSA_components

The microservice environment realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Failover;
    - Microservice Security;

- Configuration Management;
    - Networking/IP;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Container;
    - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

We are using a mix of SDLC methodologies within our IT domain. These methodologies are ranging from traditional to CI/CD. It includes:

- PRINCE2. To manage and deliver big IT projects;
- Waterfall. For the delivery of business-critical and core systems;
- Spiral Model. In some cases, we need prototyping;
- Kanban. We use this approach to realize some changes;
- Extreme Programming (XP);
- Scrum;
- BizDevOps;
- CI/CD.

#methodology_traditional_framework
#methodology_agile_framework

Interviewer: This is a quite surprising answer. Can you maybe explain more in detail why the IT domain is using various approaches?

The cause for using different methodologies is due to the diverse systems and software that we have within our IT landscape. On one side, we are using new development technologies, such as Xcode 12, Swift UI, Android Studio, and Visual Studio, to build modern systems and software. These solutions can be picked-up through an agile approach. However, we also have to deal with the "old" COBOL solutions and systems because these systems are still part of our IT landscape. To enhance or integrate a COBOL solution, we first need to perform an extensive and detailed impact analysis and comply with all the security standards. Next to these COBOL solutions, we also have several products from Microsoft, SAP, and Oracle adopted into our IT landscape. All these systems and software have their own proven SDLC methodologies.

#Fast-Speed_systems_and_software
#methodology_agile_framework
#Slow-Speed_systems_and_software
#methodology_traditional_framework
#methodology_agile_framework

Do not forget that we are a global bank. Therefore, we need to comply with certain laws, regulations, security standards, audit standards, and IT standards. This means that an assessment needs to occur for each IT project to determine which SDLC approach is the most appropriate and convenient to develop and deliver the related system.

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The impact is related to release management. Due to the three layers, the various teams need to be continuously aligned and up-to-date. Through the continuous alignment, the teams will be able to manage the different IT projects, oversee the multiple deployments, and reduce interoperability issues. Besides, this alignment will help to keep the several teams informed.

#impact_release_management
#impact_continuous_alignment

9: Do you use different methodologies across the different layers?

The systems of the Digital Platform are generally delivered through CI/CD. However, a small amount of the systems is developed and delivered using another agile method, such as Scrum or Extreme Programming (XP).

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

As already indicated in question 7, the Operational Backbone layer systems and software are developed and delivered through a mix of SDLC approaches. The Waterfall method is one of the most famous SDLC methodologies for developing and delivering a system within this layer. However, the systems and software can also be developed and delivered through the use of PRINCE2, Spiral Model, Kanban, XP, and Scrum.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework_waterfall
#methodology_traditional_framework
#methodology_agile_framework

I also like to mention that once a system has gone live within the Operational Backbone layer, the new features of that particular system are mainly developed and delivered using an agile approach, such as Kanban, Scrum, or BizDevOps.

#Slow-Speed_systems_and_software
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

To enable CI/CD, we have adopted Azure DevOps from Microsoft. CI/CD is used within the Digital Platform and the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption
#continuous_integration_and_delivery_systems_and_software

Through CI/CD, we can ensure early and often deployments. Therefore, CI/CD lends itself as an ideal candidate to realize and utilize the systems and solutions within the Digital Platform layer. And also for the development of microservices.

#Fast-Speed_characteristics
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery

10b: Which layers are affected by CI/CD?

The affected layers are:

- Digital Platform;
- And the Service Layer. Particularly, the microservices environment, such as Azure.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The components of the Azure DevOps tool are:

- Azure Boards. To create, manage and maintain the Backlog, User Stories, Sprints, and Work items;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. This functionality has an integration with Jenkins. This functionality is used to automate the build (via Maven, Gradle, Visual Studio, etc.), test (via Selenium, FitNesse, JUnit, etc.), and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. Azure DevOps can use Azure Functions, Azure Service Fabric, Azure Kubernetes Service, or Docker containers as a deployment technology;

- Azure monitor (Application Insights and Azure Alerts). This tool is used for monitoring and analysis.

To test and mock the APIs, Postman is used. Postman provides the following functionality:

- Newman. This functionality is used to automate and test the various API calls.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following configuration practices are necessary to facilitate CI/CD:

- It is crucial to have a CI/CD strategy in place;
- A recovery strategy should be in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Only required features and functionalities should be build and released;
- A verification step should be in place before a deployment;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

The following tools and components are applicable when we develop and deliver the systems and software within the different layers:

- A tool or system to manage the planning, requirements, IT projects, IT Service Management activities, and tickets;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

As already mentioned in question 7 and 12, we have various systems and software within our IT landscape. Therefore, we are using multiple tools to develop and deliver the systems within the different layers. Since my main focus is on Microsoft products and associated web and mobile developments, I can and will only provide you the tools related to the development of Microsoft solutions.

From my point of view, we are using the following tools to realize and facilitate the Bimodal IT environment:

- Functional and technical design tools:
    - Office 365;
- Collaboration tools:
    - Teams;
- Document Management System:
    - SharePoint;
- Version Control systems/Repositories:
    - Azure Repos;
- Development tools:
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse (.NET and C/C++ developments);
    - Azure Pipeline and Azure Release;
    - Azure Logic Apps;
- Test tools:
    - Azure Test Plans (part of Azure DevOps to perform manual and load tests)
- Deployment tools:
    - App Service;
    - Azure Functions (serverless computing);
    - Azure Service Fabric;
    - Azure Kubernetes Service;
    - Container Instances, such as Docker;
- Monitoring and Analytics tools:
    - Azure monitor;
    - Application Insights;
    - Azure Alerts.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

No. There is no observable relationship between the deployment models and the three layers within a Bimodal Architecture environment. The systems of the Digital Platform are generally hosted in the cloud. For the systems of the Operational Backbone layer applies that they can be hosted anywhere. It depends on a system which deployment model is chosen. In the case of a critical information system, only the deployment models on-premise, data center, or a private cloud environment are eligible. The reason is that an organization wants to have control over these systems since the systems are supporting core business processes and store critical data. However, since the last couple of years, there is a trend noticeable towards the cloud.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud
#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#deployment_model_cloud
#deployment_model_cloud_Private
#deployment_model_movement_towards_cloud

Interviewer: Thank you for the answer. Can you also elaborate a little bit on the technical infrastructure of the mentioned deployment models?

Yes. In the cloud environment, the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology, like Azure Functions or Azure Storage Account, and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- Hybrid.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

In an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can be reached and executed via a private network established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

A movement towards the cloud is noticeable. However, some systems and software will never move towards the cloud due to their importance.

#deployment_model_movement_towards_cloud
#notes_not_all_systems_can_be_deployed_in_the_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

The bank crisis in 2008 changed the way how banks are conducting business and support their customers. After the credit crisis, we needed to have flexible systems to compete with other parties. On the other hand, we needed systems that could alert us early when problems occurred in the stock and financial markets. Also, we had to deal with more regulations to cope with bankruptcy and ensure that we could pass the various bank stress tests. These changes required a new IT landscape that could support the goals and help the bank become competitive and healthy again.

#organization_IT_milestones
#organization_IT_strategy
#organization_current_IT_landscape

16: What is the future roadmap of the IT landscape of your organization?

At this moment, all IT projects are frozen due to covid 19. Our first focus and priority are to support our customers, clients, and employees by enabling remote services. Once the environment stabilizes, we can rethink our IT strategy to help our customers, clients, and employees even better with smart solutions and banking products.

#organization_IT_strategy
#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Bimodal Architecture will stay a relevant concept within the future. The underlying reason is that we need to cope with two types of systems. One type of system will help us to support and optimize the business processes and automate the operations. While the other type of system will enhance the customer interactions and experience even further.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

The respondent did not answer the question due to time.

Universiteit Leiden

Candidate 6 – INL-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I started my career immediately after my graduation in 2006 at {Utility-02}. My first role was at the call center to get some working experience. After a year, I moved towards the Functional Maintenance (FM) team, where I fulfilled the role of a Functional Consultant with a main focus on CRM (SAP CRM, Xaris and Genesys), IS-U (SAP IS-U, Mijnaansluiting.nl, SAP FICO), and associated integration patterns. I executed this role for seven years. Subsequently, I became the manager of the FM group and fulfilled this role for another four years. The last two years, I am one of the three Innovation Leads {INL-01} within the organization. Through this role, I help the organization to adopt and implement the different technologies and innovations within the IT landscape. In which SAP is the biggest component.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The organization operates within the Utility industry {Utility-02) and is responsible for the water supply within some regions of The Netherlands. The IT capability has around 80 FTE's. The IT resources are divided internally (around 55%) and externally (around 45%). All the IT resources are part of one or more Scrum teams, depending on their role and skills.

#company_profile
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal Architecture comprises two separate but highly integrated IT environments. One environment is stable and is the single source of truth. This environment mainly consists of Systems of Record and Systems of Differentiation, like SAP ECC, SAP IS-U, SAP CRM, Genesys, Salesforce, and GIS. Besides, this environment also comprises systems and software related to innovation, like sensors and the Digiduinkaart App. In contrast, the second environment is built around the fast delivery of increments and serves the fast-changing markets and environments. The main focus of this environment is on exploration and experimentation.

#concept_Pace-layerd_Application_strategy_terminology
#bimodal_IT_layers_Slow-Speed

Universiteit Leiden

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics
#bimodal_IT_layers_Communication_layer

Interviewer: That's interesting. Why do you split the Systems of Innovation? Should not all the Systems of Innovation be part of the second (Fast-Speed) environment?

No, I disagree with this assumption. There is a clear split between the systems. Let's take IoT, such as sensor technology, as an example. Sensors are relatively new within our sector and even within the industry. However, the related APIs, systems, and applications of these sensors, for measuring water volumes and water quality within the pipelines are very standardized and stable. It delivers operational value by indicating leakage or pollution within our water infrastructure and guarantees the water distribution. Therefore, this IoT technology and associated components create value for the whole organization instead of for a certain user or customer. If we look at another IoT initiative, like the HoloLens, we can determine that the related environment is constantly changing. Besides, the technology is also only relevant to a certain group of users and will not benefit the whole organization.

#Slow-Speed_stabilized_systems_of_innovation
#Fast-Speed_systems_and_software

Also, note that when the development of a system or application within the Explorative IT layer stabilizes, or the digital App becomes mature and generates operational value, the particular system becomes part of the Stable IT layer.

#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

From my perspective, three components are fundamental and forms the building blocks of a Bimodal Architecture environment. The first component is related to a Stable IT layer. The second component is related to an Explorative IT layer. And the third, and last component, is related to an Integration environment.

The Stable IT layer comprises:

- Systems and software to standardize and optimize business processes, increase efficiency, and improve and enhance operational excellence;

- Solutions, systems, and "digital" applications to extent specific business processes, enhancement of internal digitalization, and for creating and delivering operational value;
- Multiple integration patterns to connect and integrate the systems and applications within and outside the layer by exposing the functionalities as services. The integration patterns include RFCs, EDIs, adapters, proxies, Enterprise Services, APIs, web services, and microservices.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

The Explorative IT layer comprises:

- Web and mobile solutions to improve the experience, interaction, and engagement.
- Systems and applications to experiment and explore new technologies;
- Multiple integration patterns. These patterns are used to connect the digital systems and applications with each other through services.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_integration_patterns_Fast-Speed

The Integration layer comprises:

- A service-oriented environment, such as an ESB, a cloud integration platform, or a microservice environment. It can occur that an organization has a combination of service-based environments to establish and leverage the integration patterns. These environments establish and promotes loose coupling, interoperability, and modularity;
- Middleware systems and tools to create, adjust, monitor, and manage the services (web services and microservices) and APIs.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer _ESB
#bimodal_IT_layers_Communication_layer _MSA
#bimodal_IT_layers_Communication_layer _CIP

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, the models are very recognizable. The presented models cope with the view of how I see Bimodal Architecture.

The only comment that I have is, is that I define the layers slightly differently. Slow-Speed and Fast-Speed can be a bit misleading for an organization.

#bimodal_IT_layers_terminology

Interview: Can you explain why the description of the layers (within the presented models) are misleading?

Yes, I can. The Slow-Speed domain refers to systems that rarely change. This is not true. Let's take SAP ECC as an example. SAP ECC is known as a slow, rigid and inflexible system. However, SAP ECC also contains components that support digitalization and user engagement, like Fiori. Besides, SAP ECC also allows its functionalities to be exposed as services via proxies and Enterprise Services to facilitate the applications within the Explorative IT layer.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed

Another example is related to the Fast-Speed layer. For example, if we look at Salesforce, this system can be marked as a fast-changing customer-facing system. However, the majority of adjustments are performed during the implementation of the system. After the systems' go-live, we do not want to change the system a lot because it contains, stores, and manages business-critical objects and data.

#bimodal_IT_layers_Fast-Speed
#Slow-Speed_systems_and_software

Interviewer: I understand your point, but from your perspective, in which environment/layer would you allocate the mentioned systems then?

In my case, I would say that SAP, Salesforce, and the sensors are part of the Stable IT layer instead of the Slow-Speed layer. And when we look at IoT initiatives, like the HoloLens, or portals, we could say that these systems are part of the Explorative IT layer.

#bimodal_IT_layers_terminology
#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

Interviewer: I agree with your statement. However, the description of the layers is retrieved from the theory, and that's why I used it within my models. But, your description of the layers is exactly how I distinguish the two layers. So, one layer focuses on the business continuity and is stable, while the other layer is explorative and supports innovation, interaction and commitment.

Yes, that is exactly what I mean!

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

Typical systems of the Stable IT layer are:

- SAP CRM;
- SAP ECC;
- SAP IS-U;
- Genesys;
- Salesforce;
- GIS;
- Xaris;
- SAP Fiori;
- Microsoft Outlook;
- Industry-specific Apps;
    - Digiduinkaart App;
- IoT;
    - Sensor APIs and related systems and applications;
- Internal APIs and related Apps.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Typical systems of the Explorative IT layer are:

- Office 365;
- SharePoint;
- Teams (Collaboration App);
- Power Apps;
- Portals (Sitecore);
    - Customer Portal;
    - Employee Portal;

- Partner Portal;
  - Supplier Portal;
- Industry-specific Apps;
  - SAP Work Manager on iOS;
- GeoWEB;
- IoT;
  - HoloLens;
- 3D printing;
- Digital Twins;
  - Creation of 3D models of a physical environment;
- Web Apps;
- Mobile Apps (iOS and Android);
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of the Stable IT layer are:

- Stable;
- Reliable;
- Secure;
- Low in errors;
- Focus on standardization;
- Delivers operational value;
- The systems and software are supporting the business processes, objects, data and ensures business continuity;
- The systems and software do not change a lot once they are deployed into production;
- Traditional delivery of IT. The agile way of working is not always possible;
- Implementation costs are high;
- The systems and software are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The characteristics of the Explorative IT layer are:

- Features are small and independent;
- 24/7 Available. Off-line capability is required to guarantee this;
- Delivers business value at each release or iteration and ends at a given moment;
- Flexible. The systems and applications are easily replaceable;
- Only an Agile way of development is supported;
- Focus on innovation, experimentation, and exploration;
- Less focus on Risk Management. Errors are already factored in;
- Rapid delivery oriented. Often and frequent deployments.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

There are three middleware systems available to set-up the connectivity and integration within the IT landscape. We are using Biztalk as an ESB to establish the connectivity between the on-premise and data center systems and software. After moving the SAP landscape to the cloud, we also got SAP Cloud Platform (SCP) as a middleware. This integration platform is used to realize the integration between SAP and non-SAP systems. And on top of that, we have an Azure platform available within our IT landscape to realize and facilitate microservices.

#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA

As already indicated, we are using Biztalk and WCF adapters to realize the connectivity with the on-premise and data center systems and software.

#bimodal_IT_layers_Communication_layer_ESB

The components of this middleware are:

- Message Builder;
    - Set-up of Adapters;
    - Creation of Services;
    - Set-up of Mediation Flows;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;

    - Message Security;
    - Encryption;
  - Access Controller;
    - Authorization;
  - Exception handler;
  - Event Handler;
  - Mediation flow Manager;
  - System Landscape/Tenant Manager;
  - Auditor;
  - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflow;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Message Tracking and Throttling;
  - Exception Management;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;

    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

Azure is used as an IaaS environment to host SAP and to help the organization to explore and innovate. The Azure environment is also used as a microservice environment to connect the external systems, to orchestrate containers to leverage multiple functionalities and Apps, enables Continuous Integration (CI) and Deployment (CD), and last but not least, support innovation initiatives, like 3D modeling.

#bimodal_IT_layers_Communication_layer_MSA
#continuous_integration_and_delivery_adoption

The Azure platform comprises the following components:

- Scheduler;
- Controllers;
- Key-value Database;
- API server;
- Several Nodes;
    - Hosts several Pods;
- Resource Manager;
- Container Orchestration engine.

Other components in the infrastructure are:

- Storage;
- Queues;
- Container registry;

- Container repository (optional);
- Network;
- Provisioning;
- Logs;
- Infrastructure Security.

#Communication_layer_MSA_components

The Azure platform realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Failover;
    - Microservice Security;
    - Configuration Management;
    - Networking/IP;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Container;
    - Microservices.

#Communication_layer_MSA_capabilities

The SCP Integration Suite enables the integration between SAP, non-SAP, cloud, and on-premise solutions through messages. The API management functionality provides access to simple, scalable, and secure digital assets through APIs and lets internal and external developers consume it. The Open Connectors functionality provides pre-built connectors and adapters to establish seamless connectivity between SAP and non-SAP applications. And the Integration Advisor accelerates the development of business-oriented interfaces, mappings, runtime artifacts and reduces integration efforts.

#bimodal_IT_layers_Communication_layer_CIP

The components of SCP are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;

        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;
            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;

- SMTP;
- Database Adapter;
- 3rd Party Adapters;
- Custom Adapters.

#Communication_layer_CIP_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

The IT organization has adopted Scrum and DevOps as its main SDLC methodologies.

#methodology_agile_framework_devops
#methodology_agile_framework_Scrum

Occasionally, it happens that an IT project is too big and too complex to deliver it in an agile way, like governmental changes or the implementation of a new and complex system within the Stable IT layer. In these scenarios, we are using Waterfall or the PRINCE 2 methodology to deliver the IT project. The same applies when we have to deal with tenders.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_traditional_framework_waterfall

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

I would say release management. Due to the Bimodal environment, the teams need to be continuously aligned and up-to-date. Through alignment and weekly communication mechanisms, the teams will be able to oversee the increments and align the multiple releases. Besides, this alignment will also help reduce connectivity and interoperability issues and keep the several teams informed.

#impact_release_management
#impact_continuous_alignment
#impact_communication

9: Do you use different methodologies across the different layers?

That there are four types of software delivery approaches recognizable within the IT landscape.

Within the Explorative IT environment, the majority of the increments are delivered through CI/CD. In some cases, the features and increments are delivered through Scrum.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery

#methodology_agile_framework_Scrum

In contrast, the systems and software within the Stable IT layer are initially implemented via a traditional approach. After go-live, the systems and software are enhanced and exploit through an agile approach. For the Stabilized Systems of Innovation applies that they are always developed and delivered through an agile approach (in our case Scrum).

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#methodology_agile_framework_Scrum

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

Within our organization, CI/CD is embedded through the use of Scrum and DevOps. We have Azure DevOps Boards in place to create, manage and maintain the Backlog, Sprints, User Stories, and associated Work items.

#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption
#methodology_agile_framework_devops
#methodology_agile_framework_Scrum
#continuous_integration_and_delivery_systems_and_software

The adoption of DevOps helps to set-up and keep the CI/CD pipeline filled. However, it is crucial that the organization is committed and supports this way of working. The organization needs to realize that the business requirements need to be clear, small, accurate, and concrete so the developer can easily perform the adjustments within the system. If this is not the case, CI/CD will fail because the requirements will hold up the developer in implementing the various features.

#methodology_agile_framework_devops
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_practices

10b: Which layers are affected by CI/CD?

Currently, the Explorative IT layer and microservices are supported by CI/CD. We are using Azure Pipeline to build the software and to realize the CI/CD pipeline.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption
#continuous_integration_and_delivery_systems_and_software_functionalities

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

As mentioned before, we have Azure DevOps in place to realize CI/CD. The following functionalities are incorporated into Azure DevOps:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, User Stories, and Work items;
- Azure Repos. This functionality is used as a Version Control tool;
- Azure Pipeline. This functionality is used to automate the build, test, and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. It uses Service Fabric and Azure App Service to perform the deployments;
- Azure Monitor and Grafana. These tools are used for monitoring, analysis, and visualization.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The common configuration practices for CI/CD are (not related to a certain environment):

- A CI/CD strategy is in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A Release manager tool needs to be in place to deploy the application towards any environment;
- A verification step should be in place before a deployment;
- Configurations and passwords should be easily accessible and always up-to-date;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

Multiple tools and components are used to develop and deliver the systems and software within the Stable and Explorative layers:

- A tool or system to manage the planning, requirements, IT projects, IT Service Management activities, and tickets;
- Functional and technical design tools;
- Collaboration tools;
- Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The tools and components are embedded in:

- Office 365;
- SharePoint;
- Testersuite. To create test plans and test scripts (traditional and agile development);
- JIRA and TOPdesk to support the planning, requirements, IT projects, IT Service Management activities, and tickets. Also, there is an integration between TOPdesk and Azure DevOps to support the Scrum teams.
- SAP Netweaver and SAP Workbench is in place to support the development in SAP ABAP;
- SAP Implementation Guide to configure the SAP systems;
- Java IDE (Integrated Development Environment), JDK, SDK, Eclipse, and Visual Studio to support the development of systems, applications, programs, methods, APIs, and services;
- SAP Transportation Management (to release the SAP transports).

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

Not really.

The systems and software of the Explorative IT layer are deployed in the cloud. For the Stable IT layer applies, that we are moving the systems and software from the on-premise and data center environments towards the cloud. The same happened with the SAP landscape. We had SAP running in a data center until 2018, and it is now being hosted on Azure.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud
#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#deployment_model_cloud
#deployment_model_movement_towards_cloud

I think that it is worth mentioning that a movement towards the cloud is taking place. The on-premise and data center models are becoming old-fashioned. However, this does not mean that all the systems and software will move towards the cloud. Some systems will still be deployed on-premise or in a data center environment because of the criticality and the purpose of that particular system.

#deployment_model_movement_towards_cloud
#notes_not_all_systems_can_be_deployed_in_the_cloud

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

The majority of the systems and software that are currently deployed on-premise or in a data center are moving slowly towards the cloud.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

I do not know the IT landscape's exact history, but the current IT landscape is dominated by SAP. Currently, the organization has adopted a cloud-first vision. This means that almost all the systems are, or will be, moved towards the cloud within the upcoming few years.

#organization_current_IT_landscape
#organization_IT_strategy

16: What is the future roadmap of the IT landscape of your organization?

First of all, the organization needs to decide what it will do with its SAP landscape because the support on the current version will stop. Secondly, the organization wants to move towards one integration platform instead of having a separate ESB and an Azure environment. And last but not least, the organization wants to focus more on innovation, like sensors, HoloLens, Digital Twin, AI, and Augmented Reality.

#organization_IT_roadmap
#organization_IT_strategy

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I believe that Bimodal Architecture will remain an important topic and approach within the IT domain for a long period. CI/CD will become the common development practice for the Explorative IT environment because CI/CD is gaining more and more popularity. Regarding the Stable IT environment, the traditional frameworks will make place for the agile frameworks to develop, enhance and maintain the various systems.

Conclusively, I believe that the two separate but highly integrated IT environments will become more visible within each IT landscape. Because every organization will need stable systems and software to leverage its operations and internal services and digital services to secure the engagement and interaction with its customers (digital offerings).

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

We are already using Scrum and DevOps. So, from that perspective, there is no impact.

#methodology_agile_framework_Scrum
#methodology_agile_framework_devops
#concept_bimodal_IT_agile_teams

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The priorities are set by a PO. Also, the costs of the teams are fixed and based on time and material.

#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

This question is not applicable from an innovation perspective because innovation has recently become an important point/topic on the agenda. Therefore, it has a different priority setting and budget allocation within the projects and organization.

#concept_bimodal_IT_prioritization

19: Can you elaborate on the technical debt of a Bimodal Architecture environment?

The technical debt is mainly applicable within the Stable IT layer because the related systems and software are complex and contain many custom codes. Also, there is an urgent need to expose the various functionalities as services. Subsequently, these services can be used to facilitate the systems and applications within the Explorative IT layer.

The systems and applications within the Explorative IT layer can also have some technical debt. The reason is that most of the systems and applications are built in Java, .NET, C/C++, or Node.js and invoke one or more APIs. However, if there are any issues or problems, it can be fixed very fast, mostly in another sprint.

#concept_bimodal_IT_technical_debt

Candidate 7 – EA-03

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I am already working for forty-two years within the IT domain. During this period, I have executed many IT projects, worked for multiple companies, and fulfilled several IT roles. For the last ten years, I am working for the current organization as an Enterprise Architect {EA-03}.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The organization operates within the Utility industry {Utility-02) and is together with other parties responsible for the water supply within The Netherlands. The IT capability exists of 45 internal IT resources and approximately 35 external resources, which are supplied by one or more external parties. All the IT resources are assembled in agile teams (Scrum) to perform all the IT-related tasks, activities, and work.

#company_profile
#IT_capability

For the IT landscape applies that some components are fully outsourced, like the maintenance and exploitation of our office software (SharePoint, Office 365, etc.) and the hosting of SAP.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I am already familiar with the Bimodal concept from Gartner.

For me, Mode 1 reflects the slow speed of change. The goal is long-term. And the associated business value is created at the end of the implementation or just after the systems' go-live. Therefore, the agile way of working is not always needed to achieve the goal. Also, the value does not change a lot during its life cycle.

Let's take SAP as an example. The system starts generating value after the completion of the green- or brownfield implementation or the go-live. Also, the core functionalities stay stable during its life cycle with a limited number of adjustments.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_characteristics

In contrast, Mode 2 is flexible, fast-changing, and adds business value after every release cycle (after one or two iterations depending on the increment). Besides, the increments' lifecycle can be very short due to the hostile and changing environment in which it operates.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics

Also, there is a tight dependency between Mode 1 and Mode 2. Without a stable Mode 1, it is impossible to facilitate and utilize a Mode 2 system or application.

#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The main components are:

- Mode 1. This layer includes transactional systems and software, such as SAP. Mode 1 also comprises mobile and web applications that are built to support a specific goal, such as the Meter Reading App. Note that these applications are initially developed and delivered in Mode 2. However, when the applications reach their purpose, the developments stop. Subsequently, a decision is taken whether to move the applications and their associated APIs to Mode 1 or not.
- Mode 2. This layer includes mobile and web systems to establish intuitive interactions between the organization and its stakeholders. Because there is a continuous need to fulfill the expectations, the systems can change rapidly or be replaced by another application.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

The models are recognizable. As already indicated, Mode 1 comprises the Systems of Record (SAP ECC, SAP IS-U, etc.), Systems of Differentiation (SAP C4C, SAP SuccessFactors, Youforce, etc.), and the Stabilized Systems of Engagement (Digiduinkaart App, Meter Reading App, etc.).

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#concept_Pace-layerd_Application_strategy_terminology

In contrast, Mode 2 comprises the Systems of Innovation and Systems of Engagement. The purpose of these systems is to improve and enhance the user and customer experience. Therefore, the layer and the incorporated systems can be referred to as "fit for purpose." This definition indicates that the layer and associated systems are highly focused on exploration and experimentation. Systems and software that are typically incorporated into Mode 2 are:

- Applications for mobile and smart devices;
- Portals;
- Web applications;
- Dashboards and reports.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#concept_Pace-layerd_Application_strategy_terminology

Regarding the Communication layer and the associated service-based environment, I like to make an additional comment. We are integrating objects and data instead of services. In my view, a service is nothing else than a container that contains specific attributes and values that belongs to a particular object, like a customer, sales order, or service order.

#bimodal_IT_layers_Communication_layer

Interviewer: I agree with that statement. However, to create the Reference Architecture models and make them understandable, I would like to refer to services. We can also distinguish between SOAP and REST APIs if you want?

That is not needed. But from that perspective, we have a Communication layer in place that establishes the connection and communication patterns within and between Mode 1 and Mode 2. Currently, the SAP Cloud Platform (SCP) and Biztalk/Windows Communication Foundation (WCF) are used as middleware.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_integration_patterns_through_Communication_layer

#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP

I also agree with the statement that some Mode 2 systems and applications are moving to Mode 1 once they are stabilized/matured.

#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

Typically, Mode 1 systems and functions are:

- SAP ECC (Business Suite 7);
    - Financial Management (Accounting, Controlling and Costing);
    - Sales Management;
    - Material Management (MRP);
    - Production Management (Manufacturing);
    - Quality Management;
    - Procurement;
    - Supply Change Management;
    - Warehouse and Inventory Management;
- SAP IS-U;
    - Device Management;
    - Work Management;
    - Contract Management for moving in/out;
    - Consumption Management;
    - Billing and Invoice Management;
    - Customer Service;
- SAP CRM;
    - Marketing (Campaigns);
    - Account and Contact Management;
    - Interaction Center;
    - Service Management;
    - Contract Management;
    - Back-Office;
- SAP C4C;
- SAP SuccessFactors;
- SAP Fiori;
- Genesys;

- Call Center Management;
  - Social Media Management;
- Youforce;
  - Human Capital Management (HRM);
- GIS;
  - Field Operation Management;
  - Data Management;
  - Asset Management;
- Xaris;
  - Planningstool;
- Esize
  - Invoice Management
- Microsoft Outlook;
  - Email and Calendar Management;
- Industry-specific Apps;
  - Meter Reading App;
  - Digiduinkaart App;
- Power BI;
  - Business Analytics;
  - Business Intelligence;
- Sensor APIs;
- Internal APIs.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Typically, Mode 2 systems and functions are:

- Office 365;
- SharePoint;
- Teams;
- Power Apps;
- Portals (Sitecore);
  - Customer Portal;
  - Employee Portal;
  - Partner Portal;
  - Supplier Portal;
- ArcGIS (Esri);
  - Data Management App;
- GeoWEB;
- Power BI;
  - Dashboards;
  - Reports;
- Web Apps;
- Mobile Apps;
  - iOS Apps;
  - Android Apps;

- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of Mode 1 are:

- High stability;
- Reliable;
- Secure;
- Facilitator for Mode 2;
- A Minimum of errors are allowed;
- Next to agile approaches, also traditional methodologies can be involved in the development of a system;
- Fail-over strategy in place to guarantee business continuity;
- Focus on standardization;
- The systems and software of Mode 1 are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The main characteristics of Mode 2 are:

- Systems and applications are easily and fast adjustable;
- Rapid delivery oriented. Often and frequent deployments;
- Off-line capability is a necessity to ensure 24/7 availability and for a high experience;
- Focus on innovation, experimentation, and exploration;
- Adds business value after almost every iteration (within 1 or 2 sprints);
- Requirements and planning change continuously.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

We are using SCP and Biztalk as middleware. The Biztalk server is used to integrate the systems that are deployed on-premise and within the data center. And SCP is responsible for the integration between SAP and non-SAP systems. Also, we have an Azure platform in place that is used to leverage the microservices.

#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA

As for the components, Biztalk comprises:

- Message Builder;
- Adapter Builder;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;
- Gateway;
- Configuration Manager;
- Message Security Manager;
    - Authentication;
    - Message Security;
    - Encryption;
    - Authorization;
- Exception handler;
- Event Handler;
- System Landscape/Tenant Manager;
- Monitoring;
- Visualizer;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

There is also a separate service registry tool (UDDI) that includes the following components:

- Service Catalog (for Service Discovery);
- Change Center;
- Import/Export Manager;

- Virtualization Manager;
- Policy Manager;
- Monitoring.

The service registry tool acts as a central library, through which the ESB can search and consume the discoverable services. The ESB itself acts as a service broker and gateway by sending, receiving, and routing the service messages.

#Communication_layer_ESB_components
#Communication_layer_ESB_capabilities

The components of SCP are:

- Cloud Integration Server;
  - Enterprise Messaging Hub;
    - Repository for Service Discovery and pre-build Integration Flows;
    - Integration Builder;
    - System Landscape/Tenant Manager;
    - Transaction Handler;
    - Configuration Manager;
    - Security Manager;
      - Message Security;
      - Encryption;
      - Authorization;
    - Event Handler;
    - Enterprise Service Repository;
    - Message Queues;
    - Exception handler;
    - Monitoring;
    - Visualizer;
  - Connection Hub;
    - Repository for Adapter Discovery;
    - Adapter Builder;
    - Adapter Configuration Manager;
    - Adapter Security Manager;
      - Authentication;
      - Identity Access Manager;
      - Authorization;
    - Monitoring;
    - Visualizer;
  - Integration Engine;
    - Adapter Engine;

        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Manager;
    - Repository for API Discovery;
    - API Builder;
    - API Configuration Manager;
    - API Test Manager;
- Business Process Manager;
    - Business Rules;
    - BPEL;
    - Workflow Builder;
    - Business Activity Monitor;
- Gateway;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Monitoring;
    - Visualizer.

#Communication_layer_CIP_components

And the components of the Azure platform are:

- Container Orchestration Platform;
    - Scheduler;
    - Resource Manager;
    - Container Orchestration engine;
    - Storage;
    - Queues;
    - Container Registry;
    - Image Repository;
    - Logs;
    - Container Security;
    - Load Balancer;
    - Clustering Manager;
    - Workflow Builder;
- API Manager;

- Service Repository;
- Mediation Flow Manager;
- Traffic Manager;
- Configuration Manager;
- Exception Handler;
- Policy Manager;
- Version Manager;
- Call Handler;
- Rate Limit Manager;
- Circuit Breaker;
- Security Manager;
    - Authentication;
    - Identity Access Manager;
    - Authorization;
- Monitoring;
- Visualizer.

#Communication_layer_MSA_components

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

Within the IT organization, we are making use of Scrum. And in the case of CI/CD, we are using DevOps.

#methodology_agile_framework_Scrum
#methodology_agile_framework_devops
#methodology_agile_framework_continuous_integration_and_delivery

In practice, however, the agile way of working is not always sufficient. This scenario occurs when a particular IT project is too big and complex to break it down into small stories/increments or when it needs to realize a specific goal, such as governmental changes, or during the implementation of a Mode 1 system. The same also apply to tenders. These trajectories are heavily relying on traditional approaches to reach the defined goal(s).

#Slow-Speed_systems_and_software
#methodology_traditional_framework

Note, even in Waterfall, some tasks and activities can be executed in an agile way.

#methodology_traditional_framework_waterfall
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

Check and rethink the release strategy because each Mode has its speed and release cycles.

#impact_release_strategy
#impact_release_management

Due to the increase in APIs and services, the teams' cooperation has become more critical. Therefore, it is crucial as an IT organization to conduct a weekly cross-team alignment session. In this session we can determine the dependencies and discusses upcoming developments. The session helps us to mitigates problems, and to support and exploit the (future) IT landscape.

#impact_continuous_alignment
#impact_communication

9: Do you use different methodologies across the different layers?

Yes, we are using different methodologies across the Modes. Mode 2 focuses on agility and the continuous delivery of increments. Therefore, agile methodologies are preferred.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

Mode 1 uses a mix of SDLC approaches. For instance, the Systems of Differentiation, Stabilized Systems of Innovation, and Stabilized Systems of Engagement are implemented, enhanced, and exploited through an agile approach. In comparison, the majority of the Systems of Record are initially implemented using a traditional approach. Once a System of Record is live in production, an agile method is used to exploit the related functionalities and features.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

Currently, CI/CD is only applicable within Mode 2 and the microservice environment. We are using Azure DevOps to realize and ensure CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption
#continuous_integration_and_delivery_systems_and_software

10b: Which layers are affected by CI/CD?

As already indicated, CI/CD is only used within Mode 2 and the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

Although, I believe that all layers will be supported by CI/CD eventually. The problem is that the Mode 1 systems vendors lack behind when it comes to agility adoption. However, the vendors are moving and are taking significant steps towards the cloud. This movement will change how the vendors and we are developing, configuring, maintaining, and exploiting the systems and applications. Consequently, the agile way of development and especially CI/CD, will become the standard SDLC approach for all layers within the upcoming 3 – 5 years.

#bimodal_IT_layers_Slow-Speed
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The capabilities of Azure DevOps are:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, User Stories, and Work items;
- Azure Repos. This functionality is used as a Version Control tool, and we are using Git as a Repository;
- Azure Pipeline. This functionality has an integration with Jenkins. The Jenkins automation server is used to automate the build through Maven, perform the tests via Selenium, and to publish the activities;
- Azure Release. The release functionality is used to automate the deployment step. It uses Service Fabric and Azure App Service to execute the deployments;
- Azure Monitor and Grafana. These tools are used for monitoring, analysis, and visualization.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following practices are essential in the case of CI/CD:

- A CI/CD strategy should be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small;
- The developer should test the increment rigorously on his/her local environment;

- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before every deployment;
- Configurations and passwords should be easily accessible and always up-to-date;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

The following tools and components are utilized to develop and deliver the systems and software within both Modes:

- A tool to manage the planning;
- Requirements tool;
- An IT Service Management and ticketing system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The tools and components are embedded in the following systems and applications:

- TOPdesk. This system is used for the planning, requirements gathering, and analysis. Also, there is an integration between TOPdesk and Azure DevOps to support the Scrum teams;
- JIRA is used for IT projects, IT Service Management activities, and tickets;
- Office 365 is used for the functional and technical specifications;
- Teams is used for collaboration;
- SharePoint is used to store all documentation;
- For the development and configuration of increments, systems, and software, the following tools are used:

- SAP NetWeaver and SAP Workbench is in place to support the development in SAP ABAP;
- SAP Implementation Guide to configure the SAP system;
- SDK (SAP Development Kit);
- JDK (Java Development Kit);
- JavaScript, PHP, and HTML5;
- Visual Studio (.NET, Python, C/C++ and Node.js developments);
- Visual Studio extensions (PowerShell and YAML);
- Eclipse;
- Xcode 12 and Swift UI;
- Android Studio;
- Azure DevOps;
- Microsoft Intune is used to configure and maintain mobile devices (mobile device management (MDM) and mobile application management (MAM));

- We are using several Test Management Tools;
  - Azure Test Plan;
  - Testersuite. To create test plans and test scripts (traditional and agile development);
- For the deployment, we are using the following tools:
  - Azure Service Fabric and Azure App Service for deployments of CI/CD increments;
  - SAP Transportation Management to release and deploy SAP transports;
  - We have also SAP Solution Manager in place to execute the transportation activities;
  - Eclipse is used for SAP IFlow creation and deployments;
  - To deploy Java software, we are making use of the Java Deployment Toolkit;
  - Apple Business Manager is used to deploy iOS devices;
- Monitoring tools:
  - Application Insights;
  - Azure Alerts;
  - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

From my perspective, there is no relationship between the deployment models and Bimodal Architecture. In practice, the majority of the Mode 2 systems and applications are deployed in the cloud. In contrast, Mode 1 systems and software can be hosted in any environment because the deployment model depends mainly on an organizations' IT strategy. In our case, the SAP landscape is running in the cloud.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud
#bimodal_IT_layers_Slow-Speed

#deployment_model_data_center
#deployment_model_on-premise
#deployment_model_cloud

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

As already indicated, there are no common deployment practices in place. It depends on the organizational IT strategy and vision where and how a system is hosted.

#deployment_model
#organization_IT_strategy

15: What were the major milestones that lead to the current set-up of the IT landscape?

The current landscape is inherited from 2002. SAP is the biggest component in our IT landscape. As an organization, we have adopted a cloud strategy in 2015/2016. This strategy led to the implementation of Office 365, Power BI, and the movement of the SAP landscape towards Azure.

#organization_IT_milestones
#organization_current_IT_landscape
#organization_IT_strategy

16: What is the future roadmap of the IT landscape of your organization?

We have a project called "Digital Transformation." The goal of this project is to migrate towards a new system landscape. Also, the replacement of the ESB is on the roadmap. We want to adopt Azure Integration Services, so all the middleware is available in the cloud instead of on-premise.

In the upcoming two years, we have to decide what we will do with our SAP landscape. It is currently not clear if we will replace the SAP systems or perform an upgrade towards SAP S/4HANA.

And, last but not least, we have to rethink our call center strategy due to the changed market and increased use of smart devices.

#organization_IT_strategy
#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Eventually, Mode 1 will follow Mode 2. In other words, the Modes will vanish. The fact is that vendors and suppliers of Mode 1 systems need to change and adjust their business models to compete with the competitors of Mode 2.

On the other hand, we will still need stable and secure core systems to store our business-critical objects and data. Also, innovation will never stop. So, the future will determine what will happen with the Bimodal IT concept.

#concept_bimodal_IT_future
#IT_trends

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

We are already making use of Scrum teams, so the team set-up is not jeopardized.

#methodology_agile_framework_Scrum
#concept_bimodal_IT_agile_teams

Also, the transition towards agile has shown that the architecture needs to change. An agile architecture is different in comparison with a traditional Architecture. Therefore, the main question is: does the current architecture provide the procedures, principles, and guidelines to adjust a system in an agile way? If not, we have to define a sub-architecture or another viewpoint to facilitate the iterative way of working.

Mode 1: Traditional architecture: more complex, so a detailed architecture is needed.
Mode 2: Agile architecture to support agility.

#concept_bimodal_IT_traditional_architecture
#concept_bimodal_IT_agile_architecture

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The priority is set by the PO. The costs of the Scrum teams are fixed and based on time and material.

#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

Also, IT Service Management does not distinguish between both Modes.

#IT_Service_Management

19: Can you elaborate on the technical debt?

The technical debt is minimal in Mode 2. The reason is that most of the systems and applications are built in Java, .NET, C/C++, or Node.js and invoke one or more services. However, if there are any issues or problems, it can be fixed very fast, mostly in another sprint.

In contrast, Mode 1 has a high technical debt because the incorporated systems and software can have a lot of custom code. Also, there is an urgent need to expose the various functionalities as services. Subsequently, these services can be used to facilitate the systems and applications within Mode 2.

#concept_bimodal_IT_technical_debt

Candidate 8 – ITL-02

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I have fifteen years of experience in SAP with the modules Finance and Controlling, Material Management, Quality Management, Procurement, Sales and Delivery, Production Planning, Customer Service, and Inventory Management. On top of that, I have seven years of experience in integrating SAP with external systems, like Salesforce, Teamcenter, Net@Pro, Ariba, ADP Workforce Now, etc.

I am one of the five Integration and Technical Leads {ITL-02} within the organization. My main focus area is SAP. Since the SAP landscape represents the backbone of the IT landscape, I am often asked to involve in the selection, adoption, and implementation trajectories of non-SAP systems.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

Our company manufactures and distributes smart thermal comfort solutions {Consumer Goods-01} over more than 100 countries. The organization comprises several brands. Therefore, our IT organization is divided into Group IT and Local IT (each brand has its own local IT team).

#company_profile
#IT_capability

My work domain is part of the Center of Excellence (CoE) and falls under the umbrella of Group IT. The CoE team counts 25 members, and Group IT itself has approximately around 150 employees. Regarding the employee count of Local IT, I have to refer you to the Enterprise Architect or the Chief Technology Officer (CTO).

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal Architecture comprises two different but coherent IT environments that communicate with each other through services/APIs. One IT environment is stable and represents the organizations' Operational Backbone, while the other IT environment is explorative and forms the Digital Platform.

#bimodal_IT_layers_Slow-Speed
#concept_digital_transformation_operational_backbone
#bimodal_IT_layers_Fast-Speed
#concept_digital_transformation_digital_platform
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The stable IT environment consists of:

- Systems of Record. For example, SAP S/4HANA and Teamcenter;
- Systems of Differentiation, such as Dynamics 365 CRM, ServiceNow, and the Egencia Travel Management system;
- Stabilized Systems of Innovation and Engagement, such as SAP Fiori Applications and the Egencia Travel Management App.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#concept_Pace-layerd_Application_strategy_terminology

The Explorative environment consists of:

- Systems of Innovation, such as AssistAR (from Siemens), and other IoT solutions;
- Systems of Engagement, such as Sitecore, Power Apps, and other web and smart device applications.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#concept_Pace-layerd_Application_strategy_terminology

Interviewer: The different environments are clear, but can you indicate why you are dividing the Systems of Innovation/Engagement between a stabilized and Explorative environment?

Not all digital applications are explorative or bounded to rapid changes. Let's take SAP Fiori as an example. When we have realized a Fiori App, the application itself hardly changes. We may need to realize a new version because we need to add a new field or adjust an attribute in the underlying OData service. However, there is no need to adjust the application and associated services weekly or two-weekly. The same applies to other features and applications in Fiori. Therefore, these applications can be marked as stable and will never be part of the Explorative environment. Consequently, we also do not use a CI/CD pipeline to realize or enhance these applications.

#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

When we compare the Stabilized Systems of Engagement with the Systems of Engagement within the Explorative environment, we see a big difference in the release cycle frequency and the delivery approach. When we look, for example, at Power Apps, Facebook, or Sitecore, which are all integrated into our IT platform, we are obligated to release new features or versions daily to meet the expectations. To facilitate and realize the daily deployments, we have a proper CI/CD pipeline in place. Otherwise, we will never be able to meet these expectations.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

We could say that there is a need for two Digital Platforms (internal and external) within an organization. One platform has an internal impact and realizes digital applications and solutions to support, simplify, improve, and automate specific tasks and activities. Also, the information and metrics providing solutions and applications are incorporated into this platform. Conclusively, the internal Digital Platform comprises digital applications and solutions that focus on the operations of a company and provides the mechanisms to improve this.

#Slow-Speed_stabilized_systems_of_innovation
#Slow-Speed_characteristics

The second Digital Platform focus is to facilitate digital applications and solutions to improve and gain market share. This can be achieved by promoting digital branding and advertisements, offering digital products and services, enhancement of the user and customer experience, embracing and leveraging omnichannel capabilities, integration with social media, etc. Therefore, this platforms' digital applications and solutions mainly focus on exploration, experimentation, and innovation to improve engagement and interaction. Also, all the self-service features are realized through this platform. Some of the solutions, like chatbots, stabilize at a given moment and moves towards the Stable environment. The reason is that these technologies, applications, and solutions can have a positive effect on the operations by making the processes more efficient and simpler.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_integration_patterns_Fast-Speed

#Fast-Speed_characteristics
#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed

Interview: Thank you for this clarification. It's evident why we need to deviate the digital applications and solutions between a stabilized and an Explorative environment.

Before we move to the next question, I also like to add something else.

Integration is an important aspect of a Bimodal environment. The Explorative environment has a tight dependency with the Stable environment.

#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

In general, there are three integration patterns applicable. The first integration pattern is visible within the Stable environment. This environment has a complex EAI pattern because all systems are connected through point-to-point integration or via a service-based approach. If we look at the most critical integration patterns within this environment, we can define the following connections:

- SAP connections. The internal SAP interoperability mechanisms, like RFC, EDI (IDocs), and adapters, are used to connect to various SAP systems.
- SAP PO and proxies. The functionalities of SAP can be exposed as services due to the use of proxy technology. This mechanism creates services and sends it to SAP PO to communicate with the non-SAP systems and applications like CRM Dynamics 365, Net@Pro, or the Stabilized Systems of Engagement;
- SAP Enterprise Services. These out-of-the-box services are also used to communicate with the non-SAP systems and applications.
- SAP Cloud Platform (SCP). The integration between SAP S4/HANA and Ariba (SCM) is established through SCP. SCP is used to connect SAP, non-SAP, on-premise, and cloud applications. Whereas SAP PO is mainly focused on integrating applications and systems that are hosted on-premise or in a data center.
- SAP Gateway. The SAP Gateway is used to facilitate the SAP Fiori applications.
- T4EA. T4EA is an integration software that provides connectors to establish a connection between the Teamcenter system and other enterprise applications through different technologies, such as file exchange (XML), web services, database connectors (JDBC and JMS), and messaging (API).
- T4S4. This is an out-of-the-box adapter to connect Teamcenter directly with SAP S/4HANA.

Besides the mentioned connections, several other systems interact and communicate with each other within the Stable environment.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software

#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP

The second integration pattern is visible between the Stable and Explorative environments. The communication between these two environments is established through an Integration layer, which embeds a service bus, such as SAP PO or T4EA, a cloud integration platform, like SCP, or a microservice platform, such as Azure. This layer can even comprise all three middleware mechanisms to realize and establish the IT landscape's integration patterns.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

The third integration pattern is visible within the Explorative environment and used to connect the digital systems and applications with each other through services. In general, like Azure, a microservice environment is used to create, maintain, and integrate the APIs within this environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can link our IT landscape to the presented models. However, the Stabilized Systems of Innovation and Engagement are missing in your models. In my opinion, you should add these groups of systems, and the associated integration patterns, into the Slow-Speed layer to cover all the aspects of this environment.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_stabilized_systems_of_innovation

Also, I would not use the terminologies "Slow-Speed" and "Fast-Speed." I can imagine that you got this description from the Two-Speed IT approach, but it can confuse the business.

#bimodal_IT_layers_terminology

Interview: Yes, you are right. I got the description from McKinsey. However, can you explain why the layers' descriptions (within the presented models) could be misleading?

The problem with the terminologies is that it implicates that some systems are slow while other systems are fast. That is not true. As already indicated in the previous question, SAP can be described as stable, secure, and robust. However, SAP also contains modern and user-facing components, such as Fiori applications and OData services. Another example is the Egencia App. This application supports the booking process, is modern, and does not change frequently. Therefore, it is embedded into the Stable environment instead of the in the Explorative environment.

#Slow-Speed_systems_and_software
#Slow-Speed_characteristics
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The following systems and functionalities can be found within the Stable environment:

- ERP and associated systems, such as SAP S/4HANA, Ariba, and Dynamics 365 CRM;
    - Financial Management (Accounting, Controlling, Costing, and Treasury);
    - Quote to Cash and After Sales (Sales Management, Loyalty Management; Event Management, Warranty Management, Service Management, Case Management, and Invoice Management);
    - Source to Pay (Procurement, Contract Management, SRM, Invoice Management, and Sourcing);
    - Manufacturing (Production Management, Order Management; and Quality Management);
    - HRM (Compensation Management, Recruitment, and Staffing);
    - SCM (MRP, Transport Management, Inventory Management, and Warehouse Management);
- SCADA systems, such as Teamcenter and Net@Pro;
    - PLM (Material Management, Product Configuration Management, and Quality Management);
- Collaboration, travel and reservation systems, such as Microsoft Outlook and Egencia Travel Management;
    - Email and Calendar Management;

- Travel and Expense Management;
- Business Intelligence systems, such as Power BI;
    - Business Analytics;
    - Business Intelligence;
- Ticketing and IT Service Management tool, such as ServiceNow;
- Stabilized Systems of Innovation and Engagement;
    - SAP Fiori Apps;
    - Egencia Travel Management App;
- Internal APIs and web Apps.

#Slow-Speed_systems_and_software
#Slow-Speed_systems_and_software_functionalities
#Slow-Speed_stabilized_systems_of_innovation

The following systems and functionalities can be found within the Explorative environment:

- Portals (Sitecore);
    - Customer Portal;
    - Employee Portal;
    - Partner Portal;
    - Supplier Portal;
- Office automation systems;
    - SharePoint;
    - Power Apps;
- Collaboration apps;
    - Teams;
    - WebEx;
    - Facebook;
    - Linked-In;
- Self Service systems;
    - Self-Service Portal;
    - Self-Service Apps;
- Analytics and Data Management tools;
    - Dashboards;
    - Reports;
- Augmented Reality solutions;
    - AssistAR;
- Web Apps and other smart device applications (mobile);
- IoT solutions;
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of the Stable environment are:

- Stable, robust, and reliable;
- Security is essential and an ongoing topic;
- The systems are supporting the business processes and holds business-critical objects and data;
- The systems do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- The agile way of working is not always possible. Results in long release cycles;
- Fail-over strategy in place to guarantee business continuity and high availability;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance;
- High influenced by third parties.

#Slow-Speed_characteristics

The main characteristics of the Explorative environment are:

- Features are small and independent;
- Flexible. The applications can change daily;
- Off-line capability is a necessity to ensure 24/7 availability and for a high experience;
- Rapid development and delivery of software; Often and frequent deployments;
- Focus on innovation and exploration;
- Less focus on Risk Management. Errors are already factored in;
- Adds business value continuously. At each release or iteration;
- Planning changes constantly;
- Apps can contain business-critical data, but it is not a standard requirement;
- Only an Agile way of development is supported;
- The environment compromises mostly modern systems and applications (cloud-based).

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

As indicated in question 2, multiple middleware systems are responsible for the connectivity and integration of the various systems, software, applications, and solutions within the IT landscape. All systems and applications deployed on-premise and in the data center are connected through the ESB. This middleware mechanism ensures the performance and that the systems are loosely coupled.

#deployment_model_on-premise
#deployment_model_data_center
#bimodal_IT_layers_Communication_layer_ESB

The (technical) components of SAP PO are:

- Integration Builder;
    - Enterprise Service Repository;
        - Creation and Discovery of Services;
    - Integration Directory;
        - Set-up of Mediation Flows;
- Integration Server;
    - Adapter Engine;
        - Transformations;
    - Integration Engine;
        - Orchestration;
        - Routing;
        - Publish and Subscriptions;
    - Business Process Engine;
        - Rules;
- System Landscape;
    - System Landscape/Tenant Manager;
- Configuration and Monitoring;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Mediation flow Manager;
    - Event Handler;
    - Adapter Manager;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Logger;
    - Auditor;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;

- Rules;
- BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Message Tracking and Throttling;
  - Exception Management;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - XML;
  - FTP and SFTP;
  - SMTP;
  - Flat File;
  - Database Adapter;
  - 3rd Party Adapter;

- Custom Adapter.

#Communication_layer_ESB_capabilities

SCP is used to develop and manage end-to-end integration scenarios across cloud-based and on-premise solutions.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of SCP are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;

- Workflow Builder;
  - Set-up of Workflows;
  - Editing of Workflows;
- Access Controller;
  - Identity Access Manager;
  - Set-up of Authorization;
- Logger;
- Auditor;

o Integration Engine;
  - Adapter Engine;
  - Orchestration Engine;
  - Transformation Engine;
  - Routing Engine;
  - Rule Engine;
  - Publisher and Subscriber Engine;

- API Management Service;
  - Repository for API Discovery;
  - API Designer;
    - Set-up of APIs;
    - Edit APIs;
  - API Configuration Manager;
  - API Test Manager;
- BPM Service;
  - Business Process Manager;
    - Set-up of Business Rules;
    - BPEL;
  - Workflow Builder;
    - Set-up of Workflows;
    - Editing of Workflows;
  - Business Activity Monitor;
- Gateway Service;
  - Traffic Manager;
  - Exception Handler;
  - Policy Manager;
  - Version Manager;
  - Call Handler;
  - Logger;
  - Auditor.

#Communication_layer_CIP_components

The SAP cloud integration platform realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Service Discovery;
  - Call handling;
  - Message Tracking and Throttling;
  - Rate Limits;
  - Traffic Management;
  - Exception Management;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - OData;
  - XML;
  - FTP and SFTP;
  - Flat File;
  - SMTP;

  - Database Adapter;
  - 3rd Party Adapters;
  - Custom Adapters.

#Communication_layer_CIP_capabilities

And last but not least, we are using Azure as a Container Orchestration Platform to connect the Explorative environment and to realize and leverage the microservices.

#bimodal_IT_layers_Communication_layer_MSA

The (technical) components of this platform are:

- Container Orchestration Platform;
  - Container Orchestration Engine;
  - Scheduler;
  - Controllers;
  - Key-value Database;
  - API Server;
  - API Gateway;
  - And a Resource Manager.

#Communication_layer_MSA_components

The Azure environment realizes the following capabilities and functionalities:

- Container Orchestration Platform;
  - Container Orchestration;
  - Container clustering;
  - Image Discovery;
  - Container Security;
  - Network settings;
  - Key Vault;
  - Load Balancing;
  - Scheduling;
  - Rollouts and Rollbacks;
  - Self-Healing;
  - Workflow Management;
- Microservices Mesh or an API Gateway (whereas an API Gateway is more preferred);
  - Service Discovery;
  - Messaging;
  - Routing (of services);
  - Call handling;
  - Transformation;

    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Fail-over;
    - Microservice Security;
    - Configuration Management;
    - Networking/IP;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Container;
    - Microservices.

#Communication_layer_MSA_capabilities

The infrastructure of the Container Orchestration Platform also comprises:

- Storage;
- Queues;
- Container registry;
- Networking;
- Provisioning;
- Infrastructure Security;
- Logs;
- And Serverless Computing.

#Communication_layer_MSA_components

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

In the Explorative environment, the development and delivery of the software (increments) are achieved through CI/CD or by making use of another agile approach. In general, the majority of the increments are delivered through CI/CD. In all other cases, the preferable method is Scrum.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum

In the Stable environment, the systems are developed and delivered through a mix of traditional and agile approaches. The most famous delivery approach is Waterfall. This approach is used during the implementation, adoption, or integration of a complex system into the IT landscape. After go-live, a switch is made towards an agile approach, such as Scrum or DevOps, to perform the exploitation. And, for extensive IT projects applies that the PRINCE2 method is used.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_traditional_framework_waterfall
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

Because we are using various delivery approaches, there is a constant need for alignment between the different environments and teams. The fact is that an uncontrolled release can jeopardize the whole operation. To avoid this, we have a weekly session with all the integration teams to create awareness of the various changes and help guide the releases. This approach reduces or even eliminates connectivity errors and other (integration) issues.

#impact_continuous_alignment
#impact_release_management

9: Do you use different methodologies across the different layers?

Yes, most of the systems and applications within the Explorative environment are realized and facilitated by CI/CD. In some situations, the increments are realized and delivered through Scrum. In contrast, the Stable environment uses a mix of delivery approaches (traditional and agile) to deliver, maintain, and exploit a system.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum
#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

As indicated in the previous question, most increments are delivered through CI/CD within the Explorative environment. The same applies to the increments within Azure (microservice environment).

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA

#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

This means that the developers are committing and updating the repository daily, even multiple times in a day. In the night, the build, test, publish, and deploy activities are executed automatically. Once a package is generated, it is automatically deployed on the Acceptance environment for a UAT. When the UAT is approved the next day, the automation tool will queue the package until a specific time in the evening. When the release time is met, the package will be automatically released and deployed in the Production environment.

#continuous_integration_and_delivery_components

10b: Which layers are affected by CI/CD?

Only the Explorative layer and the microservice environment are affected by CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

Within our IT landscape, CI/CD is enabled through Azure DevOps.

Azure DevOps provides the following functionalities:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, and Work items;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. This functionality is used to automate the build, test, and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. We are using Azure Service Fabric and Azure Functions to realize the automated deployments;
- Azure monitor. This tool is used for monitoring and analysis.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following conditions need to be in place to realize and facilitate CI/CD:

- A CI/CD strategy is obligatory;
- Recovery strategy;

- A proper branching structure should be in place;
- Increments need to be small. This approach also minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before every deployment;
- Configurations and passwords need to be easily accessible and always up-to-date;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

We have the following tools in place to support the developments within the Stable and Explorative environments:

- A tool to manage the planning;
- A tool to manage the requirements;
- A project and IT Service Management system;
- A ticketing system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release and Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

- ServiceNow for the creation of tickets and to support the project management and IT Service Management activities;
- Office 365 is used to capture and manage the requirements. This tooling is also used to create functional and technical specifications and documentation.
- SharePoint to store all documentation;
- We are using the following development kits:
    - SDK (SAP Development Kit);

    - SAP NetWeaver and SAP workbench for ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - JDK (Java Development Kit);
    - Visual Studio for .NET, Python, C/C++, and Node.js developments;
    - Visual Studio extensions (PowerShell and YAML);
    - C/C++ for Teamcenter;
    - Eclipse for SCP and Teamcenter extensions;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Azure DevOps;
- We are using several Test Management Tools;
    - Micro Focus UFT one for SAP, Azure, and Java;
    - Test Manager for Teamcenter;
- For the deployment, we are using the following tools:
    - CHARM in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for IFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
    - And to deploy Teamcenter functionality, we are using Deployment Center from Siemens;
- Monitoring tools:
    - Application Insights;
    - Azure Alerts;
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

Some of the tools and systems are deployed on-premise or in a data center. In comparison, the other systems are deployed in the cloud.

#software_delivery_life_cycle_systems_and_software
#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud

Interviewer: Thank you for the answer. Regarding the deployments of the tools and systems, I would like to ask you if you can elaborate on this in the next question.

Yes, of course.

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

There is no relationship between the deployment model and Bimodal Architecture. However, there is a noticeable trend. The Stable environment systems are mainly deployed on-premise or in a data center, such as SAP S/4HANA and Teamcenter. In contrast, most of the Explorative environment systems are deployed in the cloud.

#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

The same trend is noticeable in the Integration layer. The ESB systems (SAP PO and T4EA) are mainly deployed on-premise or in a data center, while SCP and Azure are deployed in the cloud and offered as a PaaS or iPaas solution.

#bimodal_IT_layers_Communication_layer_ESB
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#deployment_model_cloud
#deployment_model_cloud_Paas
#deployment_model_cloud_iPaas

Interviewer: Thank you for the answer. Can you also elaborate a little bit on the technical infrastructure of the mentioned deployment models? And maybe also explain what serverless computing means?

Let's start with serverless computing. Serverless computing provides the ability to run custom code on-demand and at scale in the cloud. It is not in the organizational or users' interest where the hardware and physical devices are located or how the underlying physical infrastructure is set-up. As a developer or user, you just log into the solution and start developing, configuring, and integrating systems and applications.

#deployment_model_cloud_serverless_computing

To continue with the other point, for the cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- or Hybrid models.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

In an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can be reached and executed via a private network established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

There is no common deployment practice for the layers. However, I believe that all deployment models will move towards the cloud in the upcoming few years.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

The current IT landscape originates due to several acquisitions and mergers, including their local IT.

#organization_IT_milestones
#organization_current_IT_landscape

16: What is the future roadmap of the IT landscape of your organization?

The roadmap is to simplify the current IT landscape by consolidating 500 applications back to 60 applications.

#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

The future of Bimodal Architecture is bright. It is inevitable for a company not to split up their IT environment into two separate environments and link this through an Integration layer. One IT environment will be stable and form the Operational Backbone to establish operational excellence. Whereas the other IT environment will focus on exploration and provide the digital offerings to its stakeholder to maintain, improve and gain more market share.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

Bimodal Architecture requires a different team set-up. The DevOps approach is the most suitable team set-up in a Bimodal environment. However, the business itself should also change to commit to the way of working of DevOps.

#methodology_agile_framework_devops
#concept_bimodal_IT_agile_teams

Also, an agile architecture differs from a traditional architecture. Therefore, a sub-architecture or another viewpoint needs to be designed to facilitate the iterative way of working.

#concept_bimodal_IT_traditional_architecture
#concept_bimodal_IT_agile_architecture

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

In the Stable environment, the priority setting for the systems and applications is mostly done by the business. Before a system can be built or adjusted, the IT organization or third-party must provide a detailed impact analysis to determine the IT projects' risks and costs. Once a steering community approves the estimations, the IT project can start with its development and delivery.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_estimations

In the Explorative environment, the priority is set by the PO. The costs of the associated agile teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt?

The technical debt is minimal in the Explorative environment. The reason is that most of the systems and applications are built in Java, .NET, C/C++, or Node.js and invoke one or more services. However, if there are any issues or problems, it can be fixed very fast, mostly in another sprint.

In contrast, the systems and applications within the Stable environment can have a high technical debt because they can contain custom build functionalities. Also, there is an urgent need to expose the various functionalities as services to realize and leverage the API platform.

#concept_bimodal_IT_technical_debt

Candidate 9 – EA-04

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I started at one of the brands in 2000. After a merger between two brands, my department became part of the newly established firm. As you already know, I am fulfilling the role of an Enterprise Architect {EA-04}. I have been fulfilling this role for the last sixteen years. Therefore, I know a lot about the evolution of the current IT landscape and how the SAP landscape has become its backbone.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The company is a leader in smart heating technology. It manufactures several brands and sells intelligent thermal comfort solutions {Consumer Goods-01} in more than 100 countries. Due to its size, the IT organization is divided into Group IT and Local IT (each brand has its own local IT team).

#company_profile
#IT_capability

I am part of Group IT. Group IT has around 150 employees. In contrast, Local IT has around the same number of employees. However, it is not divided equally over each brand. Some brands have only 5 IT employees, while other brands can have up to 30.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I am already familiar with the Bimodal IT concept from Gartner. In this concept, Gartner differentiates between Mode 1 and Mode 2. Mode 1 can be defined as "predictable" and comprises the back-end systems, while Mode 2 is focused on "exploration" and leans on agility.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#bimodal_IT_layers_Fast-Speed

Also, Gartner came a few years later with the Pace-layered approach. This approach is used to categorize, select, manage, and govern a system to support business change, differentiate, or innovate. The classification of a system is based on three groups. These groups are:

- Systems of Record;
- Systems of Differentiation;
- Or Systems of Innovation.

#concept_Pace-layerd_Application_strategy

From my perspective, Bimodal Architecture is a cohesive environment with two separate IT platforms that are tightly integrated through a Service layer. The first IT platform can be described as "stable" and comprises the Systems of Record, Systems of Differentiation, and some Systems of Innovation. The Systems of Innovation are either; evolved, stabilized, still needed, and, therefore, marked as mature. Or the particular system does not solely focus on exploration but also delivers a certain amount of operational value. The Stable IT platform forms the Operational Backbone of the organization. In contrast, the second IT platform can be described as "explorative" and comprise the Systems of Innovation and an external Developer Platform. This IT platform can be eulogized as the Digital Platform of the organization and provides digital products and services to its stakeholders.

This definition also covers the building blocks for Digital Transformation.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#concept_digital_transformation_operational_backbone
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#concept_digital_transformation_digital_platform
#bimodal_IT_layers_Communication_layer

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The Stable IT platform consist of:

- Systems of Record, like SAP S/4HANA and Teamcenter;
- Systems of Differentiation, like Dynamics 365 CRM, ServiceNow, and Egencia Travel Management;
- Stabilized Systems of Innovation, such as sensor APIs, SAP Fiori, and the Egencia Travel Management App.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software

#Slow-Speed_stabilized_systems_of_innovation
#concept_Pace-layerd_Application_strategy

The Explorative IT platform consists of:

- Systems of Innovation, such as AssistAR (from Siemens), Sitecore, Power Apps, Portals, Social Media, Collaboration applications, etc.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#concept_Pace-layerd_Application_strategy

And the Service layer consists of the following components:

- Service Bus, like SAP PO and T4EA;
- Cloud Integration Platform, like SAP Cloud Platform (SCP) and Azure Integration Services;
- Microservice environment and associated Container Orchestration Platform, like Azure;
- Web services and Microservices.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

Also note, that the Service layer is an important component within the Bimodal Architecture environment because the Explorative IT platform is built on top of the Stable IT platform.

#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

Interviewer: That is an interesting statement. One of the Integration and Technical Leads {ITL-02} notified me that a dependency exists between the stable and explorative environment. I also got to know that there are three integration patterns applicable when we talk about Bimodal Architecture. Can you maybe also elaborate on these integration patterns?

Yes, of course! As already mentioned by the Integration and Technical Lead of SAP, there are three integration patterns visible within the IT landscape. The first pattern can be determined within the Stable IT platform. We have an EAI strategy to connect the various systems via a direct point-to-point integration through the Service layer or a combination of both patterns. The Service layer compromise at least a Service Bus and can have a Cloud Integration Platform, a microservice environment, or a combination of these middleware technologies in place. The

EAI strategy supports the interaction between the Systems of Record and Systems of Differentiation, like SAP S/4HANA and Dynamics 365 CRM. Or, even let a System of Record interact directly with a matured/stabilized System of Innovation via services, like SAP S/4HANA with the Egencia Travel Management App, and of course with Fiori.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

If we look closer at the integration patterns within this IT environment, we can determine the following technologies and components:

- SAP connections. The internal SAP interoperability mechanisms, like RFC, EDI (IDocs), and adapters, are used to connect to various SAP systems.
- SAP PO and proxies. The functionalities of SAP can be exposed as services due to the use of proxy technology. This mechanism creates services and sends it to SAP PO to communicate with the non-SAP systems and applications like CRM Dynamics 365, Net@Pro, or the Stabilized Systems of Engagement;
- SAP Enterprise Services. These out-of-the-box services are also used to communicate with the non-SAP systems and applications.
- SAP Cloud Platform (SCP). The integration between SAP S4/HANA and Ariba (SCM) is established through SCP. SCP is used to connect SAP, non-SAP, on-premise, and cloud applications. Whereas SAP PO is mainly focused on integrating applications and systems that are hosted on-premise or in a data center.
- SAP Gateway. The SAP Gateway is used to facilitate the SAP Fiori applications.
- T4EA. T4EA is an integration software that provides connectors to establish a connection between the Teamcenter system and other enterprise applications through different technologies, such as file exchange (XML), web services, database connectors (JDBC and JMS), and messaging (API).
- T4S4. This is an out-of-the-box adapter to connect Teamcenter directly with SAP S/4HANA.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB

#bimodal_IT_layers_Communication_layer_CIP

In the second integration pattern, the Service layer connects the Stable IT platform with the Explorative IT platform. The communication is established through a Service Bus, such as SAP PO or T4EA, a Cloud Integration Platform, such as SCP or Azure Integration Services, or a microservice platform, such as Azure. Or even through a combination of these solutions.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

The third integration pattern is used within the Explorative IT platform to connect the digital systems and applications with each other. We are using Azure to create, change, manage, and integrate the microservices.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can recognize the presented models.

I would like to make two additional comments on your models. The first comment is related to the Systems of Innovation. As already explained, not all the Systems of Innovation are explorative. In your case, I would add a new Grouping into the Slow-Speed layer, referring to the Stabilized Systems of Innovation, and connect this Grouping with the Communication layer.

#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_through_Communication_layer

And the second comment is related to the descriptions of the layers. I would describe the layers as follow:

- Slow-Speed Architecture becomes "Stable IT layer";
- Communication layer becomes "Service layer";
- And Fast-Speed Architecture becomes "Explorative IT layer."

These descriptions (of the layers) will prevent confusion within an organization and are aligned with the latest concepts and approaches.

#bimodal_IT_layers_terminology

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The following systems are incorporated within the Stable IT platform:

- Systems of Records;
    - SAP S/4HANA;
        - Financial Management (Accounting, Controlling, Costing, and Treasury);
        - Order to Cash and After Sales (Sales Administration, Service Management, and Invoice Management);
        - Source to Pay (Procurement, Contract Management, SRM, Invoice Management, and Sourcing);
        - Manufacturing (Production Management, Order Management; and Quality Management);
        - HRM (Compensation Management, Recruitment, and Staffing);
    - Teamcenter;
        - PLM (Material Management, Product Configuration Management, and Quality Management);
    - Net@Pro;
        - MES (Production Planning, Manufacturing Execution, Maintenance Management, Warehouse Management, and Quality Management);
- Systems of Differentiation;
    - Dynamics 365 CRM;
        - Quote to Order (Sales Management, Loyalty Management; Event Management, Warranty Management, and Case Management);
    - Power BI;
        - Business Analytics;
        - Business Intelligence;
    - SAP Ariba;
        - SCM (MRP, Transport Management, Inventory Management, and Warehouse Management);
    - ServiceNow;
        - Tickets;
        - Projects Management;

        - IT Service Management;
    - Egencia Travel Management;
        - Travel Management;
        - Reservation Management;
        - Expense Management;
- System of Innovation;
    - Sensor APIs;
    - Egencia App;
        - Travel Management;
        - Expense Management;
    - SAP Fiori;
    - Internal APIs and related Web Apps.

#Slow-Speed_systems_and_software
#Slow-Speed_systems_and_software_functionalities
#Slow-Speed_stabilized_systems_of_innovation

The following systems are incorporated within the Explorative IT platform:

- Systems of Innovation;
    - Sitecore;
        - Customer Portal;
        - Employee Portal;
        - Partner Portal;
        - Supplier Portal;
    - SharePoint;
    - MS Power Apps;
    - Collaboration and Social Media Apps;
        - Teams;
        - WebEx;
        - Facebook;
        - Linked-In;
    - Augmented Reality;
        - AssistAR;
    - Analytics and Data Management tools (Power BI);
        - Dashboards;
        - Reports;
    - Other Web Apps;
    - Smart Device and Mobile Apps;
        - iOS Apps;
        - Android Apps;
    - Specific IoT solutions;
    - External APIs and related Apps;
- External Developer Platform;
    - Azure.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

Common characteristics of the Stable IT platform are:

- Stable;
- Robust;
- Reliable;
- Secure;
- The systems are supporting the business processes and holds business-critical objects and data;
- The systems do not change a lot once they are deployed into production;
- A maximum of errors is allowed (SLA);
- Solid test processes;
- Delivers operational value;
- The agile way of working is not always possible. Results in long release cycles;
- Fail-over strategy in place to guarantee business continuity and high availability;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance;
- High influenced by third parties.

#Slow-Speed_characteristics

Common characteristics of the Explorative IT platform are:

- Less stable;
- Features are small and independent;
- Flexible. The applications can change daily;
- Off-line capability is a necessity to ensure 24/7 availability and for a high experience;
- Rapid development and delivery of software; Often and frequent deployments;
- Focus on innovation, exploration, and experimentation;
- Less focus on Risk Management. Errors are already factored in;
- The creation of business value is a must. Preferable after each iteration;
- Planning changes constantly;
- Apps can contain business-critical data, but it is not a standard requirement;
- Only an Agile way of development is supported;
- Trend to have only a development and production environment.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

In our IT landscape, multiple middleware technologies are used to connect and integrate the various systems, software, applications, and solutions.

All systems and applications deployed on-premise and in the data center are connected through SAP PO and T4EA. Because the SAP landscape forms the IT landscape's backbone, I will only mention the SAP PO system components. But, remember that the T4EA service bus has almost identical components, features, and capabilities.

#deployment_model_on-premise
#deployment_model_data_center
#bimodal_IT_layers_Communication_layer_ESB

The (technical) components of SAP PO are:

- Integration Builder;
    - Enterprise Service Repository;
        - Creation and Discovery of Services;
    - Integration Directory;
        - Set-up of Mediation Flows;
- Integration Server;
    - Adapter Engine;
        - Transformations;
    - Integration Engine;
        - Orchestration;
        - Routing;
        - Publish and Subscriptions;
    - Business Process Engine;
        - Rules;
- System Landscape;
    - System Landscape/Tenant Manager;
- Configuration and Monitoring;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Mediation flow Manager;
    - Event Handler;
    - Adapter Manager;

- Access Controller;
  - Authorization;
- Exception handler;
- Logger;
- Auditor;

- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflow;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Message Tracking and Throttling;
  - Exception Management;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;

- HTTP and HTTPS;
- SOAP;
- REST;
- XML;
- FTP and SFTP;
- SMTP;
- Flat File;
- Database Adapter;
- 3rd Party Adapter;
- Custom Adapter.

#Communication_layer_ESB_capabilities

Because we have Ariba in the cloud and S4/HANA on-premise, we are using SCP to establish the communication.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of SCP are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;

- Connectivity Hub;
  - Repository for Adapter Discovery;
  - Adapter Builder;
    - Set-up of Adapters;
  - Adapter Configuration Manager;
  - Adapter Security Manager;
    - Set-up of Authentication;
  - Adapter Event Handler;
  - Workflow Builder;
    - Set-up of Workflows;
    - Editing of Workflows;
  - Access Controller;
    - Identity Access Manager;
    - Set-up of Authorization;
  - Logger;
  - Auditor;
- Integration Engine;
  - Adapter Engine;
  - Orchestration Engine;
  - Transformation Engine;
  - Routing Engine;
  - Rule Engine;
  - Publisher and Subscriber Engine;

- API Management Service;
  - Repository for API Discovery;
  - API Designer;
    - Set-up of APIs;
    - Edit APIs;
  - API Configuration Manager;
  - API Test Manager;
- BPM Service;
  - Business Process Manager;
    - Set-up of Business Rules;
    - BPEL;
  - Workflow Builder;
    - Set-up of Workflows;
    - Editing of Workflows;
  - Business Activity Monitor;
- Gateway Service;
  - Traffic Manager;
  - Exception Handler;

- Policy Manager;
- Version Manager;
- Call Handler;
- Logger;
- Auditor.

#Communication_layer_CIP_components

The SAP cloud integration platform realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Service Discovery;
  - Call handling;
  - Message Tracking and Throttling;
  - Rate Limits;
  - Traffic Management;
  - Exception Management;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;

- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - OData;
  - XML;
  - FTP and SFTP;
  - Flat File;
  - SMTP;
  - Database Adapter;
  - 3rd Party Adapters;
  - Custom Adapters.

#Communication_layer_CIP_capabilities

And last but not least, we are using Azure as a Container Orchestration Platform to connect the Explorative environment and to realize and manage the microservices.

#bimodal_IT_layers_Communication_layer_MSA

The (technical) components of this platform are:

- Container Orchestration Platform;
  - Container Orchestration Engine;
  - Scheduler;
  - Controllers;
  - API Server;
  - API Gateway;
  - Resource Manager;
  - Storage;
  - Queues;
  - Registry;
  - Logs.

#Communication_layer_MSA_components

The Azure environment realizes the following capabilities and functionalities:

- Container Orchestration Platform;
  - Container Orchestration;
  - Clustering;
  - Image Discovery;
  - Container Security;

- Load Balancing;
  - Scheduling;
  - Rollouts and Rollbacks;
  - Self-Healing;
  - Workflow Management;
- Microservices Mesh or an API Gateway (whereas an API Gateway is more preferred);
  - Service Discovery;
  - Messaging;
  - Service Routing;
  - Call handling;
  - Transformation;
  - Rate Limiting;
  - Traffic Management;
  - Throttling;
  - Caching;
  - Fail-over;
  - Microservice Security;
  - Configuration Management;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Container;
  - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

In the Explorative IT platform, the majority of the increments are delivered through CI/CD. In all other cases, the increments are delivered through Scrum.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum

In the Stable IT platform, we are using a mix of SDLC approaches to deliver or change a system. The SDLC approach is chosen based on the goal, complexity, and length of the IT project. For example, the most convenient SDLC approach for adopting or implementing a System of Record is the Waterfall methodology, while a System of Differentiation is implemented through Scrum.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_traditional_framework_waterfall
#methodology_agile_framework

#methodology_agile_framework_Scrum

As an IT organization, our goal is to pursue an agile environment. Therefore, we switch towards an agile approach, such as Scrum or DevOps after the go-live of a System of Record.

#bimodal_IT_layers_Slow-Speed
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

Due to various SDLC approaches between the two platforms, there is a constant need for alignment between the different teams. On one side, it helps us make better decisions and explain why we deliver an increment or system in a certain way. On the other side, misalignments or uncontrolled releases can jeopardize the whole operation. To avoid issues and enhance the alignment, we have a weekly session with all the integration teams to create awareness of the various changes and guide the releases.

#impact_continuous_alignment
#impact_release_management

9: Do you use different methodologies across the different layers?

Yes, the common SDLC approach for the Explorative IT platform is CI/CD. And for the Stable IT platform, the SDLC approach (traditional or agile) depends on the goal, complexity, and length of the related IT project.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is embedded through the often and early release of increments. As mentioned before, within the Explorative IT platform, most increments are delivered through CI/CD. The same applies for the increments within the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

The Explorative IT platform and the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

Within our organization, CI/CD is done through the use of Azure DevOps and Postman.

Azure DevOps provides the following functionalities:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, User Stories, and Work items;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. This functionality is used to automate the build, test, and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. We are using Azure Service Fabric and Azure Functions for the deployments;
- Azure Monitor and Grafana. These tools are used for monitoring and analysis.

Postman provides the following functionality:

- Newman. This functionality is used to automate and test the various API calls.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following conditions need to be met to realize and facilitate CI/CD:

- A CI/CD strategy and Recovery strategy needs to be in place;
- A proper branching structure should be in place;
- Increments need to be small. This approach also minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before every deployment;
- Configurations and passwords need to be easily accessible and always up-to-date;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

We have the following tools in place to support the development within the Stable IT and Explorative IT environments:

- A tool to manage the planning and requirements;
- A project and IT Service Management system;
- A ticketing system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Version Control systems/Repositories;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Automation server (build, test, publish and deploy automation);
- Software Quality Assurance tools. This also includes test tools;
- Release and Deployment tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

- ServiceNow for the creation of tickets and to support the project management and IT Service Management activities;
- Office 365 is used to capture and manage the requirements. This tooling is also used to create functional and technical specifications and documentation.
- SharePoint to store all documentation;
- We are using the following development kits:
    - SAP NetWeaver and SAP workbench for ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - Visual Studio for .NET, Python, C/C++, and Node.js developments;
    - Visual Studio extensions (PowerShell and YAML);
    - C/C++ for Teamcenter;
    - Eclipse for SCP and Teamcenter extensions;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Azure DevOps;
- We are using several Test Management Tools;
    - Micro Focus UFT one for SAP, Azure, and Java;

    - Test Manager for Teamcenter;
- For the deployment, we are using the following tools:
    - CHARM in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
    - And to deploy Teamcenter functionality, we are using Deployment Center from Siemens;
- Monitoring tools:
    - Azure Monitor (Application Insights and Azure Alerts);
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

From my perspective, there is no relationship identifiable between a deployment model and Bimodal Architecture. The Stable IT platform systems are mainly deployed on-premise or hosted in a data center because an organization wants to control these systems. The reason is that these systems are supporting core business processes and holds critical data. In contrast, the majority of the systems within the Explorative IT environment are deployed in the cloud.

#bimodal_IT_layers_Slow-Speed
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

The same trend is noticeable for the middleware. The ESB systems (SAP PO and T4EA) are mainly deployed on-premise or hosted in a data center. In contrast, the cloud integration platforms and container orchestration platforms are deployed in the cloud and offered as PaaS or iPaas.

#bimodal_IT_layers_Communication_layer_ESB
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#deployment_model_cloud
#deployment_model_cloud_Paas
#deployment_model_cloud_iPaas

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

There is no common deployment practice in place. It depends on the particular system and the organizations' IT strategy where and how a system is deployed. However, there is a noticeable trend in which the deployment models are moving towards the cloud. A private cloud could be an alternative for the systems within the Stable IT platform.

#deployment_model_movement_towards_cloud
#bimodal_IT_layers_Slow-Speed
#deployment_model_cloud_Private

15: What were the major milestones that lead to the current set-up of the IT landscape?

The current IT landscape is formed through several acquisitions and mergers, including their local IT.

#organization_IT_milestones
#organization_current_IT_landscape

16: What is the future roadmap of the IT landscape of your organization?

The roadmap is to simplify the current IT landscape. At the moment, we have almost 500 applications available in our IT landscape. Therefore, there is a need to consolidated these number of systems to a maximum of 60 systems. Also, we want to get rid of the point-to-point connectivity by using a Service layer.

#organization_IT_roadmap
#organization_IT_strategy

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I think that the focus will move from systems and applications to services. However, there will be a split between core services and fast-changing services. Therefore, Bimodal Architecture will stay a relevant concept.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

Bimodal Architecture requires a disciplinary team set-up, as indicated within the multiple agile approaches. However, the business itself should also change to commit to the way of working of DevOps.

#methodology_agile_framework_devops
#concept_bimodal_IT_agile_teams

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

Currently, the priority is set by the business for the several systems and applications within the Stable IT platform. Before we can build or adjusted a system, we need to provide a detailed impact analysis to determine the IT projects' risks and costs. Once a steering community has approved the estimations, the IT project can start with its development and delivery.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_estimations

In the Explorative IT platform, the priority is set by the PO. The costs of the associated agile teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt?

The technical debt is minimal in the Explorative IT platform. The reason is that the systems and applications are built in Java, .NET, C/C++, or Node.js and invoke a subset of services. However, if there are any issues or problems, they can be fixed very fast, mostly in another sprint.

In contrast, the Stable IT platforms' systems and applications have a high technical debt because they contain custom build functionalities. Also, there is an urgent need to expose the various functionalities as services to realize and facilitate the API platform.

#concept_bimodal_IT_technical_debt

Candidate 10 – PM-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I started in 2005 at my current employer. In 2012, I became one of the Project Managers {PM-01} within the technical and architectural domain. My projects focus on integration, mobile, smart and intuitive applications, cloud solutions, cloud adoption, infrastructure, and identity management.

Before I joined my current employer, I was working for a big consulting firm. During this period, I have fulfilled multiple roles at various clients.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

As you maybe already know, our organization is the biggest service provider in mail and logistics within the Benelux {Postal-01}.

#company_profile

The internal IT organization consists of a mixture of internal and external IT resources. We have around 380 internal IT resources and almost 120 external IT resources. Also, we are exploiting nearly 200 IT systems within the IT landscape. This explanation gives a better view of how complex and diverse our IT landscape is.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal Architecture is a concept initiated by Gartner. The concept distinguishes the IT architecture into two modes. Mode 1 realizes standardization, operational excellence, and efficiency. This mode ensures that the Operational Backbone of the organization keeps running. Mode 2 realizes the speed of change. This mode supports the transformation towards digitalization and establishes a Digital Platform.

#bimodal_IT_layers_Slow-Speed
#concept_digital_transformation_operational_backbone

#Slow-Speed_characteristics
#bimodal_IT_layers_Fast-Speed
#concept_digital_transformation_digital_platform

Our IT landscape also comprises different IT layers and follows the principles of the Bimodal IT concept of Gartner. However, we are distinguishing three layers instead of two. These layers are a Stable layer, an Explorative layer, and a Communication layer.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Fast-Speed

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

As already indicated in the previous question, the Bimodal environment comprises a Stable layer, an Explorative layer, and a Communication layer. If we look at a more detailed level, we will find the following components in each layer.

The Stable layer comprises:

- Systems and software to standardize and optimize business processes, increase efficiency, and improve and enhance operational excellence;
- Solutions, systems, and "digital" applications to extent specific business processes, enhancement of internal digitalization, and for leveraging operational value and efficiency;
- Multiple integration patterns to connect and integrate the systems and applications within and outside the layer by exposing the functionalities as services. The integration patterns include adapters, EDIFACT standards, proxies, APIs, web services, and microservices.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

The Explorative layer comprises:

- Web and mobile solutions to enhance experience, interaction, and engagement.
- Systems and applications to experiment and explore new technologies;
- Multiple integration patterns. These patterns are used to connect the digital systems and applications with each other through services.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_integration_patterns_Fast-Speed

The Communication layer comprises:

- A service-oriented environment, such as an ESB, a cloud integration platform, or a microservice environment. An organization can use a combination of service-based environments to establish and leverage the integration patterns. These environments facilitate loose coupling, interoperability, and modularity;
- Middleware systems and tools to create, adjust, monitor, and manage the services (web services and microservices) and APIs.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

This layer is responsible for realizing and facilitating an API platform by exposing the core systems' functionalities, objects, and data as services. Subsequently, the services can be used in mobile, smart, intuitive, and digital solutions.

#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

It is also important to note that the APIs can be distinguished between internal as well external APIs.

#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can recognize the models. The three layers in your model are similar to the layers in our IT landscape.

However, your Stable layer is missing some components. There are APIs and applications that cannot be embedded in the Explorative layer. Neither can they be marked as core systems. These APIs and applications are stable and used internally. These digital solutions help the business create operational value by providing extra information, increasing responsiveness, and delivering operational efficiency. A few examples are:

- The internal Track and Trace API;
- Particular applications and APIs that are supporting the sorting process;
- API for returns.

Also, note that these applications and APIs do not expose any objects or data towards the Explorative layers' systems and applications.

#Slow-Speed_stabilized_systems_of_innovation

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The following IT systems are embedded in the Stable layer:

- SAP ECC;
- SAP Simple Finance;
- SAP SuccessFactors;
- SAP Hybris Billing;
- Salesforce;
- SAP BW (ETL, report and dashboard builder);
- Extensions. Activities, tasks, workflows to leverage the existing business processes;
- Sorting application;
- Internal APIs. For example, the Track and Trace functionality and the Return functionality.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The following IT systems are embedded in the Explorative layer:

- Portals;
    - Customer;
    - Employee;
    - Partner;
    - Supplier;

- Website;
- Webshop;
- Mail and Parcel App;
- Track and Trace App (consumers and business);
- Distribution App;
- Dashboards;
- Reports;
- Web Apps. Developed in a day or two via Java, Node.js, Python, and invokes one or more services;
- Mobile and smart device Apps;
    - iOS;
    - Android;
- Other external APIs (Business services).

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of the Stable layer are:

- Stable, robust, and reliable;
- Security is essential and an ongoing topic;
- The IT systems support the business processes. It holds and manages the business-critical objects and data;
- The IT systems do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Supports a traditional and an agile way of software delivery;
- Fail-over strategy in place to guarantee business continuity;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The main characteristics of the Explorative layer are:

- Features are small and independent;
- Delivers business value at each release or iteration and ends at a given moment;
- Comprises an off-line capability to guarantee 24/7 availability;
- An IT system can contain business-critical data, but this is not standard;
- The IT systems are easily replaceable;
- Less focus on Risk Management. Errors are already factored in;
- Focus on experimentation and exploration;
- Only an Agile way of development is supported;

- Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Communication layer comprises one or more service-based environments. This can be a service bus, like Biztalk, SAP PO, IBM WebSphere, or any other service bus. A Cloud Integration Platform, like SAP Cloud Platform (SCP) or Azure Integration Services. A microservice environment and related Container Orchestration Platform, like Azure, Kubernetes, or AWS. Or the layer can realize the API platform by combining two or more service-based environments.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA

Since my focus is on developing and delivering systems and applications within the Explorative layer, I will take the Microsoft service-based products as an example to answer your question.

The main components of Biztalk are:

- Message Builder;
    - Set-up of Adapters;
    - Creation of Services;
    - Set-up of Mediation Flows;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
    - Configuration Manager;

  - Web Service Manager;
    - Authentication;
    - Message Security;
    - Encryption;
  - Access Controller;
    - Authorization;
  - Exception handler;
  - Event Handler;
  - Mediation flow Manager;
  - System Landscape/Tenant Manager;
  - Auditor;
  - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflow;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

Biztalk realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Message Tracking and Throttling;
  - Exception Management;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;

        - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

The main components of Azure Integration Services, are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Mediation Flows;
        - Integration/Message Builder;
            - Set-up of Mediation Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;

- Connectivity Hub;
    - Repository for Adapter Discovery;
    - Adapter Builder;
        - Set-up of Adapters;
    - Adapter Configuration Manager;
    - Adapter Security Manager;
        - Set-up of Authentication;
    - Adapter Event Handler;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Access Controller;
        - Identity Access Manager;
        - Set-up of Authorization;
    - Logger;
    - Auditor;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;

- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;

  - Policy Manager;
  - Version Manager;
  - Call Handler;
  - Logger;
  - Auditor.

#Communication_layer_CIP_components

This platform realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Service Discovery;
  - Call handling;
  - Message Tracking and Throttling;
  - Rate Limits;
  - Traffic Management;
  - Exception Management;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;

- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

The best practice to run a microservice is through the use of container technology instead of VMWare. Therefore, I will only mention the components of the Azure container orchestration platform.

#bimodal_IT_layers_Communication_layer_MSA

The (technical) components of Azure are:

- Nodes;
    - Pods;
- Container Orchestration engine;
- Scheduler;
- Controllers;
- Key-value Database;
- API server;
    - API Manager;
    - API Gateway;
- Resource Manager.

Other components in the infrastructure are:

- Storage;
- Queues;
- Container registry;
- Container repository (optional);
- Network;
- Provisioning;
- Auditor;

- Logger;
- Infrastructure Security Manager.

#Communication_layer_MSA_components

Azure realizes the following capabilities and functionalities:

- Container Orchestration Platform;
  - Container Orchestration;
  - Container clustering;
  - Image Discovery;
  - Container Security;
  - Network settings;
  - Key Vault;
  - Load Balancing;
  - Scheduling;
  - Rollouts and Rollbacks;
  - Self-Healing;
  - Workflow Management;
- API Management;
  - Service Discovery;
  - Messaging;
  - Routing (of services);
  - Call handling;
  - Transformation;
  - Rate Limits;
  - Traffic Management;
  - Throttling;
  - Caching;
  - Circuit Breaker;
  - Fail-over;
  - Microservice Security;
  - Configuration Management;
  - Networking/IP;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Container;
  - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

I must acknowledge that we are using agile as well as traditional approaches in our IT landscape. Especially in the Stable layer, the traditional approach is a common practice.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

From an organizational perspective, we have adopted SAFe to realize organizational agility. This means that most IT projects and the delivery of the IT systems are performed in an agile way. Scrum is the most preferred method. However, in some cases, like tenders, big and complex migrations, governmental changes, or implementing a new core system, we are obligated to follow a project management (PRINCE2 or PMP) or traditional approach (Waterfall or V-Shape). Once the system is delivered into production, we can switch to an agile approach to exploit the concerned system.

#methodology_agile_framework_SAFe
#methodology_agile_framework_Scrum
#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

CI/CD is a famous technique within the Explorative layer to realize and facilitate systems and applications.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

I believe that the biggest concern is related to release management. When a new requirement only applies a small change within an App, then the chance is minimal that the adjustment will impact the corresponding API(s) or the underlying core system(s). In this case, we can just adjust the App or develop a new App to replace the old one. However, when a requirement indicates that we need to leverage and store new data through the API, then the chance exists that we need to identify and enhance all the interconnected systems that are related to that specific API. In this case, we need to ensure that all the related increments are in sync and released at the same moment towards another environment to guarantee the proper working of the functionality.

#impact_release_management
#impact_continuous_alignment

9: Do you use different methodologies across the different layers?

Yes. For the Explorative and Communication layers applies that the services and increments are delivered through CI/CD or via another agile approach, like Scrum.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

Also, I like to mention that we are distinguishing how web services and microservices are delivered. Web services are mainly delivered through an agile approach, while microservices are delivered through CI/CD.

#bimodal_IT_layers_Communication_layer_ESB_services
#methodology_agile_framework
#bimodal_IT_layers_Communication_layer_MSA_services
#methodology_agile_framework_continuous_integration_and_delivery

For the Stable layer applies that most of the new core systems are initially delivered through a traditional approach. Once the systems are in production, the exploitation is performed via an agile approach. For the internal APIs and applications applies that they are delivered and exploited via an agile approach.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#Slow-Speed_stabilized_systems_of_innovation
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is foremost used to deliver the systems and applications within the Explorative layer. This technique is also used to deliver the microservices.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

However, I expect that CI/CD will be adopted in every layer in the near future because the need for fast delivery is increasing. Also, the prominent and established vendors have noticed this. For example, SAP is offering CI/CD for Fiori developments in SAP S/4HANA.

#bimodal_IT_layers_Slow-Speed
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

Currently, only the Explorative layer and the microservices environment are affected by CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The development of an increment starts locally. Once a developer is done coding, the source code needs to be submitted into a Repository. After this, the AWS platform comes into play.

#continuous_integration_and_delivery_components
#continuous_integration_and_delivery_systems_and_software

The components of the AWS tools are:

- AWS CodeCommit. AWS CodeCommit is used as a Version Control System;
- AWS CodeArtifact. AWS CodeArtifact is a fully managed artifact repository service to store, publish, and share software packages securely;
- AWS CodePipeline. AWS CodePipeline is a fully managed continuous delivery service that automates the release pipeline for fast and reliable applications and infrastructure updates. However, this functionality can also be integrated with Jenkins and other third-party automation servers. The AWS CodePipeline functionality is used to automate the build, test, and publish activities;
- AWS CodeDeploy. The release functionality is used to automate the deployment step;
- Amazon CloudWatch. This tool is used for monitoring and analysis.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

To comply with CI/CD, the following practices needs to be in place:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;

- A verification step should be in place before each deployment;
- Use of a Version Control System needs to be encouraged;
- Use of an Automation Server that can automatically build, test, publish and deploy the various increments;
- Provide KPIs to measure the pipeline executions;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

The following tools and components are used for the development of systems. The tools are applicable in all layers:

- A tool to manage the planning of the various artifacts;
- A tool to manage the requirements of the various increments;
- A tool for project management or IT Service Management;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Several Development Kits (DKs) and Integrated Development Environments (IDEs) to develop, change, and configure systems, programs, features, and APIs;
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The tools and components are realized through the following systems:

- ServiceNow is used to manage the planning, backlog, and sprints. This system is also used for ticket management;
- Tracecloud is used for requirements management;
- ARIS and Sparx are used to design, model, and optimize business processes. These tools also provide comprehensive analysis and design functionalities for UML, SysML, BPMN, and many other technologies;
- Workday and Rallyteam are used to encourage collaboration and to optimize the workforces;
- SharePoint is used to store all the documentation;
- For the development and configuration of increments, systems, and software, the following tools are used:
    - SAP Netweaver and Workbench are used to support the ABAP developments;
    - SAP Implementation Guide to configure the SAP system;

Universiteit Leiden

    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Lightning platform and Heroku to support the Salesforce developments;
    - Copado to enable DevOps for Salesforce;
- We are using the following Software Quality Assurance tools:
    - Copado Test;
    - AWS CodePipeline. For automated unit tests, deployment tests, and End-To-End tests;
- We are using the following tools for deployment:
    - AWS CodeDeploy. To automatically deploy increments from the CI/CD pipeline;
    - Copado Release is used to deploy increment in Salesforce;
    - CHARM is used in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for SAP iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit.
- We are using the following monitoring tools:
    - Amazon CloudWatch;
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

No. There is no quantifiable relationship between the deployment models and Bimodal Architecture. Some systems of the Stable layer are deployed on-premise, like the systems to support the sorting processes. Other systems are subscription-based, like Salesforce and SAP Hybris Billing. While SAP ECC, associated solutions, other core systems, and the internal applications are deployed in the cloud. For the systems and applications within the Explorative layer applies that they are deployed in the cloud.

#bimodal_IT_layers_Slow-Speed
#deployment_model_on-premise
#deployment_model_cloud_SaaS
#deployment_model_cloud
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

Interviewer: Thank you for the answer. Can you also elaborate a little bit on the technical infrastructure of the mentioned deployment models?

For the on-premise and the data center environment applies that the systems are provided via a Server Farm. This Server Farm comprises Application Servers, Database Servers, File Servers, Web servers, etc.

Via the Server Farm, the following services can be used:

- Load Balancing;
- Data Access;
- File Access;
- Data Caching;
- Web Caching;
- And Process Scheduling.

These services can be reached and executed via a private network that is established through a VPN or the usage of a Proxy Server.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

For the cloud environment applies that the physical servers, network devices, and other hardware, including Application Servers, Database Servers, File Servers, Exchange Servers, Web Servers, etc., are located at a Cloud Hosting Provider. All the components are virtualized through containerization technology and offered via images as virtualized hardware services. Consequently, the virtualized hardware services are offered as an event-driven serverless compute platform to an organization or user.

The Cloud Hosting Provider can offer the following deployment models:

- Public model;
- Private model;
- or a Hybrid model.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

Universiteit Leiden

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

Cloud is becoming the standard for deploying systems, applications, and solutions.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

The organization adopted a cloud-only strategy in 2013. The transition was completed in 2018.

#organization_IT_strategy
#organization_current_IT_landscape

Note that not all the IT systems are deployed in the cloud.

#notes_not_all_systems_can_be_deployed_in_the_cloud

16: What is the future roadmap of the IT landscape of your organization?

Consolidation of the cloud environments. Also, we want to move as an organization towards a data-driven organization and enforce business model-driven IT.

#organization_IT_roadmap
#organization_IT_strategy

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I believe that Bimodal Architecture will stay as a concept because there is no other suitable alternative.

The three layers are currently supporting us with the digital transformation and provide us the tools and components to achieve agility within our IT landscape.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

I also believe that the Stable layer will never disappear. Even if we do not have any core systems anymore, we will still need stable databases, monitoring and analyzing tools (back-end functionalities), etc.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software_functionalities

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt?

The respondent did not answer the question due to time.

Candidate 11 – EA-05

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

It is already twenty-three years that I'm working within the IT domain. During this period, I have fulfilled several roles and worked for several companies. I joined my current employer in 2011, and since 2015 I am fulfilling the role of an Enterprise Architect {EA-05}.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The organization is the biggest service provider in mail, parcels, and logistics within the Benelux {Postal-01}. The internal IT organization counts around 500 members. In this count also the external IT resources are included. The IT landscape itself is complex and diverse. Therefore, we have almost 20 architects walking around to cover the different domains.

#company_profile
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Our IT landscape is built up of different layers. Therefore, the architecture can also be referred to as a "Layered Architecture." The IT landscape's essence is to decouple the front-end and the back-end by providing internal and external APIs. This type of architecture is similar to the Bimodal IT concept of Gartner. The reason for choosing this particular architecture style is because we want to facilitate and utilize a steady Operational Backbone on one side and support flexible IT through a Digital Platform on the other side.

#bimodal_IT_layers_Slow-Speed
#concept_digital_transformation_operational_backbone
#bimodal_IT_layers_Fast-Speed
#concept_digital_transformation_digital_platform
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

The IT landscape is divided into three layers. These layers are:

- A Stable core layer;
- A Process layer to enable services (APIs);
- And an Experience layer.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Fast-Speed

Various APIs and services are exposed between the layers, which can be consumed by the different IT systems.

#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The main components of a Bimodal environment are:

- Stable layer. This layer comprises the core information systems and digital solutions to realize and utilize operational excellence. The associated systems support critical business processes and store and manage the objects and data. When these systems are compromised, it can jeopardize the business continuity;
- Experience layer. This layer focuses on experimentation, exploration, and innovation. The environment is continuously developing and providing digital products and services to the end-customers and other relevant stakeholders.
- Process layer. This layer is crucial because it is used to realize and exploit an API platform. Via the usage of Business process management tools, message and event brokers, cloud integration platforms, and a microservice environment, the services can be created, transformed, routed, and consumed by the different IT systems.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#Slow-Speed_characteristics
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#Fast-Speed_characteristics
#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_CIP_services

#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

Interviewer: This is an interesting answer. Can you elaborate a bit more on the three layers and the components you expect in each layer? Also, can you explain the difference between internal and external APIs?

Yes, I can. The Experience layer comprises systems and applications that increase the experiences of the stakeholders via different channels. The environment is also used for the exploration of new services, products, and technologies. Typical IT systems and applications of this layer are:

- Portals;
- Web Apps;
- Webshop;
- Mobile Apps;
- And multiple smart device Apps.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

The Stable layer comprises systems like Salesforce, SAP ECC, SuccessFactors, and a BI/BW system. When we look a bit closer, for example, at Salesforce, you will notice that Salesforce can support the track and trace capability. However, the track and trace functionality can be used in three different channels. The first channel could be for consumers. The customers only want to know the status of their parcel. The second channel is related to businesses. A business can be interested in the status of all the parcels that they have send or received (for the return process) within a specific period. And the last channel, which is for internal use, is required for the customer contact agent because they need additional info about the parcel(s) and the related driver(s).

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

To meet all the different requirements, we can create one generic API that comprises all the attributes and values. However, we can also create separate APIs to support each channel individually (customer, business, and internal users). This makes the API platform more complex and forces us to rethink which APIs are internal and should belong to the Stable environments' IT systems. And which APIs are external and, therefore, are facilitating the IT systems within the Experience environment.

#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can link the presented models to our IT landscape. As already indicated, the main components of a Bimodal environment are the three layers.

However, there is a difference between your presented models and my view on Bimodal Architecture. I miss the internal APIs within the Stable layer, the corresponding IT systems, and the related integration patterns. As explained with the track and trace example, some APIs are used internally to support the business operations and facilitate critical information.

#Slow-Speed_stabilized_systems_of_innovation

Also, I would not call the layers Fast-Speed and Slow-Speed because the terminology can cause many problems when explaining the layered architecture to the stakeholders.

#bimodal_IT_layers_terminology

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The Stable layer comprises the following IT systems:

- SAP ECC;
- SAP SuccessFactors;
- SAP Hybris Billing;
- Salesforce;
- SAP BW (ETL, report and dashboard builder);
- Sorting application;
- Internal APIs.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The Experience layer comprises the following IT systems:

- Portals;
    - Customer;

  - Employee;
  - Partner;
  - Supplier;
- Webshop;
- Web Apps;
  - Track and Trace App (consumers and business);
  - Mail and Parcel App;
  - Distribution App;
- Dashboards;
- Reports;
- Mobile and smart device Apps;
  - iOS Apps;
  - Android Apps;
- Other external APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of the Stable layer are:

- Stable, robust, reliable, and secure;
- The IT systems support the business processes. It holds and manages the business-critical objects and data;
- The IT systems do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Supports a traditional and an agile way of software delivery;
- Solid test processes;
- Fail-over strategy in place to guarantee business continuity;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance;
- Problem-solution design.

#Slow-Speed_characteristics

The main characteristics of the Experience layer are:

- Features are small and independent;
- Delivers business value at each release or iteration;
- Off-line capability is a necessity. It ensures a high experience and 24/7 availability;
- An IT system can contain business-critical data, but this is not standard;
- The IT systems are easily replaceable;
- Less focus on Risk Management. Errors are already factored in;
- The solutions are mainly centered around experience and exploration;

- Only an Agile way of development is supported;
- Rapid delivery oriented. Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

As already indicated, the Process layer comprises one or more middleware systems. This can be a service bus, like SAP PO, IBM WebSphere, BizTalk, or another service bus. A Cloud Integration Platform, like SAP Cloud Platform (SCP) or Azure Integration Services. A microservice environment and related Container Orchestration Platform, like Azure, Kubernetes, or AWS. Or the layer can realize the API platform by combining two or more service-based environments.

#bimodal_IT_layers_Communication_layer
#Communication_layer_systems_and_software
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA

The components of a service bus are:

- Message Builder;
    - Set-up of Adapters;
    - Set-up of Mediation Flows;
- Integration Engine;
    - Adapter Engine;
    - Orchestration Engine;
    - Transformation Engine;
    - Routing Engine;
    - Rule Engine;
    - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;

        - Authorization;
    - Exception handler;
    - System Landscape/Tenant Manager;
    - Logs;
- Business Activity Monitoring;
- Business Process Manager;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Conversion;
    - Caching;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Message Security;
- Transportation.

#Communication_layer_ESB_capabilities

The main components of a cloud integration platform, for example, Azure Integration Services, are:

- Cloud Integration Service;
    - Message Hub;
        - Repository for Service Discovery and pre-build Mediation Flows;
        - Message Builder;
            - Set-up of Mediation Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
            - Set-up of Authorization;
        - Message Queues;
        - Exception handler;
        - Logs;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;

    - Workflow Builder;
        - Set-up of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logs.

#Communication_layer_CIP_components

A cloud integration platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Configuration Management;
    - Deployment;
    - Service Registry;
    - Service Discovery;
    - Load Balancing;
    - Call handling;
    - Message Tracking and Throttling;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Conversion;
    - Caching;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;

    - Message Security;
- Transportation.

#Communication_layer_CIP_capabilities

In a microservice environment, the microservices are mostly virtualized through container technology instead of VMWare. Therefore, I will only mention the components of a container orchestration platform, like Azure.

#bimodal_IT_layers_Communication_layer_MSA

The container orchestration platform comprises the following (technical) components:

- Container Orchestration engine;
- Scheduler;
- Controllers;
- API Gateway;
- Resource Manager;
- Storage;
- Queues;
- Container repository;
- Logs.

#Communication_layer_MSA_components

The container orchestration platform realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Clustering;
    - Image Discovery;
    - Load Balancing;
    - Scheduling;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Dynamic Routing;
    - Call handling;
    - Transformation;
    - Traffic Management;
    - Throttling;
    - Caching;

- Microservice Security;
- Configuration Management;
- Analytics;
- API Version Management;
- Monitoring and Logging;
- Policies and Contracts;
- Container;
  - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

Each layer uses its own SDLC methodology. From an organizational perspective, we are using SAFe. This methodology helps the management to implement organizational agility.

#methodology_agile_framework_SAFe

In the Experience and Process layers, we use CI/CD and other agile approaches, like Scrum, to deliver the services (APIs), increments, and IT systems.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum
#methodology_agile_framework

In the Stable layer, we use project management and traditional approaches, like PRINCE2, PMP, and Waterfall, to deliver a new IT system. Once the IT system is delivered, the systems' enhancements and exploitation are performed using an agile approach.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The first impact is related to prioritization. When we need to adjust an API, all the related IT systems need to be identified and investigated. After this, the right priority needs to be determined and set for each adjustment. This will ensure that the functionality is aligned with all the associated IT systems and avoids "integration" issues. If this is not done thoughtfully, the user and functional requirements could be compromised.

#impact_prioritization

#impact_continuous_alignment

The second impact is related to release management. Most of the APIs, which are exposed from the Stable layer, are used to facilitate the IT systems within the Experience layer. When adjustments are required in the APIs, the chance exists that the corresponding IT systems also need to change. This means that all the related increments need to be released simultaneously to guarantee a proper working of the functionality.

#impact_release_management

9: Do you use different methodologies across the different layers?

Yes. For the Experience layer and Process layer, especially the microservice environment, applies that the services and increments are delivered through CI/CD or via another agile approach.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

Also, a distinction needs to be made about how the services (APIs) are delivered. Webservices are delivered through an agile method. In contrast, microservices are delivered through CI/CD.

#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#methodology_agile_framework
#bimodal_IT_layers_Communication_layer_MSA_services
#methodology_agile_framework_continuous_integration_and_delivery

In the Stable layer case, the IT systems and corresponding features are delivered through a mix of SDLC approaches, ranging from traditional approaches to agile approaches.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is used to realize and facilitate the IT systems within the Experience layer. Also, is this particular technique used to create and manage the microservices.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

As already indicated, CI/CD is only applicable in the Experience layer and the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The development of increments starts locally through the usage of several IDEs. Once the coding is finished, the source code is submitted into a Repository.

#continuous_integration_and_delivery_components

After this, the AWS platform comes into play. The components of the AWS tools are:

- AWS CodeCommit. AWS CodeCommit is used as a Version Control System.
- AWS CodePipeline. AWS CodePipeline is a fully managed continuous delivery service that automates the release pipeline for fast and reliable applications and infrastructure updates. However, this functionality can also be integrated with Jenkins and other third-party automation servers. The AWS CodePipeline functionality is used to automate the build, test, and publish activities;
- AWS CodeDeploy. The release functionality is used to automate the deployment step.
- Amazon CloudWatch. This tool is used for monitoring and analysis.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

For the enablement of CI/CD, the following practices are essential:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before every deployment;
- It needs to be ensured that configurations and passwords are easily accessible and always up-to-date;

- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increments.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

The following tools and components are used to support the development of increment, features, systems, applications, and systems. It also does not deviate between the layers:

- A tool to perform Project Management and IT Service Management activities. This system can also support tickets, planning, and requirements management.
- A Document Management System;
- Several Development Kits to develop, change, and configure systems, programs, features, and APIs;
- Test Management tools;
- Release and deployment tools;
- And monitoring tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The components are realized through the following systems:

- ServiceNow is used for IT Service Management activities and the creation of tickets;
- SharePoint is used to store all the documentation;
- For the development and configuration of increments, systems, and software, the following tools are used:
    - SAP NetWeaver and SAP Workbench to support ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Lightning platform and Heroku;
    - Copado to enable DevOps for Salesforce;
- We are using the following Test Management Tools;
    - Copado Test;

- AWS CodePipeline. For automated Unit Tests, Deploy Tests and End-To-End tests;
- We are using the following tools for deployment:
    - AWS CodeDeploy. To automatically deploy increments from the CI/CD pipeline;
    - Copado Release is used to deploy increment in Salesforce;
    - CHARM is used in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for SAP iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
- We are using the following tools for monitoring:
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

There is no relationship. It should also not be determined by the Bimodal environment.

The common deployment model of an IT system within the Stable layer is in an on-premise or data center environment. However, as an organization, we have chosen for a cloud-only strategy. This focus means that we try to deploy all the IT systems in the cloud. Also, note that this does not mean that we do not have any IT systems running on-premise because we do. But we try to keep this at a minimum.

#bimodal_IT_layers_Slow-Speed
#deployment_model_on-premise
#deployment_model_data_center
#organization_IT_strategy
#deployment_model_cloud
#notes_not_all_systems_can_be_deployed_in_the_cloud

For the systems within the Experience layer applies that they are always deployed in the cloud.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

Interviewer: Thank you for the answer. Can you also elaborate on the mentioned deployment models? What are the building blocks of the deployment models, and how are the models realized or facilitated?

For the cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containerization technology, like Azure Functions or Azure Storage Account, and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- or Hybrid models.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

In an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can be reached and executed via a private network established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

Note, due to recent technological developments, the deployment strategies of organizations have been changed. Nowadays, the majority of organizations are choosing to host their systems and software in the cloud.

#deployment_model_movement_towards_cloud

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

Cloud is becoming the common deployment practice for hosting systems and applications.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

We started in 2014 a transition towards the cloud. However, not all the IT systems are in the cloud yet.

And the second milestone was the adoption of an API platform (service layer) in 2018.

#organization_IT_strategy
#organization_current_IT_landscape
#organization_IT_milestones

16: What is the future roadmap of the IT landscape of your organization?

Consolidation of the cloud environments. Also, we want to move as an organization towards a data-driven business and enforce business model driven IT.

#organization_IT_roadmap
#organization_IT_strategy

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

I foresee a bright future for the Bimodal Architecture environment because there is no other suitable alternative. One model is not an option. And Multi-model will only add more complexity to the IT landscape.

Also, I believe that adopting the three layers will help in the digital transformation and help us adopt agility within our IT landscape.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt?

The respondent did not answer the question due to time.

Candidate 12 – IA-01

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I started my career in 2003. I started as an analyst within SAP R/3. Over the years, I have worked with almost every module of the SAP business suite, executed various SAP implementations, and conducted data migrations. As of 2014, I am in the lead of SAP integrations with other IT systems within the companies' IT landscape. Nowadays, I am fulfilling the role of an Integration Architect {IA-01} at my current employer.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The organization itself is a global player in the coffee and tea market {Products-01}. It offers an extensive range of high-quality and innovative coffee and tea products and solutions in more than 100 countries.

#company_profile

The internal IT organization consists of approximately 700 IT resources. Most IT resources are primarily intern, although 15 percent of the IT employees are hired via third parties to develop and maintain specific IT systems.

We have a centralized IT organization and IT landscape. The associated IT resources are responsible for all the IT systems within the IT landscape. To support our local units and brands, we have a small local IT team available within each country. The member count differs within each local IT team and depends on the size of the specific unit.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I see Bimodal Architecture as a cohesive architecture that covers multiple IT layers. The layers that are included in this architecture are:

- A stable IT platform;
- A Connectivity layer;
- And an explorative IT platform.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Fast-Speed

Each layer has its own purpose and fulfills certain goals within the organizational strategy and product portfolio. The stable IT platform holds the core systems and operational value-adding systems. This layer is authoritative (CRUD), in the lead of the data, a single source of truth, and is responsible for an organizations' process execution and operations. The Connectivity layer comprises the APIs. This layer is accountable for modularity, scalability, and reusability. The Explorative layer consists of IT systems and applications that support the digital offerings, consume and contribute to the data (read-update), and meet the new experience standards of the various stakeholders.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The stable, connectivity, and Explorative layers are the main components of the Bimodal Architecture environment.

Within these layers, the following components are embedded:

Stable layer:

- Core systems;
    - Business software;
        - CRM system;
        - SCM system;
        - HRM system;
        - Finance system;
        - Sales system;
        - MES system;
        - CLM system;
        - PLM system;
        - Field Service Management System;
        - Knowledge base system;

    - Data Management software;
        - BI system;
    - Project and Issue software;
        - Project Management system;
        - Ticketing system;
    - Banking software;
        - Transaction Management system;
    - Industry-Specific software;
        - Calibration system;
        - CAD system;
    - Interaction and communication software;
        - Call center system;
        - Reservation system;
        - Email system;
- Operational value-adding systems;
    - Business Apps;
        - Field Service App;
        - Account App;
        - Product App;
    - Data Management Apps;
        - Report Builder;
    - Industry-Specific Apps;
        - Asset Information App;
        - CAD App;
        - Internal Track and Trace API;
    - Interaction, communication and collaboration Apps;
        - Email App;
        - Calendar App;
        - Travel and booking App;
    - Helpdesk Apps;
        - Chatbot;
        - FAQ;
    - IoT Apps;
        - Sensor APIs;
        - Smart coffee machine App.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The core systems and operational value-adding systems are connected with each other. To enable the integration, the following components are used within the Stable layer.

- Adapters;
- RFC connections;
- EDIFACT standards (IDocs);
- Proxies;

- APIs;
- Service-based environment(s);
    - ESB;
        - Web services;
    - Cloud integration solutions;
    - Microservice Architecture and the associated container orchestration platform;
        - Microservices.

The adapters, EDI, and the RFC connections are used to connect the various core systems, like SAP CRM and SAP S/4HANA.

Proxies are created within core systems. Proxies are used to facilitate communication with other systems by generating and triggering services.

SAP also has out-of-the-box services, called Enterprise Services. These services can also be used to communicate with other systems.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_integration_patterns_Slow-Speed

Note that the Service-based environments are part of the Connectivity layer. These environments are also used to set-up and facilitate the connection between the Stable and Explorative layers.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

Explorative layer:

- Portal;
    - Consumer;
    - Business;
    - Employee;
    - Supplier;
    - Partner;
- Webshop;
    - Consumer;
    - Business;
- Business Apps;
    - Track and trace App;
    - Employee self-service App;
- Data Management App;
    - Dashboards;

  - Reports;
- Industry-Specific App;
  - Route App containing the locations of the leased coffee bars and rented machines;
- Helpdesk App;
  - FAQ;
  - Ticketing App;
- Document Management system;
- Collaboration and Social Media Apps;
- Mobile Apps;
  - iOS applications;
  - Android applications.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

Note that not all the mentioned IT systems and applications within this layer are stand-alone. To establish the connectivity and integration, a service-based environment is used.

#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can recognize the models. However, I like to indicate some points.

First of all, I like to indicate that you miss some components. These missing components are:

- The operational value-adding systems. This group of IT systems could also be called "Stabilized Systems of Innovation";
- The integration patterns between the core systems and the operational value-adding systems are missing.

#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_through_Communication_layer

Subsequently, the mature apps' linkage needs to flow towards the Grouping of the operation value-adding systems instead of the core systems.

#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed

Finally, I like to indicate that the layers' description can cause a lot of confusion. Gartner has developed the Pace-layered Application Strategy after the launch of the Bimodal IT concept. This strategy can be used as a methodology to categorize, select, manage and govern IT systems. In this strategy, Gartner divides the IT systems into three layers, namely:

- Systems of Record;
- Systems of Differentiation;
- And Systems of Innovation.

#concept_Pace-layerd_Application_strategy

From my perspective, the Systems of Record and Systems of Differentiation are part of the Stable layer. The Systems of Innovation can be embedded within the Stable or Explorative layer, depending on their purpose and use.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The following IT systems are embedded in the Stable layer:

- SAP S/4 HANA;
  - Fiori;
- Salesforce;
- Apttus;
- ClickSoftware;
- Power BI;
  - ETL;
  - Report builder;
- Operational value-adding web Apps;
- Internal APIs.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

Also, I would add the following technologies within the Stable layer:

- Blockchain;
- Machine Learning;
- And some IoT solutions, devices, and applications.

#Slow-Speed_stabilized_systems_of_innovation

The following IT systems are embedded within the Explorative environment:

- Sitecore;
- SharePoint;
- Power Apps;
- Collaboration Apps;
  - Teams;
  - WebEx;
- Social Media Apps;
  - WhatsApp;
  - Facebook;
- Mobile Apps;
  - Android Apps;
  - iOS Apps;
- Power BI;
  - Dashboards;
  - Reports;
- Web Apps;
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of the Stable layer are:

- Stable and robust;
- Reliable;
- Security is an important and ongoing aspect;
- Standardization is a necessity;
- Stores and manages business-critical objects and data;
- The IT systems do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Next to agile approaches, traditional methodologies can be used to develop IT system;
- Solid test processes;
- Problem-solution design;

- Fail-over strategy is essential to guarantee business continuity and ensure high availability;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The main characteristics of the Explorative layer are:

- Features are small and independent;
- The IT systems are flexible. The IT systems can change rapidly and often;
- Delivers business value at each release or iteration and ends at a given moment;
- Off-line capability is in place;
- An IT system can contain business-critical data, but it is not a requirement;
- The IT systems are easily replaceable;
- Less focus on Risk Management. Errors are already factored in;
- Only an Agile way of development is supported;
- Rapid delivery oriented. Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market;
- Focus on experimentation and exploration;
- The IT systems are mainly centered around interaction and experience.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Connectivity layer comprises a service-based environment. This can be a service bus, a cloud integration platform, or a microservice environment. The layer can even include a subset of the environments or use all three service-based environments to establish the connectivity and integration within an IT landscape.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA

We are using SAP PO as a service bus. The SAP PO system comprises the following components:

- Integration Builder;
    - Enterprise Service Repository;

    - Creation and Discovery of Services;
  - Integration Directory;
    - Set-up of Mediation Flows;
- Integration Server;
  - Adapter Engine;
    - Transformations;
  - Integration Engine;
    - Orchestration;
    - Routing;
    - Publish and Subscriptions;
  - Business Process Engine;
    - Rules;
- System Landscape;
  - System Landscape/Tenant Manager;
- Configuration and Monitoring;
  - Configuration Manager;
  - Web Service Manager;
    - Authentication;
    - Message Security;
    - Encryption;
  - Mediation flow Manager;
  - Event Handler;
  - Adapter Manager;
  - Access Controller;
    - Authorization;
  - Exception handler;
  - Logger;
  - Auditor;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflow;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;

    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

The strategy of SAP is moving towards the cloud. Therefore, they promote using the Integration Suite of SAP Cloud Platform (SCP) instead of focusing on the on-premise middleware.

SCP can be used to develop and manage end-to-end integration scenarios across cloud-based and on-premise solutions. This platform allows an organization to connect and contextualize processes and data across their heterogeneous IT landscape. It provides the ability to integrate SAP, non-SAP, cloud, and on-premise applications and processes through messages.

The platform comprises a Cloud Integration tool that acts as a message broker, API management, Open Connectors, and an Integration Advisor.

API management provides access to simple, scalable, and secure digital assets through APIs and lets internal and external developers consume it. The Open Connectors functionality provides pre-built connectors and adapters to establish seamless connectivity between SAP and non-SAP applications. And the Integration Advisor accelerates the development of business-oriented interfaces, mappings, runtime artifacts and reduces integration efforts.

#bimodal_IT_layers_Communication_layer_CIP

The components of the SCP Integration Suite are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;

        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;
            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;

- SMTP;
- Database Adapter;
- 3rd Party Adapters;
- Custom Adapters.

#Communication_layer_CIP_capabilities

We use Copado to realize DevOps. The microservices are part of this platform. However, I do not know what the components of Copado are. I can only provide you the capabilities if you like.

#bimodal_IT_layers_Communication_layer_MSA

Interviewer: Yes, please. I also do not know this tool, so I am curious if it has similar capabilities as Azure.

The tool makes use of a container orchestration platform, an API manager, and an API gateway.

#Communication_layer_MSA_components

The Container Orchestration Platform provides:

- Container Orchestration;
- Container clustering;
- Image Discovery;
- Container Security;
- Key Vault functionality;
- Load Balancing;
- Scheduling;
- Rollouts and Rollbacks;
- Self-Healing;
- And Workflow Management.

The API Manager and Gateway provides:

- Service Discovery;
- Messaging;
- Routing (of services);
- Call handling;
- Transformation;
- Rate Limiting;
- Traffic Management;
- Throttling;

- Caching;
- Fail-over;
- Microservice Security;
- Configuration Management;
- Analytics;
- API Version Management;
- Monitoring and Logging;
- Policies and Contracts;

#Communication_layer_MSA_capabilities

Also, the container orchestration platform does not only realize and facilitates the microservices. It allows an organization to set up a (virtual) infrastructure to help manage the microservice environment. But I do not know how this exactly works. I have to refer you to another colleague for this.

#Communication_layer_MSA_components

Interviewer: That is not needed because I am only interested in the (technical) components of the service-based environments.

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

In the Explorative layer, the majority of the increments are delivered through CI/CD. Occasionally, we use another agile method to deliver an increment within this environment.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

In the Stable environment, the IT systems are delivered through a mix of SDLC approaches. Generally, a traditional approach is used for the implementation and data migration of a core system. For large IT projects applies that they are developed and delivered using a project management approach. Once a core system is deployed into a production environment, an agile approach is used to maintain and exploit its functionalities and to develop new features. For the operational value-adding systems applies that the systems are mainly developed and delivered through an agile approach.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#Slow-Speed_stabilized_systems_of_innovation
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

First of all, it is important to realize that the impact is only applicable when an API is adjusted that impacts all the layers. This means that the increment should impact as well an IT system in the Stable layer as an IT system within the Explorative layer. For example, if we only need to add a new field in an IT system of the Explorative layer to capture a certain attribute, the change will not affect the underlying API. In this case, the team can perform the field enhancement with the corresponding development kit without considering the API or the related IT systems within the Stable layer.

However, we also have scenarios in which an API adjustment is impacting all three layers. An example is a feature that needs to store a new custom attribute. In this case, there is an impact on release management because several teams need to build their increments, align these, and release it simultaneously. To control these increments, we have a weekly integration session with all the teams. This session creates awareness and helps us to prioritize and guide the releases. Besides the weekly sessions, daily communication mechanisms are enforced to support the cross-team interactions and conversations.

#impact_release_management
#impact_continuous_alignment
#impact_communication

9: Do you use different methodologies across the different layers?

Yes. The Explorative layers' IT systems are realized and facilitated through CI/CD, and in some cases, through another agile approach. In contrast, the Stable layers' IT systems are developed, delivered, and maintained through a mix of SDLC approaches (traditional and agile).

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework
#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

For the Explorative layer applies that the majority of the increments are delivered through CI/CD.

This technique is also used to deliver the increments within the microservice environment.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA

#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

The layers that are affected by CI/CD are:

- The Explorative layer;
- And in the case of a microservice environment, the Connectivity layer is also affected.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

As indicated, we use Copado to enable and ensure CI/CD.

Before we can use Copado, the increments need to be built locally by the developers. A developer can use any development kits for this. Once the coding has been performed, the source code is submitted into a Repository. Through this Repository, Copado can perform its activities.

#continuous_integration_and_delivery_components
#continuous_integration_and_delivery_systems_and_software

The features of Copado are:

- Copado Plan. To create, manage and maintain the Backlog, User Stories, and Sprints;
- Copado uses Git as a Repository;
- Copado Pipeline. Copado Pipeline is an automation server. However, this functionality can also be integrated with Jenkins and other third-party automation servers. The Pipeline functionality is used to automate the build, test, and publish activities;
- Copado Release. The release functionality is used to automate the deployment step.
- Copado Monitor. This tool can be used for monitoring and analysis.

We use Postman to test and mock the APIs.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The following practices are required to ensure CI/CD:

- It is important to have a CI/CD strategy in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before each deployment;
- Configurations and passwords should be accessible and always up-to-date;
- All the affected layers must confirm to an agile approach. Preferable DevOps;
- Use of a Version Control System;
- Use of an Automation Server that can automatically build, test, publish and deploy the increment;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

We have the following tools and components in place to support the development of software in all the layers:

- A tool to manage the planning and requirements;
- A Project management or an IT Service Management system to manage the related activities;
- A ticket management system;
- Collaboration tools;
- Functional and technical design tools;
- A Document Management System;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- An automation server to automate the build, test, publish, and deploy activities;
- Test tools;
- Monitoring and Analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The tools and components are realized through the following systems:

- JIRA is used for project management activities and the creation of tickets;
- The applications of Office 365 are used for requirements engineering, creation of functional and technical designs, and used to capture the test results;
- SharePoint is used to store all documentation;

- We use the following tools to develop, change, and configure systems, programs, applications, features, and APIs:
    - SAP Netweaver to support ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse;
    - Xcode 12 and Swift UI;
    - Android Studio;
    - Lightning platform and Heroku;
- Copado is used to enable DevOps and supports the automation activities;
- We are using the following Test Management Tools;
    - Copado Test;
    - Polarion;
- We are using the following tools for deployment:
    - Copado Release is used to deploy increment in Salesforce;
    - CHARM is used in SAP Solution Manager to deploy SAP transports;
    - Eclipse is used for SAP iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

There is no relationship between the deployment models and Bimodal Architecture. However, there is a trend visible. In general, the Systems of Record are deployed on-premise or in a data center, such as SAP S/4 HANA. Via these deployment models, an organization can control its core systems and associated functionalities. In contrast, the Systems of Differentiation, such as Salesforce, are deployed in the cloud as a SaaS solution.

#Slow-Speed_systems_and_software
#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud_SaaS

For the Explorative layer applies, that the Systems of Innovation are generally deployed in the cloud.

#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

Universiteit Leiden

Interviewer: Thank you for the answer. Can you also elaborate a bit on the infrastructure of the deployment models? For example, how are the models realized or facilitated?

Yes. First, let us look at on-premise or data center environments. In these environments, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can only be reached and executed via a private network, which is established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

In the case of a cloud environment, the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through container technology and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- or Hybrid models.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

I believe that most IT systems will be deployed in the cloud within the upcoming few years. Many established vendors, such as SAP, Microsoft, and Oracle, are already moving to these deployment practices.

#deployment_model_movement_towards_cloud

Universiteit Leiden

15: What were the major milestones that lead to the current set-up of the IT landscape?

In 2018, all core systems were consolidated into four main systems. These systems are:

- SAP S/4 HANA;
- Salesforce;
- Apttus;
- And ClickSoftware.

This consolidation allowed us to reshape our IT landscape and make it more future-proof.

#organization_IT_milestones
#organization_current_IT_landscape

16: What is the future roadmap of the IT landscape of your organization?

I foresee an introduction to Machine learning and Robotics. Our coffee machines are becoming smarter and are providing a new experience to their customers. Therefore, the IT landscape needs to be able to adopt these IoT developments and provide a smoother experience and better interaction with the several services that we have.

#organization_IT_strategy
#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

The Bimodal IT concept still has a bright future because it is inevitable not to split up the IT landscape into two separate environments and link this through a Connectivity layer. Due to this approach, we can deal with the IT systems that support our operations. And at the same time, we can deal with the systems that manage and operate the IoT devices and improve the various stakeholders' experiences.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

We are already using agile teams within the Explorative layer, so the team set-up is not jeopardized in this case.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_agile_teams

For the Stable layer applies that we still need an agile team for some core systems. However, the business itself should give priority to the agile way of working. We cannot decide that for them.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_traditional_teams
#concept_bimodal_IT_prioritization

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

In the Stable layer, the priority is set by the business. An IT project can only start after the approval of the business.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_roles
#concept_bimodal_IT_traditional_teams

In the Explorative layer, the priority is set by a PO. The costs of the associated agile teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt?

The technical debt in the Explorative layer is negligible because the increments are delivered through an automation server. Also, the IT systems and applications can invoke a maximum number of services. This means that fixes can be applied very fast if there are any errors, issues, or problems.

In contrast, the Stable layers' IT systems and applications can have a high technical debt because they contain custom build functionalities. Also, there is an urgent need to expose the various functionalities as services to realize and facilitate an API platform.

#concept_bimodal_IT_technical_debt

Candidate 13 – DEV-02

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

I have twelve years of experience in SAP, mostly in the SAP BI back-end system, such as ABAP development, Authorization, RFC connections, Data Flows, and ETL. I also have extensive experience in the related BI tools, such as the Report Builder, Crystal Reports, Dashboards, etc. Also, I have two years of experience in Master Data Management (MDM), from requirements gathering to UI development in SAP NetWeaver (Java-based developments). I have three years of web development experience on top of the SAP experience, such as PHP, MySQL, and JavaScript. And the last two and a half years, I am also working with Oracle Data Warehousing.

#candidate_experience

Currently, I am fulfilling the role of a data engineer at one of the municipalities within The Netherlands {Municipality-01}. To be more specific, my role can be better described as a Data Warehouse Developer {DEV-02}, because the municipality has around 60 source systems from which the data needs to be extracted and combined. To achieve this goal, a lot of developments need to be done to convert the data so that it becomes useful to make decisions and perform planning.

#candidate_current_role
#IT_capability

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The entire IT organization counts approximately 400 employees. Our division has 30 employees. The internal IT organizations' core function is to maintain and leverage the current IT landscape while also helping to digitize, innovate, and become a data-driven organization. It is also important to keep in mind that a municipality needs to comply with several regulations and meet its residents demand in supplying information, services, and products.

#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

From my perspective, Bimodal Architecture is used to "keep the lights on" and, at the same time, help the organization to digitalize. Therefore, one IT platform comprises core systems, like SAP, Oracle eBS, Cognos, and MDM tools to guarantee business continuity and standardize and automate the business processes. And another IT platform comprises systems that contribute to a digital city. In general, the platforms' related systems are providing information about the city and realizing digital services and products for the residents to support them with the interaction and affairs towards the municipality.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_characteristics
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

The Bimodal Architecture environment divides three main layers. The first layer is stable and covers the core systems of the organization. Besides the core systems, the layer also comprises internal APIs and associated solutions. These systems and applications are adding value for the municipality by capturing key-information and extending the business processes. The second layer is explorative and covers the fast-changing systems and applications. Based on feedback and evaluations, the layers' systems and applications can change rapidly or even be replaced by other smart, sustainable solutions. Therefore, the layer is highly focused on innovation and experimentation. With this platform, the municipality can enhance the experience of its residents and other relevant stakeholders. And the third and final layer includes one or more service-based environments to establish the connectivity, integration, and communication patterns through the usage of APIs (services).

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#Fast-Speed_characteristics
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

Also, I like to highlight that we have a separate EAI strategy in place within the Stable layer to complement the integration patterns. Through EDI and JDBC integration mechanisms, we are able to connect the core systems which each other, like Oracle eBS and Cognos, or the BRP system and the Social Security System. On top of these integration mechanisms, multiple proxies and APIs are effectuated to extract the core systems' data and objects.
#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#bimodal_IT_integration_patterns_Slow-Speed

#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services

Subsequently, these services can be used as internal or external APIs to realize extensibility, modularity, and interoperability between and within the Stable and Explorative layers to enhance the several stakeholders' experience.

#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Slow-Speed
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_integration_patterns_Fast-Speed

Presentation of the models

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can recognize the models. As already indicated, the Stable layer comprises the core systems, like Oracle eBS, Cognos, and MDM. Also, solutions for reporting bulk waste and vandalism or capturing key-information about rain volumes and crowdedness are part of this layer.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The Explorative layer is mainly focused on products and services for residents, like the resident portal, requesting parking permits, neighborhood prevention, and other smart and sustainable solutions. Also, tourist applications, like for buying a discount pass, are embedded in this layer.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The Stable layer consists of:

- SAP;
- Oracle eBS;
- Oracle Data warehouse and SAP BI;
    - ETL;
    - Reports Builder;
    - Dashboard Builder;
- Cognos;
- MDM;
- BRP (Basic Registration of Persons) system;
- Social Security System;
- Sensor APIS and associated systems;
    - Measuring rainfall;
    - Measuring crowdedness;
- Municipality adding-value systems;
    - Reporting bulk waste;
    - Reporting vandalism;
- Internal APIs and associated systems;

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The Explorative layer consists of:

- Portals;
    - Resident;
    - Employee;
    - Partner;
- Web Apps;
    - Requesting parking permits;
    - Car parking App;
    - Buying Discount passes;
    - Visit sign up App;
    - Tourism Info App;
- Social Media Apps;
    - WhatsApp;
        - Neighborhood prevention;
    - Facebook;
- Self-service Apps;
- Oracle Data warehouse and SAP BI;
    - Dashboards;
    - Reports;
- Mobile Apps;
    - iOS;
    - Android;
- Other external APIs.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software

As already indicated, the apps and portals embedded in the Explorative layer are continuously changing due to its stakeholders' fast-changing demands.

#Fast-Speed_characteristics

Interviewer: Thank you for providing the systems/applications for each layer. However, can you explain why you distinguish the Data Warehouse functionalities?

Yes. The functionalities are divided because some components of the BI system are stable, while others are not. For example, if we look at the ETL (Extract, Transform, and Load) functionality, we can mark this feature as stable. The flow to retrieve a specific object and its corresponding values (data) is only built once and applied to all the systems to retrieve the particular data set. And, of course, for each system, the mappings and the connectivity details are different. But once these activities are performed and the flow is set-up, it does not change anymore until the connection or the data set itself becomes obsolete. When we look at Dashboards and Reports, we see that the information requirements are based on a certain user, unit, or role. This means that the visualization, interpretation, and output of the data need to be changed continuously. Therefore, Dashboards and Reports can never be part of the stable environment.

#Slow-Speed_systems_and_software
#Slow-Speed_systems_and_software_functionalities
#Fast-Speed_systems_and_software
#Fast-Speed_systems_and_software_functionalities

I hope that this clearly explains the split of the functionalities within the BI domain.

Interviewer: Yes, it does. Thank you for the elaboration on this point.

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The characteristics of the Stable layer are:

- Stable;
- Robust;
- Reliable;
- Focus on standardization and automation of business processes;
- Stores and manages business-critical objects and data;
- Fail-over strategy is essential to guarantee business continuity and ensure high availability;
- Makes use of agile and traditional approaches to develop and deliver systems and applications;

- Delivers operational value;
- Reliable test procedures and processes are essential;
- Systems do not change a lot once they are deployed into production;
- Problem-solution design;
- Implementation costs are high;
- Systems are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The characteristics of the Explorative layer are:

- Features are small and independent;
- Rapid delivery oriented. Often and frequent deployments;
- Only Agile approaches are used to deliver the systems and applications;
- Delivers business value at each release or iteration;
- Off-line capability in place to guarantee 24/7 availability;
- The system can contain business-critical data, but it is not a requirement;
- The systems are easily replaceable;
- Focus on innovation, experimentation, and exploration;
- Continuous integration/deployment in place to increase time-to-market;
- The solutions are mainly centered around interaction, engagement, and experience;
- Follows the latest trends, such as the latest software development standards and technological innovations.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

We are using three service-based environments in our IT landscape to establish connectivity, integration, and communication. The service bus of Oracle is used as an ESB and integrates the core systems. We also have the middleware system of Azure in place to connect the on-premise and cloud solutions. And we are using Azure to realize the microservice environment. All the platforms are embedded within the Communication layer and are used to establish and utilize the integration patterns within and between the Stable and Explorative layers' systems.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

Universiteit Leiden

The (technical) components of the Oracle service bus are:

- Message Builder;
  - Set-up of Adapters;
  - Creation of Services;
  - Set-up of Mediation Flows;
- Integration Engine;
  - Adapter Engine;
  - Orchestration Engine;
  - Transformation Engine;
  - Routing Engine;
  - Rule Engine;
  - Publisher and Subscriber Engine;
- Integration Controller/Integration Gateway;
  - Configuration Manager;
  - Web Service Manager;
    - Authentication;
    - Message Security;
    - Encryption;
  - Access Controller;
    - Authorization;
  - Exception handler;
  - Event Handler;
  - Mediation flow Manager;
  - System Landscape/Tenant Manager;
  - Auditor;
  - Logger;
- Service Repository;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
  - Workflow;
  - Rules;
  - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
  - Statistics and Status;
  - Alerts;

    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

We use the Azure Integration Services to integrate the cloud and on-premise applications through services. The API manager and associated Gateway provides access to systems and lets internal and external developers consume the APIs.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of Azure Integration Services are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Mediation Flows;
        - Integration/Message Builder;
            - Set-up of Mediation Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;
            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;

- Routing Engine;
- Rule Engine;
- Publisher and Subscriber Engine;

- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;

    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

And we have an Azure platform in place to enable microservices because the best practice to run microservices is through containers. The containers themselves are managed via a container orchestration platform, which is also offered by Azure. Therefore, the Azure environment is deployed as a cluster and consists of a set of nodes. These nodes are hosting pods, which represent a set of running containers.

#bimodal_IT_layers_Communication_layer_MSA

Essential components of the Azure platform are:

- Container Orchestration engine;
- Resource Manager;
- Scheduler;
- Controllers;
- Key-value Database;
- API Manager;
- API Gateway;
- Storage;
- Queues;
- Logs;
- Container Registry;
- Security Manager.

#Communication_layer_MSA_components

Azure realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Fail-over;

  - Microservice Security;
  - Configuration Management;
  - Networking/IP;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Container;
  - Microservices.

#Communication_layer_MSA_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

In the case of upgrades (mostly executed on core systems), system refreshes, and the implementation of a new core system, we are making use of the Waterfall methodology. However, when we look at RFC's, initiatives, and new features, all these follow-up developments are executed in an agile way. We have adopted Scrum for this.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework_waterfall
#methodology_agile_framework_Scrum

For the Explorative layer applies that we are only using agile methodologies, such as CI/CD and Scrum, to realize the associated systems and applications.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework_Scrum

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

On one side, the impact is related to release management, and on the other side, how the business conducts the prioritization.

#impact_release_management
#impact_prioritization

When we create or adjust an API, all the associated systems using, or will use that particular API, needs to be identified and investigated. After the assessment is completed, the right priority needs to be set for each system and increment. This approach increases the alignment and guarantees the proper working of the functionality after the release.

#impact_continuous_alignment

9: Do you use different methodologies across the different layers?

Yes. The common way to deliver a new core system within the Stable layer is via a traditional approach, mainly through Waterfall. However, once the system is live and we need to make a small change or add new features, we use Scrum. In contrast, the developments within the Explorative layer are always done in an agile way. For Azure and Mendix developments applies that the majority of the increments are picked-up through CI/CD.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_traditional_framework_waterfall
#methodology_agile_framework_Scrum
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#Fast-Speed_systems_and_software
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is embedded in the Explorative layer and used within the Communication layer by a specific service-based environment. The CI/CD process is mainly triggered and used by the Mendix and the Azure platform. These platforms provide all the tools needed for the continuous integration and delivery of the user requirements and support the deployment on various platforms, like the several system environments (development, test, acceptance, and production), smart devices, etc.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_components
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

As already indicated, the Explorative layer and the microservice environment are affected by CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

We have adopted Bamboo from Atlassian to enable CI/CD.

The first step is to develop or adjust the source code locally. Once this step is performed, the code needs to be submitted into a Repository. We use Bitbucket for this. After the code is stored within the Repository, Bamboo can build the software packages and deploy these into a specific environment.

#continuous_integration_and_delivery_components
#continuous_integration_and_delivery_systems_and_software

The components of Bamboo are:

- Bamboo Build. Bamboo Build is an automation server. This functionality is used to automate the build, test, and publish activities. It can also provide status updates on successful and failed builds;
- Bamboo Deploy. This functionality is used to automate the deployment step and to release the software package to any environment;
- Bamboo Reports. This tool can be used for monitoring and analysis.

We also use Postman to mock and test the APIs.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

In general, the common configuration practices of CI/CD are:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before every deployment;
- The use of a Version Control System;
- an Automation Server needs to be in place that performs the following tasks:
    - Automated builds;
    - Automated tests;
    - Automated publisher (to generate a package that contains all the files to deploy and run a piece of software);
- A Release Manager tool that can deploy the application on almost any environment;

- Monitoring and visualization tools should be in place to measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

The following tools and components are applicable when we develop and deliver the systems, applications, and solutions within the different layers:

- A tool to manage the planning and requirements;
- A Project management or an IT Service Management system to manage the related activities;
- A ticket management system;
- Collaboration tools;
- Functional and technical design tools;
- A Document Management System;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- A Version Control System;
- An automation server to automate the build, test, publish, and deploy activities;
- Software Quality Assurance tools. This also includes Test Management tools;
- Release and deployment tools;
- Monitoring and Analyzing tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

We use the following applications to realize the development and delivery of the systems, applications, and solutions within the different layers

- JIRA. JIRA is used for the Project Management and IT Service Management activities. It is also used to create, manage and maintain the Backlog, User Stories, Sprints, and Work items;
- Office 365. The related tools are used for requirements engineering and the creation of functional and technical designs.
- SharePoint is used to store all documentation;
- Bitbucket. Bitbucket is used as a Version Control System;
- We use the following tools to develop, change, and configure systems, programs, applications, features, and APIs:
    - JDK (Java Development Kit);
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Xcode 12 and Swift UI;

- Android Studio;
- Bamboo is used to enable DevOps and supports the automation activities;
- We are using the following Test Management Tools;
    - SupportBook. This tool is used to create and manage test scripts and test scenarios in traditional and agile software developments. The tool also supports test automation;
    - Bamboo Tests. This feature is part of the Bamboo Build functionality;
- We are using the following tools for deployment:
    - Bamboo Deploy. This functionality is used to deploy the increments in an automated way.
    - JBoss EAP. This is a Java application server and runtime platform to build, deploy and host Java software and applications. JBoss EAP is also integrated with Bamboo;
- Puppet. The puppet platform is used as an infra-as-code environment to automate the IT infrastructure and to simplify configuration management;
- We are using the following tools for monitoring:
    - Bamboo Reports.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

From my perspective, there is no relationship determinable. In practice, most systems within the Stable layer are deployed on-premise or in a data center. It occurs that some systems are subscribed as a SaaS solution.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud_SaaS

In contrast, the systems within the Explorative layer are mainly deployed in the cloud and delivered as a SaaS or PaaS solution.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#deployment_model_cloud_SaaS
#deployment_model_cloud_Paas

Interviewer: Can you also elaborate a bit on the infrastructure of the deployment models. And how are the deployment models realized or facilitated?

In an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. For security reasons, the Server Farm servers and services can only be reached and executed via a private network, which is established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

In a cloud environment, the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through containers and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;
- Private;
- or Hybrid models.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

As a municipality, we are noticing that there is a movement going on towards the cloud. We believe that the common deployment practice for software will shift to the cloud within a couple of years. However, we need to be cautious because we deal with very critical data. Therefore, we need to comply with certain rules and regulations. This means we have to verify which systems can go to the cloud and which systems need to stay or be deployed on-premise or in a data center.

#deployment_model_movement_towards_cloud
#Slow-Speed_systems_and_software
#deployment_model_on-premise
#deployment_model_data_center

15: What were the major milestones that lead to the current set-up of the IT landscape?

I cannot answer this question because I am working for two and a half years for this municipality.

16: What is the future roadmap of the IT landscape of your organization?

Our IT vision is to become a data-driven organization.

Also, we want to establish the following changes within our domain in the upcoming years:

- Replacement of the Oracle eBS environment with another solution;
- Verify the possibility of cloud solutions.

#organization_IT_strategy
#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Bimodal Architecture will remain a valid concept because the need for two separate IT environments will never disappear. We will still need stable systems to ensure business continuity and store the business-critical data. While we also want to explore, experiment, and innovate.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt?

The respondent did not answer the question due to time.

Candidate 14 – ITL-03

General questions:

1: Personal information: years of experience, and current function/position/role within the company or project.

I started my career in 2010 as a technical consultant and developer (SAP ABAP, Java, Python, PHP, JavaScript, Go, and C-Sharp) at a big consulting firm. For the last four years, I am fulfilling mainly the role of an Integration and Technical Lead {ITL-03}. I joined my current client in 2017, and since then, I am roaming around within the organization. Even though I have spent the last three years at the same client, I have fulfilled multiple roles ranging from Business Integration Manager to SAP Basis and Development Lead. I have also supervised and executed several projects internally. Currently, I am fulfilling two roles within a project. The first role is that of an Integration Test Lead, and as the second role, I am a Scrum Master for one of the SAP teams. Besides these roles, I am also helping the management in shaping the future IT strategy.

#candidate_experience
#candidate_current_role

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The client itself is operating within the IT sector. It is a cloud hosting provider {Hosting-01} that is active worldwide. The organizations' focus is on automated delivery of servers and services, like hosting public/private servers, cloud services, etc. The company has around 500 employees, and most of the resources are working within the internal IT capability to provide the various services.

#company_profile
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

I see Bimodal Architecture as a combination of a Steady/Stable IT environment and a separate Digital IT platform with systems and software that need to be delivered on short notice. The Digital platform comprises mainly innovative systems and applications with a high focus on exploration, experimentation, engagement, and interaction. Each of the environments has his approach for realizing RFC's, patches, and executing maintenance and exploitation activities.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics
#methodology

Besides this description/explanation, I also like to indicate that a critical environment lacks in the Bimodal approach, namely an Integration layer. Therefore, I would like to call the concept rather "Trimodal IT" than Bimodal IT because several middleware technologies are used to establish connectivity and communication within and between the Stable and Digital layers.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

Before I mention the Bimodal environments' main components, I would like to indicate a clear split between the hosting, infrastructural (network), and software components. This is important because a Bimodal IT environment combines and uses multiple components that belongs and impacts several organizational, functional, technological, and external aspects:

Infrastructural components:

- Physical servers and devices, such as application servers, database servers, file servers, exchange servers, web servers, etc.;
- Physical network, devices, systems, and software, such as firewalls, routers, IP standards and addresses, Internet connections, domains, etc.;
- Virtualization, such as containerization (LXC, LXD, LXCFS, and Docker) or through the use of a VMware (bare-metal hypervisors).

#hosting_platform_on-premise_components
#hosting_platform_data_center_components
#hosting_platform_cloud_components

Hosting models:

- Cloud, such as Public, Private, or Hybrid models. Within these models, you can also choose how you want to subscribe to the services, systems, and applications:

    - Software as a Service (SaaS);
    - Platform as a Service (Paas);
    - Integration Platform as a Service (iPaaS);
    - Function as a Service (FaaS);
    - Back-end as a Service (BaaS);
    - Infrastructure as a Service (IaaS);
- Data centers;
- On-premise.

#deployment_model
#deployment_model_on-premise
#deployment_model_data_center
#deployment_model_cloud

Systems and software:

- Development systems, such as:
    - Integrated Development Environments (IDEs);
    - Development Kits (DKs);
    - DevOps tool to enable CI/CD;
- Quality Assurance and Test Management solutions;
- Systems and software for monitoring, visualization, and analysis;
- Middleware systems, such as:
    - An Enterprise Service Bus. Think of:
        - SAP Process Orchestration (PO);
        - Biztalk;
        - IBM WebSphere;
        - BEA AquaLogic;
        - Or Oracle Service Bus;
    - Cloud Integration Platform. Famous Cloud Integration Platforms are:
        - SAP Cloud Platform (SCP);
        - Azure Integration Services;
        - And AWS Application Integration;
    - A Microservice platform. A Microservice platform can be realized through:
        - Azure;
        - AWS;
        - Google Cloud;
        - Or Anypoint from MuleSoft;
- Container Orchestration Platforms to virtualize and orchestrate the containers in a microservice environment, such as:
    - Kubernetes;

Universiteit Leiden

    - Azure;
    - AWS Elastic Container Service;
    - Docker Swarm;
- Web- and Microservices, which are realized through:
    - APIs;
    - SOAP services;
    - REST services;
    - And ABAP proxies;
- Information systems, such as:
    - ERP;
        - SAP S/4HANA;
        - Dynamics 365;
        - Oracle eBS;
        - AFAS;
    - CRM;
        - SAP C4C;
        - Salesforce;
        - Dynamics 365 CRM;
    - SCM;
        - SAP Ariba;
        - Anaplan;
        - Oracle SCM;
        - Luminate Planning;
    - Ticketing systems;
        - JIRA;
        - ServiceNow;
        - TopDesk;
    - BI systems and analytics tools;
        - Power BI;
        - SAP Data Warehousing and BI;
        - Google Analytics;
        - Grafana;
- Intuitive engagement and interaction systems, such as:
    - Customer, user and supplier portals;
    - Collaboration systems and software;
    - Web Apps;
    - Mobile and smart device applications;
- Innovative systems and solutions, such as:
    - IoT solutions;
    - Machine and Deep Learning solutions;
    - Augmented Reality solutions;

- Blockchain solutions;
- Artificial Intelligence solutions.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#Fast-Speed_systems_and_software
#Communication_layer_systems_and_software
#software_delivery_life_cycle_systems_and_software

From my perspective, the main components of a Bimodal IT environment are:

- A Stable layer. This layer comprises Information Systems (IS) and value-adding systems, such as internal web Apps, mobile Apps, and some IoT technologies. The environment is not only used to realize and ensure operational excellence but also to gain valuable insights. Therefore, the embedded systems within this layer support and extend the business processes, store and manage business-critical objects and data, enable (process) automation, reduce complexity and costs by increasing efficiency, and support internal digitalization. When these systems are compromised, it can jeopardize the business continuity;
- A Digital layer. This layer focuses mainly on experimentation and exploration. The environment is continuously developing and providing digital products and services to its stakeholders. Therefore, the associated systems are always invoking a number of services, so they are modular, easily adjustable, and replaceable.
- A Communication layer. Without this layer, it is impossible to realize and exploit an API/service platform. The layer comprises one or more service-based environments to establish connectivity, integration, and communication within and between the Stable and Digital layers.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation
#Slow-Speed_characteristics
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#Fast-Speed_characteristics
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

Presentation of the models

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes, I can relate to the models. To put the layers into perspective, the client has several IS's that are situated in the Stable layer. Like the portals and external web Apps, some of the systems are embedded in the Digital layer. And the communication layer is used to connect the Stable layer with the Digital layer via web- and microservices. These services are effectuated through multiple service-based environments.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_systems_and_software
#bimodal_IT_layers_Fast-Speed
#Fast-Speed_systems_and_software
#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

Although your models contain all the relevant components, I like to indicate that the value-adding systems are not covered within the models. Let's take SAP S/4HANA as an example. The core modules of SAP S/4HANA, such as Sales, Sourcing and Procurement, Supply Chain, Service, Finance, Asset Management, and Manufacturing, are situated in the Stable layer. However, SAP S/4HANA also comprises components that support digitalization and user engagement, like Fiori applications. Besides Fiori, SAP S/4HANA also allows its functionalities to be exposed as services via proxies, custom function models and methods, and Enterprise Services. Subsequently, these services can be used to realize and facilitate systems and applications within the Digital layer.

#Slow-Speed_systems_and_software
#Slow-Speed_systems_and_software_functionalities
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Slow-Speed_facilitates_Fast-Speed

Also note, that the connection between the SAP back-end system and Fiori is established via OData services and the SAP Gateway. These integration patterns are also missing in your models.

#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services

Explanation Slow-Speed and Digital layer:

Systems and software within the Stable layer are not required to change often, while in the Digital layer, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Stable layer and which are typically incorporated into the Digital layer?

Typical systems of the Stable layer are:

- ERP systems and other core IS's;
  - SAP S/4HANA;
  - Dynamics 365;
  - Oracle eBS;
  - AFAS;
  - SAP C4C;
  - Salesforce;
  - SAP Ariba;
  - Anaplan;
  - Oracle SCM;
  - Luminate Planning;
- Project Management and ticketing systems;
  - JIRA;
  - ServiceNow;
  - TopDesk;
- BI Systems, such as Power BI, and SAP Data Warehousing and BI;
  - ETL;
  - Report Builder;
  - Dashboard Builder;
- SAP Fiori applications;
- Internal web Apps;
- Internal mobile Apps;
- Industry-specific systems;
  - Product database (custom application);
  - Auto-install images (custom applications);
- IoT technologies and solutions;
  - Sensors;
  - Machine and Deep Learning;
  - Augmented Reality;
  - Blockchain;
  - Artificial Intelligence;
- Internal APIs.

#Slow-Speed_systems_and_software

#Slow-Speed_stabilized_systems_of_innovation

Typical systems of the Digital layer are:

- Portals (Sitecore);
    - Customer;
    - Employee;
    - Supplier;
- E-commerce software;
- Collaboration systems and software;
- BI and analytics tools, such as Power BI, and SAP Data Warehousing and BI, Google Analytics, and Grafana;
    - Reports;
    - Dashboards;
- Power Apps;
- Web Apps;
- Mobile and smart device applications;
    - iOS Apps;
    - Android Apps;
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Digital layer? Can you elaborate on the characteristics of the systems/applications within the Stable layer and also on the characteristics of the Digital layer?

The characteristics of the Stable layer are:

- Stable;
- Robust;
- Focus on reliability;
- Highly secure;
- The IS's are supporting the business processes. These systems are also storing and managing the business-critical objects and data;
- The IS's do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Characterized by long release cycles. When an agile framework is used, it costs multiple sprints to deliver or update a particular IS;
- Makes use of agile and traditional approaches to develop and deliver an IS;
- Fail-over strategy needs to be in place to guarantee business continuity and guarantee high availability;
- Highly influenced by external parties;

- Implementation costs are high;
- The systems are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The characteristics of the Digital layer are:

- Features are small and independent. Applications and functionalities invoke one or more independent services;
- Delivers business value at each release or iteration and ends at a given moment;
- Off-line capability is a necessity to ensure 24/7 availability and for a high experience;
- A system can contain business-critical data, but it is not required;
- The systems are easily replaceable;
- Less focus on Risk Management. Errors are already factored in;
- Focus on experimentation and exploration;
- Easily adaptable. Ease of making quick changes or replace features with newer versions;
- Only an Agile way of development is supported;
- Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market;
- Planning changes continuously.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Stable layer with the Digital layer.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

As already indicated, the Communication layer can comprise one or more service-based environments to establish the integration and communication patterns.

#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

At the client, the IT-landscape comprises two service-based environments to realize the multiple integration patterns. Since all their systems are already in the cloud, they have a cloud integration platform in place to create, transform, route, and consume the services. Also, they have a microservice environment in place to realize and leverage the microservices.

#bimodal_IT_integration_patterns_through_Communication_layer

#bimodal_IT_layers_Communication_layer_MSA_services

The client uses the Integration Suite of SAP Cloud Platform (SCP) as a cloud integration platform. This middleware solution provides the ability to integrate SAP, non-SAP, cloud, and on-premise applications and processes through services.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of this middleware solution are:

- Cloud Integration Service;
    - Message and Event Hub;
        - Repository for Service Discovery and pre-build Integration Flows;
        - Integration/Message Builder;
            - Set-up of Integration Flows;
            - Editing of Flows;
        - System Landscape/Tenant Manager;
        - Transaction Handler;
        - Configuration Manager;
        - Security Manager;
            - Set-up of Message Security;
            - Set-up of Encryption;
        - Access Controller;
            - Set-up of Authorization;
        - Event Handler;
            - Set-up of Events (Event Producer and Event Listener);
        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;

                - Editing of Workflows;
            - Access Controller;
                - Identity Access Manager;
                - Set-up of Authorization;
            - Logger;
            - Auditor;
        - Integration Engine;
            - Adapter Engine;
            - Orchestration Engine;
            - Transformation Engine;
            - Routing Engine;
            - Rule Engine;
            - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;
        - Set-up of Workflows;
        - Editing of Workflows;
    - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

The platform realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;

  - Alerts;
  - Failover;
  - Configuration Management;
  - Deployment;
  - Load Balancing;
  - Service Registry;
  - Service Discovery;
  - Call handling;
  - Message Tracking and Throttling;
  - Rate Limits;
  - Traffic Management;
  - Exception Management;
  - Analytics;
  - API Version Management;
  - Monitoring and Logging;
  - Policies and Contracts;
- Mediation;
  - Transformation;
  - Protocol Translation;
  - Caching;
  - Message Enrichment;
  - Dynamic Routing;
  - Message Validation;
  - Reliable Messaging;
  - Pass-Through of Messages;
- Security;
  - Authentication;
  - Authorization;
  - Encryption;
  - Certification;
  - Message Security;
- Transportation;
  - HTTP and HTTPS;
  - SOAP;
  - REST;
  - OData;
  - XML;
  - FTP and SFTP;
  - Flat File;
  - SMTP;
  - Database Adapter;
  - 3rd Party Adapters;
  - Custom Adapters.

#Communication_layer_CIP_capabilities

To realize and facilitate microservices, the client is using a microservice environment of Microsoft, namely Azure. The Azure platform is deployed as a cluster and comprises a container orchestration platform and an API Gateway. The cluster contains a set of nodes, which on his turns hosts a set of Pods, which represents a set of running containers (microservices).

#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services

The container orchestration platform comprises the following (technical) components:

- Scheduler;
- Controllers;
- Key-value Database;
- API Gateway;
- Resource Manager;
- Container Orchestration Engine;
- Storage;
- Queues;
- Container Registry;
- Logs;
- Security Manager.

#Communication_layer_MSA_components

The platform realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;

    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limits;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Fail-over;
    - Microservice Security;
    - Configuration Management;
    - Networking/IP;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Containers;
    - Microservices.

#Communication_layer_MSA_capabilities

Interviewer: Thank you for the extensive description of the components. However, I am also curious about the components of an ESB. Can you maybe also provide the components of SAP PI/PO to make the list complete?

Yes, of course. The (technical) components of SAP PO are:

- Integration Builder;
    - Enterprise Service Repository;
        - Creation and Discovery of Services;
    - Integration Directory;
        - Set-up of Mediation Flows;
- Integration Server;
    - Adapter Engine;
        - Transformations;
    - Integration Engine;
        - Orchestration;
        - Routing;
        - Publish and Subscriptions;
    - Business Process Engine;
        - Rules;
- System Landscape;
    - System Landscape/Tenant Manager;

- Configuration and Monitoring;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Mediation flow Manager;
    - Event Handler;
    - Adapter Manager;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Logger;
    - Auditor;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

The service bus realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;

- Reliable Messaging;
- Pass-Through of Messages;

- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

The company-wide best practice for delivering systems within the Digital layer is through CI/CD. In some cases, the increments can also be delivered through another agile approach, like Scrum. By applying CI/CD, the same release moment for all the increments can be ensured.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

The systems within the Stable layer have their own SDLC methodologies. Larger IT projects are picked-up through PRINCE2 or Waterfall. While small RFC's and enhancements are picked-up in Sprints. For the adding-value systems applies that they are always picked-up through an agile approach.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#Slow-Speed_stabilized_systems_of_innovation
#methodology_agile_framework

Universiteit Leiden

Authorization requests and small customizing changes are not accommodated into a particular delivery approach. These requests are picked-up manually and in between.

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

The impact is mainly related to release management. Since there are different software delivery approaches used within the IT landscape, it can cause discrepancies during deployments/releases and even result in severe connectivity and integration errors. Therefore, to prevent and mitigate the indicated errors and issues, the multiple teams must frequently communicate to ensure alignment.

#impact_release_management
#impact_communication
#impact_continuous_alignment

9: Do you use different methodologies across the different layers?

Yes, the layers are using their own SDLC methodologies. The Stable layer uses a mix of traditional and agile SDLC methodologies. In contrast, the Digital layer is accommodated through CI/CD or another agile approach.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework_continuous_integration_and_delivery
#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is used to develop and deliver increments (applications and features) quickly. Mostly in a day or two. Therefore, CI/CD is very suitable for the Digital layer and the delivery of microservices.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

Within the indicated environments, the build, test, publish, and deploy activities are executed automatically. The developers are able to submit a new version of the source code multiple times a day. Every night, the files are picked up automatically from the Repository and generated into a software package. Subsequently, the specific programs are built,

Universiteit Leiden

tested, and deployed towards an Acceptance environment. After the UAT is approved, the respective package is automatically released the same day to the Production environment.

#continuous_integration_and_delivery_components

10b: Which layers are affected by CI/CD?

Only the Digital layer and the microservices environment are affected by CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

However, I believe that CI/CD will also become a common practice within the Stable layer. The reason is that even the prominent vendors, like SAP, have started adopting CI/CD. For example, SAP Fiori is now also supporting CI/CD.

#bimodal_IT_layers_Slow-Speed
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

To enable CI/CD, the client has adopted Azure DevOps.

The following functionalities are used to enable CI/CD:

- Azure Boards. To create, manage and maintain the Backlog, Sprints, User Stories, Test Scripts, Work items, and Bug tracking;
- Azure Repos. Azure Repos acts as a Version Control System and uses Git as Repository;
- Azure Pipeline. This functionality can be integrated with Jenkins. This functionality is used to automate the build (via Maven), test (via Sonar), and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. It uses Azure Functions to execute the deployment;
- Azure Monitor and Grafana. These tools are used for monitoring, analysis, and visualization.

The client also has Postman in place to test and mock APIs.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

There are no separate configuration practices for each layer. To comply to CI/CD, the following conditions needs to be met:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before each deployment;
- Use of a Version Control System needs to be encouraged;
- Use of an Automation Server that can automatically build, test, publish and deploy the various increments;
- Provide KPIs to measure the pipeline executions;
- Monitoring and analytics need to be in place. To measure and analyze the solutions.

#continuous_integration_and_delivery_practices

These conditions will allow an organization to commit changes daily and to shorten the time-to-market of increments.

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

Several components and tools are used to develop and deliver the systems within the Stable and Digital layer:

- A tool to manage the planning and requirements of the various artifacts;
- A Project Management or IT Service Management system;
- A Ticket Management system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Software Quality Assurance tools. This also includes test tools;
- Release and Deployment tools;
- Monitoring and analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

Yes, I can. The mentioned tools are embedded in the following systems:

- ServiceNow is used as a Project Management, IT Service Management, and Ticketing tool;
- Office 365 applications are used for requirements management, planning, functional, and technical specification;
- Teams is used as a collaboration tool;
- For the development and configuration of increments, systems, applications, and software, the following tools are used:
    - SAP NetWeaver and Workbench are used to support the ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - JavaScript;
    - PHP;
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse (CDS and Fiori);
    - Xcode 12 and Swift UI;
    - Android Studio;
- IBM Rational Quality Manager is used as a Test Management Tool;
- The following tools are used for the deployments:
    - Azure Release. To automatically deploy increments from the CI/CD pipeline;
    - The CHARM functionality of SAP Solution Manager is used to deploy the multiple SAP S/4HANA transports;
    - Eclipse is used for SAP iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
- The following tools are used for monitoring and analysis:
    - SAP Solution Manager;
    - Google Analytics;
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

From my end, there is no relationship. In general, the Stable layers' systems are commonly deployed on-premise or in a data center, and the systems of the Digital layer are mainly deployed in the cloud.

#bimodal_IT_layers_Slow-Speed
#deployment_model_on-premise
#deployment_model_data_center
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

The same rule applies for the service-based environments. The deployment model depends on the used middleware technology. So, for example, if an ESB system is used, then the particular middleware system is generally deployed on-premise or in a data center. If a cloud integration platform or microservice environment is used, then the middleware system is probability deployed in the cloud and subscribed as a PaaS or iPaaS solution.

#bimodal_IT_layers_Communication_layer_ESB
#deployment_model_data_center
#deployment_model_on-premise
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#deployment_model_cloud
#deployment_model_cloud_Paas
#deployment_model_cloud_iPaas

At the client, all the systems are deployed in the cloud.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud

Interviewer: Thank you for the answer. However, I still have a question regarding the indicated models, namely, what are the infrastructural building blocks of the indicated deployment models?

When we look at an on-premise or data center environment, a Server Farm is used to realize and offer the systems, applications, and services to a user or organization. The Server Farm comprises application servers, database servers, file servers, exchange servers, web servers, etc., and offers load balancing, data access, file access, data cache, web cache, process scheduling, etc. Primarily for security reasons, the Server Farm servers and services can only be reached and executed via a private network, which is established through a VPN or through the usage of a (Reverse) Proxy server that acts as a Gateway.

#hosting_platform_on-premise_components
#hosting_platform_data_center_components

For a cloud environment applies that the physical servers, network devices, and other hardware, including application servers, database servers, file servers, exchange servers, web servers, etc., are located in a hosting center. All the components are virtualized through container technology and offered via images as virtualized hardware services. Subsequently, the virtualized hardware services are provided as an event-driven serverless compute platform to an organization or user.

The hosting center or cloud provider can offer the following cloud models for deployment:

- Public;

- Private;
- or Hybrid models.

Within these models, an organization or user has the following options to subscribe to the services, systems, and applications:

- Software as a Service (SaaS);
- Platform as a Service (Paas);
- Integration Platform as a Service (iPaaS);
- Function as a Service (FaaS);
- Back-end as a Service (BaaS);
- Infrastructure as a Service (IaaS).

#hosting_platform_cloud_components

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

Cloud is becoming a standard deployment model for all systems. Also, note that the private model is preferred over the public model.

#deployment_model_movement_towards_cloud
#deployment_model_cloud_Private
#deployment_model_cloud_Public

15: What were the major milestones that lead to the current set-up of the IT landscape?

Because the client itself is a hosting provider, there is a continuous focus on automatization. This focus has been the main driver for many decisions within the IT landscape.

#organization_IT_strategy

16: What is the future roadmap of the IT landscape of your organization?

For so far, I know there are no significant plans for the future roadmap.

#organization_IT_roadmap

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Bimodal Architecture will stay a relevant concept in the future because an organization needs stable systems and services on one side. At the same time, they also need to provide and support digital products and services. Even when all the systems are in the cloud, it does not make any difference or impact the Bimodal Architecture approach. We see this also at the client.

Universiteit Leiden

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

The respondent did not answer the question due to time.

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

The respondent did not answer the question due to time.

19: Can you elaborate on the technical debt?

The respondent did not answer the question due to time.

Candidate 15 – EA-06

General questions:

1: Personal information: name, years of experience, and current function/position/role within the company or project.

Currently, I have twenty-two years of experience within the IT domain. Currently, I am fulfilling the role of an Enterprise Architect {EA-06}. I am doing this for the last six years.

#candidate_current_role

To be brief, my journey started in 1998 with programming in Java. In the year 2000, I got the chance to enhance my programming experience by learning SAP ABAP. I did several Java and SAP developments until 2003. In 2003, I got the opportunity to move to integration. Consequently, I fulfilled several roles and performed various integration projects with SAP and non-SAP products until I became an Integration Architect in 2009. In 2013, I decided to broaden my scope by extending my focus beyond integration patterns. Therefore, I made a switch towards the Enterprise domain in 2014.

#candidate_experience

2: General information about the IT organization: the size of the IT capability, are there different internal IT capabilities available (for example, each geographical span has its own IT capability), and what are the core functions that the internal IT capability fulfills.

The organization is operating within the hosting industry {Hosting-01} and is active worldwide. The organizations' focus is on highly automated delivery of servers and services, like hosting, provisioning public/private servers, cloud services, etc. The company has around 500 employees worldwide. Most of the resources work within the internal IT capability to provide and leverage the organizations' various services.

#company_profile
#IT_capability

Bimodal Architecture questions:

1: What means Bimodal Architecture to you? Or what do you think Bimodal Architecture means?

Bimodal means two speeds for developing and delivering IT systems. On one side, the business wants to have flexible IT developments and systems. On the other side, the internal IT organization, mainly the maintenance and support domains, wants to have a stable IT landscape without disturbance. They prefer stable and predictable IT systems. Bimodal Architecture is a concept to manage both modes.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed

2: What are, in your opinion, or can you identify and define the main components of a Bimodal Architecture?

Within a Bimodal Architecture environment, three main components can be identified. The first component is related to a Stable IT layer. The second component is related to a flexible and Explorative IT layer. And the third, and last component, is related to an integration and Communication layer.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer

The Stable environment comprises:

- Critical Information Systems (IS) to achieve and enable business process standardization, efficiency, and operational excellence;
- Systems and applications that are capturing and storing business-critical objects and data;
- Digital applications that are extending specific business processes, improving and increasing the efficiency, providing insights and other organizational information, and adds and delivers operational value;
- Various integration patterns to connect and integrate the systems and applications. The patterns include RFC's, proxies, adapters, EDIFACT standards, APIs, and web- and microservices.

#bimodal_IT_layers_Slow-Speed
#Slow-Speed_stabilized_systems_of_innovation
#Slow-Speed_characteristics
#bimodal_IT_integration_patterns_Slow-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

The Explorative environment comprises:

- Digital systems and applications (mobile, intuitive, certain IoT) to increase and enhance engagement, experience, brand recognition, commitment, and market share;
- Digital systems and applications to improve interaction and enable social channels;
- Systems and applications to experiment and explore new technologies;
- Various integration patterns. The patterns include web services and especially microservices to connect the digital systems and applications.

#bimodal_IT_layers_Fast-Speed
#Fast-Speed_characteristics
#bimodal_IT_integration_patterns_Fast-Speed
#bimodal_IT_integration_patterns_through_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA_services

The Communication environment comprises:

- A service-based environment, such as a service bus, a cloud integration platform, a microservice environment, or even use all three the environments to establish and promote loose coupling and message interchangeability (services);
- Middleware systems and tools to create, adjust, monitor, and manage the various services (APIs).

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_ESB
#bimodal_IT_layers_Communication_layer_ESB_services
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_CIP_services
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_layers_Communication_layer_MSA_services
#bimodal_IT_integration_patterns_through_Communication_layer

The service-oriented environment(s) and middleware systems enable the integration, connectivity, and interoperability between the Stable and Explorative layers' systems and applications. Also, the environment(s), systems, and tools are used internally within the Stable and Explorative layers to establish the various integration patterns between the (digital) systems and applications.

#bimodal_IT_integration_patterns_through_Communication_layer

Presentation of the models (figure 1.1 and 1.2).

3: Can you recognize, identify, or link the presented models to the IT landscape of your organization? Do you recognize similarities or differences between the presented model and the current IT landscape of your organization?

Yes. I can link the presented models to the environments within our IT landscape. However, I like to indicate some differences. First of all, your models miss the internal organizational value-adding applications, which are generally embedded within the Stable layer. This is also applying for the associated integration patterns.
#Slow-Speed_stabilized_systems_of_innovation
#bimodal_IT_integration_patterns_Slow-Speed

The second point that I like to highlight is related to a digital applications' maturity level and moving the solution from the Explorative layer towards the Stable layer. I support the fact that an API can become stable over time. However, the digital application that is invoking the particular API can remain in the Explorative layer. The reason is that the digital solution is created for a certain purpose. Once the purpose changes or becomes obsolete, we still want to be able to change the specific solution in an agile way because of the changed requirements.

#bimodal_IT_software_movement_from_Fast-Speed_to_Slow-Speed

Explanation Slow-Speed and Fast-Speed domain:
Systems and software within the Slow-Speed domain are not required to change often, while in the Fast-Speed domain, the systems and software are subjected to rapid and frequent changes to meet customer, user, or supplier demands.

4: Can you mention which system/applications from your current IT landscape are typically incorporated into the Slow-Speed domain, and which are typically incorporated into the Fast-Speed domain?

The following systems and application are traditionally incorporated in the Stable layer:

- SAP S/4HANA;
- SAP Fiori applications;
- SAP Service Cloud;
- ServiceNow;
- IBM Rational Quality Manager;
- Internal web Apps;
- Internal mobile Apps;
- Product database (custom application);
- Auto-install images (custom applications);
- Specific IoT solutions;
- Multiple internal APIs.

#Slow-Speed_systems_and_software
#Slow-Speed_stabilized_systems_of_innovation

The following systems and application can be found within the Explorative layer:

- Sitecore;
    - Customer Portal;
    - Employee Portal;
    - Supplier Portal;
- Grafana;
- Office 365;
- E-commerce;

- Mobile Apps
    - iOS Apps;
    - Android Apps;
- Web Apps;
- Power Apps;
- External APIs.

#Fast-Speed_systems_and_software

5: How do you decide on what is part of the Slow-Speed or Fast-Speed domain? Can you elaborate on the characteristics of the systems/applications within the Slow-Speed domain, and also on the characteristics of the Fast-Speed domain?

The main characteristics of the Stable layer are:

- Stable, robust, and reliable;
- Security is essential and an ongoing topic;
- The IT systems are supporting the business processes and store and manage the business-critical objects and data;
- The IT systems do not change a lot once they are deployed into production;
- A Minimum of errors are allowed;
- Delivers operational value;
- Makes use of agile and traditional approaches to develop and deliver a particular IT system;
- Fail-over strategy in place to guarantee business continuity and high availability;
- Implementation costs are high;
- IT systems are not easily replaceable due to their importance.

#Slow-Speed_characteristics

The main characteristics of the Explorative layer are:

- Features are small and independent;
- Delivers business value at each release or iteration and ends at a given moment;
- Off-line capability is a necessity to ensure 24/7 availability and for a high experience;
- An IT system can contain business-critical data, but this is not required;
- The IT systems are easily replaceable;
- Less focus on Risk Management. Errors are already factored in;
- Focus on experimentation and exploration;
- Only an Agile way of development is supported;
- Rapid delivery oriented. Often and frequent deployments;
- Continuous integration/deployment in place to increase time-to-market.

#Fast-Speed_characteristics

Explanation Communication layer:
The Communication layer connects/integrates the Slow-Speed domain with the Fast-Speed domain.

6: In case of a communication/integration layer, such as a service-based architecture (for example, an SOA or MSA), what are the essential components or features of this layer? And why?

The Communication layer can comprise one or more service-based environments to establish connectivity, integration, and communication. In our IT-landscape, we have two service-based environments available to realize and facilitate the multiple integration patterns. On one side, we have a cloud integration platform in place to create, transform, route, and consume the services. And to realize and leverage the microservices, we have a microservice environment in place.

#bimodal_IT_layers_Communication_layer
#bimodal_IT_layers_Communication_layer_CIP
#bimodal_IT_layers_Communication_layer_MSA
#bimodal_IT_integration_patterns_through_Communication_layer

We use SAP Cloud Platform (SCP) as a cloud integration platform. This integration platform provides the ability to integrate SAP, non-SAP, cloud, and on-premise applications and processes through services.

#bimodal_IT_layers_Communication_layer_CIP

The (technical) components of SCP are:

- Cloud Integration Service;
  - Message and Event Hub;
    - Repository for Service Discovery and pre-build Integration Flows;
    - Integration/Message Builder;
      - Set-up of Integration Flows;
      - Editing of Flows;
    - System Landscape/Tenant Manager;
    - Transaction Handler;
    - Configuration Manager;
    - Security Manager;
      - Set-up of Message Security;
      - Set-up of Encryption;
    - Access Controller;
      - Set-up of Authorization;
    - Event Handler;
      - Set-up of Events (Event Producer and Event Listener);

        - Enterprise Service Repository;
            - Repository for Enterprise Service Discovery;
        - Message Queues;
        - Exception handler;
        - Logger;
        - Auditor;
    - Connectivity Hub;
        - Repository for Adapter Discovery;
        - Adapter Builder;
            - Set-up of Adapters;
        - Adapter Configuration Manager;
        - Adapter Security Manager;
            - Set-up of Authentication;
        - Adapter Event Handler;
        - Workflow Builder;
            - Set-up of Workflows;
            - Editing of Workflows;
        - Access Controller;
            - Identity Access Manager;
            - Set-up of Authorization;
        - Logger;
        - Auditor;
    - Integration Engine;
        - Adapter Engine;
        - Orchestration Engine;
        - Transformation Engine;
        - Routing Engine;
        - Rule Engine;
        - Publisher and Subscriber Engine;
- API Management Service;
    - Repository for API Discovery;
    - API Designer;
        - Set-up of APIs;
        - Edit APIs;
    - API Configuration Manager;
    - API Test Manager;
- BPM Service;
    - Business Process Manager;
        - Set-up of Business Rules;
        - BPEL;
    - Workflow Builder;

            - Set-up of Workflows;
            - Editing of Workflows;
        - Business Activity Monitor;
- Gateway Service;
    - Traffic Manager;
    - Exception Handler;
    - Policy Manager;
    - Version Manager;
    - Call Handler;
    - Logger;
    - Auditor.

#Communication_layer_CIP_components

SCP realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Service Discovery;
    - Call handling;
    - Message Tracking and Throttling;
    - Rate Limits;
    - Traffic Management;
    - Exception Management;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;

    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - OData;
    - XML;
    - FTP and SFTP;
    - Flat File;
    - SMTP;
    - Database Adapter;
    - 3rd Party Adapters;
    - Custom Adapters.

#Communication_layer_CIP_capabilities

As already indicated, we use a microservice environment to realize and leverage the microservices. Azure is deployed as a cluster and comprises a container orchestration platform and an API Gateway. The cluster contains a set of nodes, which on his turns hosts a set of Pods, which represents a set of running containers (microservices).

#bimodal_IT_layers_Communication_layer_MSA

Azure contains the following (technical) components:

- Scheduler;
- Controllers;
- Key-value Database;
- API Gateway;
- Resource Manager;
- Container Orchestration Engine;
- Storage;
- Queues;
- Container Registry;
- Logs;
- Security Manager;
    - Infrastructure;
    - Containers.

#Communication_layer_MSA_components

The platform realizes the following capabilities and functionalities:

- Container Orchestration Platform;
    - Container Orchestration;
    - Container clustering;
    - Image Discovery;
    - Container Security;
    - Network settings;
    - Key Vault;
    - Load Balancing;
    - Scheduling;
    - Rollouts and Rollbacks;
    - Self-Healing;
    - Workflow Management;
- API Management;
    - Service Discovery;
    - Messaging;
    - Routing (of services);
    - Call handling;
    - Transformation;
    - Rate Limiting;
    - Traffic Management;
    - Throttling;
    - Caching;
    - Circuit Breaker;
    - Fail-over;
    - Microservice Security;
    - Configuration Management;
    - Networking/IP;
    - Analytics;
    - API Version Management;
    - Monitoring and Logging;
    - Policies and Contracts;
- Containers;
    - Microservices.

#Communication_layer_MSA_capabilities

Interviewer: Thank you for the extensive description of the components. However, can you also provide the components of an ESB to make the list complete?

Yes, of course I can. I will take SAP PO as an example.

The (technical) components of SAP PO are:

- Integration Builder;
    - Enterprise Service Repository;
        - Creation and Discovery of Services;
    - Integration Directory;
        - Set-up of Mediation Flows;
- Integration Server;
    - Adapter Engine;
        - Transformations;
    - Integration Engine;
        - Orchestration;
        - Routing;
        - Publish and Subscriptions;
    - Business Process Engine;
        - Rules;
- System Landscape;
    - System Landscape/Tenant Manager;
- Configuration and Monitoring;
    - Configuration Manager;
    - Web Service Manager;
        - Authentication;
        - Message Security;
        - Encryption;
    - Mediation flow Manager;
    - Event Handler;
    - Adapter Manager;
    - Access Controller;
        - Authorization;
    - Exception handler;
    - Logger;
    - Auditor;
- Business Activity Monitoring;
- Business Process Manager and Process Automation;
    - Workflow;
    - Rules;
    - BPEL.

#Communication_layer_ESB_components

SAP PO realizes the following capabilities and functionalities:

- Operations and Management;
    - Statistics and Status;
    - Alerts;
    - Failover;
    - Configuration Management;
    - Deployment;
    - Load Balancing;
    - Service Registry;
    - Message Tracking and Throttling;
    - Exception Management;
- Mediation;
    - Transformation;
    - Protocol Translation;
    - Caching;
    - Message Enrichment;
    - Dynamic Routing;
    - Message Validation;
    - Reliable Messaging;
    - Pass-Through of Messages;
- Security;
    - Authentication;
    - Authorization;
    - Encryption;
    - Certification;
    - Message Security;
- Transportation;
    - HTTP and HTTPS;
    - SOAP;
    - REST;
    - XML;
    - FTP and SFTP;
    - SMTP;
    - Flat File;
    - Database Adapter;
    - 3rd Party Adapter;
    - Custom Adapter.

#Communication_layer_ESB_capabilities

7: Which (different) Software Delivery Life Cycle (SDLC) methodologies are used with the IT landscape of your organization?

The systems and applications that are incorporated within the Explorative layer are realized, delivered, and exploited via agile approaches. These approaches can be Scrum, CI/CD, or even Extreme Programming. However, in our IT landscape, CI/CD is foremost the preferred method.

#bimodal_IT_layers_Fast-Speed
#methodology_agile_framework
#methodology_agile_framework_continuous_integration_and_delivery

For the systems and applications that are incorporated within the Stable layer applies that some solutions are delivered through an agile approach, while the majority of the IS's, which have a certain complexity, are delivered through a mix of approaches. The approaches that are particularly used within this environment are:

- PRINCE2;
- Waterfall;
- V-Shape;
- Kanban;
- or Scrum.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework

8: What is the impact of a Bimodal Architecture environment on your software delivery/methodology?

It is important to keep in mind that automation is a crucial aspect in one of the environments. This means that the build, test, publish and deploy steps can be performed more quickly than the other environment. The IT organization needs to be able to deal with these two speeds regarding the release of increments. Therefore, release management, planning, and communication are crucial within a Bimodal Architecture environment and significantly impact the software delivery.

#impact_release_management
#impact_communication
#impact_planning

9: Do you use different methodologies across the different layers?

Yes, the Stable layer uses a mix of traditional and agile SDLC approaches. While the Explorative layer is only supported by agile methodologies.

#bimodal_IT_layers_Slow-Speed
#methodology_traditional_framework
#methodology_agile_framework
#bimodal_IT_layers_Fast-Speed

#methodology_agile_framework

10a: In case of Continuous Integration/Continuous Delivery (CI/CD), how is this methodology, or can it be, embedded within the Bimodal Architecture environment?

CI/CD is used for the delivery of systems and applications within the Explorative layer, as well as for the delivery of microservices.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10b: Which layers are affected by CI/CD?

Currently, only the Explorative layer and the microservices environment are affected by CI/CD.

#bimodal_IT_layers_Fast-Speed
#bimodal_IT_layers_Communication_layer_MSA
#methodology_agile_framework_continuous_integration_and_delivery
#continuous_integration_and_delivery_adoption

10c: What are the application components of the CI/CD environment? And in which applications/system software are those application components realized?

The development of increments starts locally by using an Integrated Development Environment (IDE) or Development Kit (DK). Once the coding is finished, the source code is submitted into a Repository, in our case Git. After this, the Azure DevOps tool comes into play to build and deploy the software packages (increments).

#software_delivery_life_cycle_components
#continuous_integration_and_delivery_components

The DevOps tools contains the following functionalities:

- Azure Boards. To create, manage and maintain the Backlog, User Stories, Sprints, Test Scripts, Work items, and Bug tracking;
- Azure Repos. Azure Repos is a Version Control System and uses Git as a Repository;
- Azure Pipeline. Azure Pipeline is an out-of-the-box automation server. However, this functionality can also be integrated with Jenkins and other third-party automation servers. The Azure Pipeline functionality is used to automate the build, test, and publish activities;
- Azure Release. The release functionality is used to automate the deployment step. Azure DevOps can use Azure Functions, Azure Service Fabric, Azure Kubernetes Service, or Docker containers as a deployment technology;
- Azure Monitor. This tool is used for monitoring and analysis.

Postman is used to test and mock the various APIs.

#continuous_integration_and_delivery_systems_and_software
#continuous_integration_and_delivery_systems_and_software_functionalities

11: In case of a CI/CD pipeline, what are the implications or common configuration practices for each layer?

The common configuration practices for CI/CD are:

- A CI/CD strategy needs to be in place;
- A recovery strategy should be in place;
- A proper branching structure should be in place;
- Increments should be small. This minimizes the branches;
- Collaboration needs to be encouraged. This also impacts the sharing and reuse of existing software packages;
- The developer should test the increment rigorously on his/her local environment;
- Within the pipeline, the proper tests should be configured and kicked off;
- A verification step should be in place before each deployment;
- Use of a Version Control System needs to be encouraged;
- Use of an Automation Server that can automatically build, test, publish and deploy the various increments;
- Provide KPIs to measure the pipeline executions;
- Monitoring and analytics need to be in place to measure and analyze the solutions.

#continuous_integration_and_delivery_practices

12: Which other tools or components are used to realize and facilitate the Bimodal IT environment?

Several components and tools are used to develop and deliver the systems and applications for the Stable and Explorative layers:

- A tool to manage the planning and requirements of the various artifacts;
- A Project Management or IT Service Management system;
- A Ticket Management system;
- Functional and technical design tools;
- Collaboration tools;
- A Document Management System;
- Several DKs and IDEs to develop, change, and configure systems, programs, applications, features, and APIs;
- Software Quality Assurance tools. This also includes test tools;
- Release/Deployment tools;
- Monitoring and analytics tools.

#software_delivery_life_cycle_components

13: How are these tools or components implemented? Can you also name the actual applications where these components are embedded in?

The components and tools are implemented through the following systems and applications:

- ServiceNow and Azure Boards are used to manage the planning, backlog, and sprints. The ServiceNow system is also used for Project Management and IT Service Management activities and ticketing;
- Tracecloud is used for requirements management;
- Multiple Office 365 applications are used to deliver the functional and technical specifications;
- Teams is used for collaboration;
- SharePoint is used to store all the documentation;
- For the development and configuration of increments, systems, applications, and software, the following tools are used:
    - SAP NetWeaver and Workbench are used to support the ABAP developments;
    - SAP Implementation Guide to configure the SAP system;
    - SDK (SAP Development Kit);
    - JDK (Java Development Kit);
    - JavaScript;
    - PHP;
    - Visual Studio (.NET, Python, C/C++ and Node.js developments);
    - Visual Studio extensions (PowerShell and YAML);
    - Eclipse (CDS and Fiori);
    - Xcode 12 and Swift UI;
    - Android Studio;
- We are using the following Software Quality Assurance tools:
    - Azure Test Plan;
    - IBM Rational Quality Manager. To create test plans and test scripts (traditional and agile development);
- We are using the following tools for deployment:
    - Azure Release. To automatically deploy increments from the CI/CD pipeline;
    - The CHARM functionality of SAP Solution Manager is used to deploy the multiple SAP S/4HANA transports;
    - Eclipse is used for SAP iFlow creation and deployments;
    - To deploy Java software, we are making use of the Java Deployment Toolkit;
- The following tools are used for monitoring and analysis:
    - SAP Solution Manager;
    - Google Analytics;
    - Grafana.

#software_delivery_life_cycle_systems_and_software
#software_delivery_life_cycle_systems_and_software_functionalities

14a: Is there a relationship between the deployment/hosting model and Bimodal Architecture?

No. There is no relationship from my perspective. The organization itself is a cloud hosting provider. Therefore, all our systems and applications are deployed into the cloud. To be more specific, most systems and applications are deployed in a private cloud environment. This applies to the Stable as well as for the Explorative layer.

#bimodal_IT_layers_Slow-Speed
#bimodal_IT_layers_Fast-Speed
#deployment_model_cloud
#deployment_model_cloud_Private

14b: If yes, what do you consider as a common deployment practice for each layer, or how are the layers deployed within your current IT architecture?

Cloud deployments models are becoming more prominent. When we look to other companies and businesses, we can determine a trend towards the cloud.

#deployment_model_movement_towards_cloud

15: What were the major milestones that lead to the current set-up of the IT landscape?

As an organization, we need to focus on automatization and how to improve our services. Therefore, this focus was, is, and will be the primary driver for many decisions within the IT landscape.

#organization_IT_strategy

16: What is the future roadmap of the IT landscape of your organization?

At this moment, there are no significant changes planned for the roadmap. Due to the COVID-19 pandemic, we are currently expanding our services to other countries to gain more market share.

#organization_IT_roadmap
#organization_IT_strategy

17: How do you see the future of Bimodal Architecture/Two-Speed IT?

Flexibility will become more important. I also believe that modular and flexible systems and applications will replace the core information systems. However, even in this scenario, we still need to distinguish between a Stable layer and an Explorative layer because the need for stable components, like databases, will remain.

#concept_bimodal_IT_future
#concept_bimodal_IT_relevancy

Other questions (when time is left):

18a: Which impact has a Bimodal Architecture environment on roles, team set-up, processes, and KPIs?

Bimodal Architecture requires a different team set-up to support the various layers. The best practice for the team set-up should follow the principles as indicated within the agile manifesto. However, the business itself should also change to commit to the agile way of working.

#methodology_agile_framework
#concept_bimodal_IT_agile_teams

18b: How is the priority setting, budgeting, and contracting done in case of Bimodal IT?

For the Stable layer applies in general that the business sets the priority for the several systems and applications. Before the build can start, a detailed impact analysis must be provided and outline the IT projects' risks and costs. Once a steering community has approved the estimations, the IT project can start with its development and delivery.

#bimodal_IT_layers_Slow-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_estimations

In the Explorative layer, the priority is set by the PO. The costs of the associated agile teams are fixed and based on time and material.

#bimodal_IT_layers_Fast-Speed
#concept_bimodal_IT_prioritization
#concept_bimodal_IT_agile_teams

19: Can you elaborate on the technical debt?

The technical debt is minimal in the Explorative layer. The reason is that the systems and applications are built in Java, .NET, C/C++, or Node.js and invoke a subset of services. However, if there are any issues or problems, it can be fixed very fast, mostly in another sprint.

In contrast, the Stable layers' systems and applications have a high technical debt because they contain custom build functionalities. Also, there is an urgent need to expose the various functionalities as services to realize and facilitate the API platform.

#concept_bimodal_IT_technical_debt

Appendix 14 – Feedback interviews

Candidate 16 – IA-02

Questions:

1: What is your first impression when looking at the Reference Architecture?

When you sent over the models, I mostly wondered what the expected goal or added value of the diagrams would be. Does the Reference Architecture capture the whole IT landscape and its related integration patterns or just a part of it, e.g., only the IT systems that are communicating through services? Are the diagrams targeted to enterprise architects, project managers, or is it intended for solutions architects? Because from an integration perspective, I saw many diagrams, but all these diagrams are very high-level.

However, after talking with you, I got a better impression of the Reference Architecture and its purpose. From an integration perspective, the diagrams are clear and seem to be complete and applicable. We have SAP Cloud Platform, Copado, and AWS in place to connect the various IT systems within our landscape. Next to these patterns, we also use point-to-point and SFTP patterns to send and receive data between the different IT systems.

#RA_is_clear
#RA_looks_complete
#RA_is_applicable
#integration_patterns_are_beyond_services

2: What is your impression when looking at the different layers and components? Do you recognize those, or do you miss any elements?

I recognize the service-based environments and their components. Also, I don't miss any components. Although, an explanation would be nice to differentiate some of the components, like the Service Mesh and the API Gateway within the Cloud Orchestration Platform diagram. Many people don't know that both components are important for the communication within a Microservice Architecture and, therefore, need to collaborate.

#recognizes_all_elements
#does_not_miss_any_critical_elements
#prefers_explanation_of_essential_components

Also, another concern from my end is that not all the communication patterns are based on services. We have web apps in place that communicates through SFTP or point-to-point integration with the back-end IT systems. An example is a template app, which is used to create templates by the customer to print the letters in a particular format. Or a web app to run and support the sorting system. In this case, the connectivity is established through point-to-point integration because we do not want to expose this pattern outside a particular business unit.

#integration_patterns_are_beyond_services

3: Looking at the Reference Architecture and the experiences in your current or past projects, in which environments/projects do you see a match? Which environments/projects would benefit from this Reference Architecture?

First of all, it is interesting how you visualize the Communication Layer because this is not done before so far, I know.

#visualization_of_patterns

Regarding a match, I see indeed a match for all the projects that we are or will conduct in the upcoming future. For instance, we have created multiple Track and Trace functionalities. One API is created for the internal customer care agents, while the other two APIs and associated web apps are created for the customers to trace their parcels. In this case, the APIs and associated web apps, back-end IT systems, and integration patterns could be modeled through the Reference Architecture.

#visualization_of_patterns

Ultimately, I believe that the provided Reference Architecture can be used in any IT project to expose the communication patterns between the front-end and back-end IT systems.

#RA_is_applicable_in_a_project

4: Can the Reference Architecture be helpful or even act as a framework for the current architectural views and models? For example, to create better views and models because the Reference Architecture can help structure and visualize the IT landscape?

The Reference Architecture can support multiple purposes. From an enterprise-wide perspective, the provided diagrams can help to visualize the entire IT landscape. And from a project perspective, it can provide guidelines and show the multiple integration components and patterns that are used to connect the various IT systems.

#RA_is_applicable_on_a_enterprise-wide_level
#RA_is_applicable_in_a_project

5: Where do you see advantages of this model? Where do you see potential challenges or drawbacks?

An advantage of the Reference Architecture is that it can be used to explain the different integration patterns and why we need certain service-based environments within the IT landscape. Hopefully, this will also prevent the next time that a manager asks us to phase out one of the most vital IT systems in the landscape because he/she doesn't know why we need that particular (middleware) IT system, such as SAP PO.

#RA_can_support_in_analysis_and_discussions

6: Any other remarks/thoughts you would like to share?

No, not from my end.

I want to advise you to discuss this Reference Architecture with the enterprise architect. As already indicated, the Reference Architecture is interesting and can be helpful to us regarding the visualization of missing patterns. Also, it could be a nice add-on to our existing diagrams.

#RA_is_applicable
#RA_can_help_to_create_better_views_or_complete_existing_views

Candidate 17 – EA-07

Questions:

1: What is your first impression when looking at the Reference Architecture?

My first impression regarding the Reference Architecture was that it provides a good proposal on how the IT landscape could be visualized. Looking at the models and their details, I can model all the systems of our IT landscape and verify if we are using patterns associated with a Bimodal IT environment. The Reference Architecture can also help our department uncover hidden patterns, such as systems that are classified as "Explorative" but should be embedded in the system category "Operational value-adding and Stabilized Systems of Innovation." For instance, the Fiori Apps that we are using on top of SAP S/4HANA.

#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture
#disclose_the_usage_of_Bimodal_IT_patterns
#visualization_of_patterns
#RA_helps_to_distuigish_between_system_categories

2: What is your impression when looking at the different layers and components? Do you recognize those, or do you miss any elements?

Yes, I recognize the different layers, environments, and components. However, regarding the components within the service-based environments, I must admit that I cannot say with 100% guarantee if the models are complete because I have only a high-level overview of the used middleware within our landscape. Therefore, it is hard for me to say if anything is still lacking within these particular models. But, regardless of my knowledge in these specific systems, I do not miss any components. So, from that end, the models seem to be reliable and correct.

#recognizes_all_elements

3: Looking at the Reference Architecture and the experiences in your current or past projects, in which environments/projects do you see a match? Which environments/projects would benefit from this Reference Architecture?

Looking at the Reference Architecture, I think that the whole IT landscape and all upcoming projects can benefit from this visualization. The reason is that we currently do not distinguish between the fast-changing and stable systems or components within our models. For instance, we have on one side Salesforce and Apttus for our Customer and Contract Management and SAP S/4HANA as our back-end. Also, we have created a web app for our Sales Representatives that shows data from Salesforce, Apttus, and as well as SAP S/4HANA. In this scenario, our current models just show that these systems are related to each other but do not cover how they are connected. Also, I agree with you that we can treat Salesforce and Apttus as a business-critical system because they hold the Customers and Contracts and, therefore, could be embedded in the same category as SAP S4/HANA. Although, the configuration can be done much faster in Salesforce in comparison to SAP S4/HANA.

#RA_is_applicable_in_a_project

#visualization_of_patterns
#RA_helps_to_distuigish_between_system_categories

4: Can the Reference Architecture be helpful or even act as a framework for the current architectural views and models? For example, to create better views and models because the Reference Architecture can help structure and visualize the IT landscape?

As mentioned in the previous question, I believe that the Reference Architecture can be quite helpful. We already have multiple diagrams regarding our IT landscape and systems. By adding the details presented in your Reference Architecture, we can create a new dimension into our own models and views.

#RA_is_applicable
#RA_can_help_to_create_better_views_or_complete_existing_views

5: Where do you see advantages of this model? Where do you see potential challenges or drawbacks?

This Reference Architecture can help us to visualize the IT landscape in another way. Also, it can help us to create different outsourcing strategies for each individual environment.

#visualization_of_patterns
#RA_can_help_to_create_outsourcing_strategie_for_each_system_category

The only drawback is that we have to find a way to embed your Reference Architecture patterns in our models without starting from a blank sheet. I have to discuss this with my colleagues.

#concern_regarding_adoption_of_RA_due_to_the_creation_of_multiple_views

6: Any other remarks/thoughts you would like to share?

No, not at this moment. You have created an excellent baseline on how an organization can model its IT landscape and visualize a Bimodal IT environment.

#RA_can_be_used_as_a_baseline

Universiteit Leiden

Candidate 18 – EA-08

Questions:

1: What is your first impression when looking at the Reference Architecture?

The first thing that I noticed is the focus of the Reference Architecture. The views are trying to capture an organizations' entire IT landscape and fit it into a Bimodal IT environment instead of highlighting when it is appropriate to use an agile or traditional approach. It seems that you like to disclose that a Bimodal IT is a concept and an architecture, which is a very interesting thought when you ask it me.

#RA_visualizes_a_Bimodal_IT_environment
#Bimodal_IT_is_also_an_architecture_rather_than_only_a_concept

When I purely look from an architectural point of view, the Reference Architecture looks clear. It presents an excellent overview of how an architecture can capture, allocate, and model the multiple systems used within a single organization. Also, the layers allow to drill down into specific system domains to reveal their implemented systems and related integration patterns.

#RA_is_clear
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture
#visualization_of_patterns

In addition, when comparing the provided views with current views created by the clients, I must say that the Reference Architecture provides a decent guideline on how to model an IT landscape.

#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture

2: What is your impression when looking at the different layers and components? Do you recognize those, or do you miss any elements?

I recognize all the elements and components. Also, I don't miss any crucial elements within the provided views.

#recognizes_all_elements
#does_not_miss_any_critical_elements

However, I would like to make two comments on the Reference Architecture. First of all, it would be nice to create some additional or detailed views that can actually support the business processes instead of only providing the architecture point of view. And the second point relates to the explanation of certain components. For instance, you have mentioned in one of the views the components Service Mesh and API Gateway. Many people don't know the difference between these components. The explanation is especially important when certain components are essential to realizing a certain pattern or environment, which is the case for both the mentioned components.

#detailed_views_required_within_the_RA_to_link_processes
#prefers_explanation_of_essential_components

3: Looking at the Reference Architecture and the experiences in your current or past projects, in which environments/projects do you see a match? Which environments/projects would benefit from this Reference Architecture?

From my perspective, every organization and all its upcoming projects can benefit from the Reference Architecture.

#RA_is_applicable_in_a_project

The Reference Architecture can tell a story from an architectural point of view. For instance, how out-systems (web-apps) are linked to the back-end systems, like a CRM system. Or how a stand-alone DMS is linked to an ERP system. The views present which systems are placed in which layer and system domain and reveal which patterns are used to establish the connectivity between the systems.

#visualization_of_patterns
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture

However, I believe that it would be nice to create some additional views regarding the data flows. These views could complete the story. For instance, a view could tell how customers and quotations are created in out-systems (web-apps) and replicated to a CRM system and vice versa. Or how attachments from an external DMS are linked to cases and projects in an ERP system. And, last but not least, show how requests could lead to an addition of attachments into the DMS. These views will help to bridge the gap between the architectural point of view and the business processes used by a specific organization.

#detailed_views_required_within_the_RA_to_link_processes

4: Can the Reference Architecture be helpful or even act as a framework for the current architectural views and models? For example, to create better views and models because the Reference Architecture can help structure and visualize the IT landscape?

Yes, I believe the Reference Architecture can be very helpful in setting up a proper architecture. The layers and associated categories can also help an organization to differentiate between digitized systems and digitalized systems.

#RA_can_help_to_set-up_an_architecture
#RA_helps_to_distuigish_between_system_categories

Consequently, the Reference Architecture can also help indicate if a particular organization is already using the patterns of a Bimodal IT environment or help set-up and implement a Bimodal IT environment within the organizations' IT landscape.

#disclose_the_usage_of_Bimodal_IT_patterns
#RA_can_help_to_set-up_a_Bimodal_IT_platform

5: Where do you see advantages of this model? Where do you see potential challenges or drawbacks?

The Reference Architecture can be used as a baseline to visualize the entire IT landscape of an organization. On top of that, the views can support in the analysis and discussion when implementing or removing a system because it can show the dependencies with other systems.

#RA_can_be_used_as_a_baseline
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture
#RA_can_support_in_analysis_and_discussions

Another important point is that the views need to be maintained on a frequent base because it captures the entire IT landscape. This activity will also reduce the resilience to use the views by the business and project members.

#concern_regarding_adoption_of_RA_due_to_maintenance_of_RA

A drawback for the current Reference Architecture could be that there is no alignment between the business processes and the IT landscape.

#concern_regarding_adoption_of_RA_due_to_IT_business_alignment

6: Any other remarks/thoughts you would like to share?

I want to take the Reference Architecture one step further. It would be good to plot the IT landscape of an organization into the provided views. After completing this activity, all the relevant detailed views should be created and subsequently linked to the business processes of a specific organization. In this case, the architecture can help change or improve the processes by highlighting the systems, parts of the communication layer, and components related to a specific process. Conclusively, the views can help optimize the IT landscape, improve the analysis, mitigate risks, and help the business understand how the different systems are related to the processes and vice versa.

#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture
#detailed_views_required_within_the_RA_to_link_processes
#remarks_to_improve_the_RA

Candidate 19 – IA-03

Questions:

1: What is your first impression when looking at the Reference Architecture?

At first sight, the diagrams look clear. What I appreciate about the Reference Architecture is that it showcases in a unique way how the systems are embedded within an IT landscape and how they are communicating with each other. I cannot remember that this is done before or attempted by someone else.

#RA_is_clear
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture

2: What is your impression when looking at the different layers and components? Do you recognize those, or do you miss any elements?

Yes, I recognize the layers and their components, especially those within the service-based environments.

#recognizes_all_elements

Maybe one additional comment from my end, not all the communication patterns are solely based on services. Occasionally, a web app is connected directly, through another mechanism instead of services, with the back-end because services are not appropriate for certain scenarios. An example can be a Mendix App that is connected through SFTP with a DMS. Or certain critical applications that require a point-to-point integration with the back-end systems. If these connections are realized through services, the change exists that the services could expose critical data outside the organizational boundaries, which can have a big impact if it ends up on the streets, like intelligence data or personal data.

#integration_patterns_are_beyond_services

3: Looking at the Reference Architecture and the experiences in your current or past projects, in which environments/projects do you see a match? Which environments/projects would benefit from this Reference Architecture?

Yes, I see a match for all the digitalization projects that are being or will be conducted by an organization. For instance, here at my current client, we are creating multiple mobile- and web apps. However, all these systems are retrieving and updating their data within the back-end systems. All these mobile apps, web apps, associated APIs, back-end systems, and integration patterns could be modeled through the Reference Architecture. Conclusively, the provided Reference Architecture can be used in any IT project to expose the integration patterns from all the related front-end and back-end systems.

#RA_is_applicable_in_a_project
#visualization_of_patterns
#RA_provides_a_proposal_or_guideline_to_visualize_an_architecture

Universiteit Leiden

4: Can the Reference Architecture be helpful or even act as a framework for the current architectural views and models? For example, to create better views and models because the Reference Architecture can help structure and visualize the IT landscape?

The Reference Architecture can be quite helpful because, in many cases, at a client, the diagrams are lacking the integration components and patterns. By adding these patterns into their diagrams, they can better understand how the different systems are related to each other. Also, the Reference Architecture can help us explain and discuss how to improve or upgrade a specific IT landscape to support the digitalization journey.

#RA_is_applicable
#RA_can_help_to_create_better_views_or_complete_existing_views
#RA_can_support_in_analysis_and_discussions

5: Where do you see advantages of this model? Where do you see potential challenges or drawbacks?

Through the Reference Architecture, everyone will be forced to think about the interactions between each system and component. It will also support the usage of an external and internal API platform and help determine if existing communication patterns are efficient and sustainable or need to change.

#RA_can_support_in_analysis_and_discussions
#RA_supports_a_seperate_external_and_internal_API_platform

The reference architecture can help an enterprise architect determine how and where a new system should be placed in to the landscape and which software development approaches would be appropriate to develop it.

#visualization_of_patterns
#RA_helps_to_distuigish_between_system_categories

Ultimately, the Reference Architecture can support the creation of enterprise-wide integration patterns and set-up principles for each system domain to build sustainable increments and connect these properly. And it can help in the discussion and decision process regarding the future of an IT landscape. So, from that end, the Reference Architecture has a lot of potentials.

#RA_can_help_to_set-up_an_architecture
#RA_can_support_in_analysis_and_discussions

6: Any other remarks/thoughts you would like to share?

No, the Reference Architecture looks solid. By adopting the patterns of the Reference Architecture into the current views and diagrams, an organization can have an excellent baseline to visualize its enterprise-wide architecture.

#RA_can_be_used_as_a_baseline
#RA_can_help_to_create_better_views_or_complete_existing_views