Universiteit Leiden
ICT in Business and the Public Sector

Master Thesis Project:

From infrastructure to platform: A study on factors influencing migrations to the Cloud

Name: Jamal Thijssen
Student number: S2454971

Date: September 5th , 2023

1st supervisor: Drs. Bas Kruiswijk
2nd supervisor: Dr. Werner Heijstek

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Introduction

Cloud Computing refers to the distribution of various computing services over the internet. These services are frequently offered by cloud providers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure (Li et al., 2010).

The three main delivery models in cloud computing are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software as a Service (SaaS). Organisations still tend to be using IaaS solutions, rather than embracing the more wide-ranging possibilities offered by PaaS. The reasons why organisations may still be using the early cloud model, IaaS, may depend on factors such as control over the infrastructure, noticeable complexity, lack of technical expertise and vendor lock-in (Martins et al., 2016).

Objective

This study explored the factors influencing an organisation's transition from IaaS to PaaS. Specifically, the study aims to provide insights into the motivations, challenges, and potential benefits of such a migration.

Method

In order to provide a broad understanding of the research topic, the study adopted a qualitative approach. Alongside looking into existing literature, interviews with cloud experts, and a detailed case study were conducted. The literature review provided a foundation for identifying key themes, challenges, and best practices regarding transitions to the cloud. This led to the identification of a research gap and formulation of the research questions.

Building upon the literature review, nine (9) in-depth interviews were conducted with cloud experts whose areas of expertise surrounded a deep understanding of cloud architectures, industry trends and migration strategies.

Complementing the expert interviews, the case study involved an Independent Software Vendor (ISV) planning on transitioning from IaaS to PaaS. This case served as a realistic setting for demonstrating the real-world effects of a transition from IaaS to PaaS.

Results

The main benefits of the transition to PaaS were found to include cost optimisation, enhanced scalability, and improved developer productivity. However, potential risks were also identified, such as learning curve, Data migration and multi-cloud architectures.

Among the notable technical requirements that surfaced the implementation of Continuous Integration/Continuous Deployment (CI/CD) pipelines, and the strategic leverage of Infrastructure as Code (IaC) were particularly pronounced.

The final part of the results regarding the architectural changes two architectural transformations emerged, specifically: breaking down a monolithic application into modular components and re-architecting the codebase to better suit the PaaS paradigm.

Conclusion

It is important to recognize that while the results aim to provide a solid starting point for organisation that intend to transition to PaaS, their applicability and significance are dependent upon the unique context of each business. Future research should consider broadening the scope of the investigation through additional case studies.

Contents

1 Introduction

In recent years, the cloud computing market has experienced significant growth where early cloud computing (CC) offerings were largely focused on Infrastructure as a Service (IaaS) models (Bello et al., 2021). Under the IaaS model customers were responsible for managing their own operating systems, storage, and applications (Mohammed and Zeebaree, 2021). However, as the market has matured, there has been a shift towards Platform as a Service (PaaS) models, where providers offer more completely managed environments for developing, deploying, and scaling applications (Cohen, 2013). While IaaS gives businesses flexibility and control, it also forces them to spend time and resources on infrastructure management diverting their focus towards the core business and access to the latest technologies (Wulf et al., 2021). Wulf et al. (2021) reveal in their article the factors influencing the implementation of CC at the information system (IS) based on the chosen delivery model.

Also explained by Wulf et al. (2021) is that each cloud delivery model offers distinct features and benefits that satisfies different use cases. For example, PaaS offers ready-to-use development tools and environments allowing developers to focus solely on writing code and building applications. This cloud solution is desired for businesses seeking to accelerate development cycles and reduce the complexity of managing infrastructure components (Cohen, 2013). This consequently reduces operating expenses by outsourcing the responsibilities of infrastructure management and maintenance (Gass et al., 2014).

Opposed to IaaS, PaaS solutions offer a more controlled and abstracted environment, enabling businesses to prioritize application development and innovation (Cohen, 2013). Organisations may, for instance, simplify their processes, spend less time on maintenance, and launch new features and applications faster by outsourcing infrastructure management to a PaaS provider. Amazon Web Services (AWS), that provides fully managed platform capabilities for businesses, is one example of a cloud provider that has established itself as a key player in the CC environment (Dantas et al., 2022).

Early adopters of CC may likely still be using IaaS instead of PaaS, which may lead to missed opportunities in optimising their application development. According to Repschlager et al. (2013) the main obstacles for accepting PaaS were concerns about vendor lock-in and lack of control over the infrastructure. However, as the PaaS market has matured, providers have introduced more flexibility and interoperability options (Cohen, 2013). Over the recent years various strategies and approaches have emerged that mitigate vendor lock-in risks, such as the use of open standards, multi-cloud architectures, and the adoption of containerisation and microservices (Quint and Kratzke, 2016).

This research aims to explore why businesses should consider making the transition from IaaS to PaaS, and the potential advantages and challenges associated with such a shift. This is important for making informed decisions and discover fitting cloud computing strategies.

Finding the optimal approach to transition to cloud technology holds equal importance as recognizing the factors driving this change (Jamshidi et al., 2013). While multiple methods of cloud adoption exist, this study concentrates on the methodologies outlined by Ahmad et al. (2018). These methodologies include the options of Rehosting, Replatforming, Repurchasing, Restructuring, and Rebuilding.

IaaS and PaaS have different architectural philosophies (Mohammed & Zeebaree, 2021, which calls for careful consideration. This may require breaking down monolithic applications into modular components that are suited for PaaS. As seen in the chapter where the case is discussed, it might take careful design and execution to successfully break down monolithic structures while maintaining functionality (Ali et al., 2016).

Service-Oriented Architecture (SOA) provides a conceptual and practical framework for breaking down monolithic applications into modular components that are better suited for cloud adoption, especially when transitioning to specific cloud models as PaaS (NIST, 2009). Along with Event Driven Architecture (EDA), it promotes a design philosophy that supports flexibility, reusability, and interoperability, which are key considerations when making the shift to cloud technologies (Bukhsh et al, 2015).

Finally, the study acknowledges that a planned approach is necessary due to the complex nature of such, transition. The Technology Organizational and Environmental (TOE) framework by Borgman et al. (2013) for an academic perspective and Amazon Web Services Cloud Adoption Framework (AWS CAF) for a practical viewpoint are discussed here to determine which areas require an in-dept analysis.

2 Objectives

Understanding the precise point at which an organisation is ready to transition from IaaS to PaaS is critical to avoid premature or misinformed decisions (Loebecke et al., 2012). While organisations may recognise the potential benefits of PaaS, it remains essential to assess their existing infrastructure, technological capabilities, and business objectives to determine if they have reached a state of readiness (Ogunlolu and Rajanen, 2019).

Transitioning from IaaS to PaaS poses several challenges and complexities that require careful research and analysis of key considerations (Carrasco et al., 2018). One key consideration is the impact on application architecture and design. IaaS and PaaS have distinct differences in terms of the underlying infrastructure and services provided (Wulf et al., 2021). Therefore, organisations need to assess how their existing applications and systems will need to be restructured or modified to take advantage of the platform-specific features and capabilities offered by PaaS.

Another crucial consideration is vendor lock-in. While IaaS provides more flexibility and control over the underlying infrastructure, transitioning to PaaS can introduce a higher dependency on the chosen platform provider. Researching and understanding the extent of vendor lock-in associated with various PaaS offerings is essential. It allows organisations to make informed decisions about the level of flexibility, interoperability, and portability they require, reducing the risk of being locked into a specific PaaS provider without viable alternatives (Opara et al., 2014).

While many studies have explored the benefits and challenges of migrating to IaaS or PaaS (Amin et al., 2021; Wulf et al., 2021; Hosseini et al., 2010), very few have delved into the architectural changes or technical requirements that businesses need to make during this process (Pahl and Xiong, 2013).

Most articles do describe key opportunities and challenges in the context of data migration to the cloud (Amin et al., 2021). However, these scientific publications have a starting point of either local host applications or other on-premises data (Corradini et al., 2015).

Due to the limited available research in this area, organisations may face challenges in fully understanding and anticipating the difficulties of transitions form IaaS to PaaS (Ansar et al., 2018). It is worth noting that a subset of this research is specifically dedicated to businesses that are contemplating such transitions (Opara et al., 2014).

Since cloud computing has become more popular over the years, other common situations occur more frequently such as moving data or applications between cloud vendors (Buksh et al., 2015). As organisations evolve and their requirements change, they may find it necessary to switch from one cloud vendor to another for various reasons, such as cost optimisation, service enhancements, or better alignment with their business objectives.

Furthermore, the lack of information regarding architectural changes and technical requirements can hinder an organisations' ability to make informed decisions and optimise their PaaS implementations (Linthicum, 2017). Without this knowledge, organisations may face compatibility issues, performance bottlenecks, or limitations in fully utilising the platform-specific features and capabilities offered by PaaS providers (Cohen, 2013).

Therefore, organisations must carefully assess their existing infrastructure, evaluate compatibility with PaaS platforms, and plan for potential disruptions during the transition (Pahl and Xiong, 2013).

This research aims to explore the factors that determine an organisation's readiness for transitioning to PaaS. By addressing these considerations, organisations can make well-informed decisions, effectively plan their transition, and harness the benefits of PaaS to drive innovation, efficiency, and competitiveness in the cloud computing landscape.

Giving up control of the infrastructure can be a significant consideration for organisations, as it entails relying on the PaaS provider for these operational aspects (Linthicum, 2017). Therefore, the research will also investigate the differences between IaaS and PaaS and emphasize the trade-offs organisations are required to make.

3 Research question

This study concentrates on the important factors that businesses must take into account while switching from IaaS to PaaS. It focuses on aspects that have been ignored within existing literature, such as technological needs, and architectural distinctions. These aspects, together with the evident benefits and possible risks of the transition to PaaS, play a decisive role in deciding the outcome of such a transition.

To ensure clarity and simplicity for the reader in understanding this topic, the definitions of the cloud computing models are described as followed:

Infrastructure as a Service (IaaS): IaaS is a cloud computing model where organisations rent virtualized computing resources, such as servers, storage, and networking, from a cloud service provider. With IaaS, organisations have control over the infrastructure layer and are responsible for managing and maintaining the virtualized resources, operating systems, and applications (Pogarcic et al., 2012).

Platform as a Service (PaaS): PaaS is a cloud computing model where organisations can develop, deploy, and manage applications without worrying about the underlying infrastructure. PaaS providers offer a platform that includes the necessary hardware, operating systems, development tools, and runtime environments (NIST, 2009). This allows organisations to focus more on application development and deployment while leaving the infrastructure management to the PaaS provider.

3.1 Research question

What are the key factors that should be considered when an organisation migrates from IaaS to PaaS?

3.2 Sub-questions

1. What are the potential risks and benefits of organisations when moving to a PaaS platform?
2. What technical requirements must be met to operate an application on a PaaS platform?
3. What architectural changes do businesses need to make when transitioning from IaaS to PaaS?

4 Related work

This chapter presents a summary of the most appropriate findings from previous research and studies related to the areas of cloud computing (CC), migration strategies, and architectural methods and tools. The objective here is to inform readers about the most significant findings collected in each area, which also sets the fundament for future research and understanding.

4.1 Cloud Computing

The term cloud computing has been widely discussed and simplified by scholars, including Wang et al. (2008), Bello et al. (2021) and Saraswath and Tripathi (2020). The way these scholars describe the term "cloud computing" is often derived from the definition provided by the National Institute of Standards and Technology (NIST, 2009). The NIST has widely recognised expertise in the field and has provided a comprehensive and universally accepted definition.

The NIST describes cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

There are several specific characteristics that may be used to describe the definition of cloud computing. According to the NIST (2009) these are the following characteristics:

1. **On-demand self-service**
   First off, it provides the ease of on-demand self-service, enabling customers to provide and utilise computer resources as needed without involving a service provider.
2. **Broad network access**
   In addition, it offers customers the freedom of broad network access, allowing them to access their online resources from a variety of locations and devices.
3. **Resource pooling**
   Thirdly, resource pooling is a technique used in cloud computing. This allows several users to share the same physical resources, improving resource consumption and efficiency.
4. **Rapid elasticity**
   Rapid elasticity is another benefit of cloud computing, which enables users to swiftly scale resources up or down in response to shifting demands.
5. **Measured service**
   Users are charged based on their actual usage, enabling cost optimisation, and providing accountability.

Additionally, the NIST defines three service models and four deployment models within cloud computing. The main difference between these two is that service model defines the level and type of services provided by the cloud service provider, while the deployment model describes how the cloud infrastructure is implemented and made accessible to users.

The three service models are:

1. Infrastructure as a Service (IaaS),
2. Platform as a Service (PaaS), and
3. Software as a Service (SaaS)

IaaS

IaaS is a cloud computing model that uses the internet to deliver virtualized computer resources. It provides services such as virtual machines, storage, networks, and other infrastructure components. Users have control over the underlying infrastructure and may manage and customize virtualized resources to meet their needs using IaaS.

PaaS

PaaS is a cloud computing approach that offers developers with a platform and environment to design, launch, and manage applications. Infrastructure, development tools, and runtime environments required for application development and deployment are often included in PaaS packages (Mohammed & Zeebaree, 2021). It encapsulates the underlying infrastructure, allowing developers to concentrate on code and application logic rather than server or operating system management.

SaaS

SaaS is a cloud computing approach in which programs are offered as a service via the internet. Users can access and utilise software applications hosted by a provider via SaaS, eliminating the requirement for local installation or administration. The program is often accessed via a web browser or a thin client (Rani, 2020).

The four main deployment models (NIST, 2009) in cloud computing are:

1. Public Cloud: In the public cloud deployment model, cloud resources are owned and operated by a third-party provider. These resources are made available to the public or a large industry group over the internet.
2. Private Cloud: The private cloud deployment model provides dedicated cloud resources exclusively to a single organisation. These resources may be located on-premises or be hosted by a third-party provider. Private cloud offers more control, customization, and privacy compared to public cloud services.
3. Community Cloud: In the community cloud deployment model, cloud resources are shared and utilised by multiple organisations that have common concerns, such as industry-specific regulations, security requirements, or compliance needs.
4. Hybrid Cloud: The hybrid cloud deployment model combines two or more distinct cloud deployment models (public, private, or community) that remain separate entities but are interconnected and can share data and applications. It allows organisations to leverage the benefits of different deployment models and optimise their resource allocation.

4.2 IaaS and PaaS

The primary focus of this research will be on the service models of IaaS and PaaS in the context of cloud computing.

The difference between IaaS and PaaS lies in the level of abstraction, specifically the runtime, middleware, operating systems, and virtualization as illustrated in Figure 1, created by the researcher. The terms are explained below with an example of each abstraction tier.

IaaS	PaaS	SaaS
Applications	Applications	Applications
Data	Data	Data
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
Operating system	Operating system	Operating system
Virtualization	Virtualisation	Virtualisation
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

= Managed by the vendor   = Managed by the customer

Figure 1 Cloud Delivery Models

- **Networking:** Networking components enable communication between cloud resources and ensure connectivity and data exchange. Amazon VPC (Virtual Private Cloud) allows users to create isolated network environments within the cloud, controlling network traffic and connectivity.

- **Storage:** Storage services offer scalable and secure solutions for customer data, files, and resources. Amazon S3 is an object storage service that provides scalable and durable storage for various types of data, such as documents, images, videos, and backups (Palankar et al., 2008).

- **Servers:** The physical hardware layer consists of servers that provide computing power and resources for hosting virtual machines and applications. Dell PowerEdge servers offer the physical hardware foundation for hosting virtual machines and running cloud services (Pogarcic et al., 2012).

- **Virtualisation:** Virtualisation involves creating virtual instances of computing resources, which enables resource sharing and effective usage by allowing numerous instances or workloads to run on a single physical server. In an IaaS environment users can deploy and manage virtual machines (VMs) using platforms like AWS EC2 or Google Compute Engine

(Vaquero, 2011). PaaS systems, on the other hand, provide higher-level abstractions like containers such as AWS Elastic Beanstalk or Google Kubernetes Engine and serverless functions such as AWS Lambda or Azure Functions (Dalbhanjan, 2015). Within these containers developers can specify their configuration options such as programming language, packaging type and even scaling policies. This simplifies the deployment and management of applications by abstracting away the underlying infrastructure.

- **Operating system:** Operating systems on an IaaS platform are installed and managed by the users (Mohammed & Zeebaree, 2021). Within PaaS users do not have direct access to and control over the operating system. The OS is controlled and updated by the provider.

- **Middleware:** Middleware indicates the layer of software that exists in between an operating system and an application and offers several services and features to help with communication and integration between various software parts (Lee et al., 2012). Middleware is imperative in the context of cloud computing for abstracting underlying infrastructure difficulties and enabling smooth interactions between applications and resources. pre-configured middleware services such as message queues and authentication services.

- **Runtime:** The execution environment in which programs or code can be executed is referred to as the runtime. It contains all the parts and libraries needed to run the program. Users may install and customize the required runtime components on virtual machines under the IaaS paradigm, giving them control over the runtime environment. PaaS provides controlled runtime environments with all the components and libraries required to run programs. For example, AWS Elastic Beanstalk offers pre-configured runtime environments for Java, .NET, PHP, Python, and other languages, allowing developers to deploy their applications without worrying about installing and administering the runtime themselves (AWS, 2011).

- **Data**: This tier involves services dedicated to storing, managing, and safeguarding customer-specific data, including files, databases, and other information.

- **Applications**: The top layer represents the end-user applications that are accessible over the internet (Freet et al., 2015). These are fully functional software solutions that users can use without needing to install or manage the software locally (e.g., Google mail, Microsoft 365, etc.).

4.3 The TOE framework

The research question of this study aims to identify key considerations for organisations undergoing such a transition. Therefore, it is just as important to be aware of which areas to focus on to identify the aspects which determine the adoption of cloud computing.

Furthermore, understanding the various technical offerings can pose a significant challenge when determining whether they will be beneficial for organisations. However, numerous frameworks exist for evaluating and analysing the potential advantages of adopting a new technology within an organisation. These notable frameworks include diffusion of innovations (DOI) by Rogers (2002), Unified Theory of Acceptance and Use of Technology (UTAUT) by Ahmad (2015), Technology Acceptance Model (TAM) by Davis (1989) and even combinations of these frameworks exemplified by the work of Ogunlolu and Rajanen (2019).

In the context of cloud adoption, the Technology-Organization-Environment (TOE) Framework emerges as exceptionally well-suited and adaptable (Borgman et al., 2013). This model is used in the field of information systems research to analyse and understand the factors that influence technology adoption and implementation within organisations (Jesus et al, 2019).

Borgman (2013) underscores in his article that the technological component of the framework emphasizes aspects and attributes of the technology itself. This includes facets such as its functionality, compatibility with existing systems, complexity, reliability, security, and potential for customisation.

The organisational factors encompass the internal organisational context and capabilities that influence technology adoption. This includes factors such as the organisation's structure, culture, resources, skills, knowledge, decision-making processes, and management support.

The final part of the framework is the environmental part, which includes external factors that impact technology adoption. This refers to aspects such as market conditions, industry norms, regulatory requirements, competitive pressures, and technological trends in the broader environment.

Application of the TOE framework can provide a structured approach to comprehensively evaluating the transition from IaaS to PaaS, considering technological, organisational, and environmental perspectives, and supporting informed decision-making and successful implementation. However, scholars have had several concerns regarding the framework, namely:

Dedrick and West (2003) assert that the TOE Framework serves more as a taxonomy for organising variables rather than a fully integrated conceptual framework or a well-established theory. This suggests that the TOE Framework might lack the depth and sophistication required to comprehensively study organisational adoption.

Musawa and Wahab (2012) point out that the TOE Framework has limitations in explaining the variance in technology adoption, citing the example of Electronic Data Interchange (EDI) adoption. A significant portion of the adoption variance remains unexplained, questioning the framework's effectiveness in providing comprehensive insights.

4.4 Service oriented Architecture (SOA) and Event Driven Architecture (EDA)

There have been many definitions of Service oriented Architecture (SOA) differing based on the perspective including technology, business, and architecture (Niknejad et al., 2020). These diverse perspectives highlight the multidimensional nature of SOA. At its core, SOA is an architectural approach that structures software systems as a collection of loosely coupled, interoperable and reusable services (NIST, 2009).

CC is a paradigm for large-scale distributed computing that makes use of existing technologies such as virtualisation and service-orientation. According to Zhelev and Rozeva (2019), choosing the right architecture is imperative for the scalability and extensibility needs of the application.

SOA and CC are two closely related concepts that have emerged as important paradigms for building and deploying modern distributed systems. Raines (2009) describes below the most common aspects between CC and SOA.

1. **Application Layer Components/Services**: Both CC and SOA emphasize the decomposition of applications into modular components or services at the application layer. This enables the development, deployment, and management of discrete units of functionality that can be independently operated and accessed.

2. **Network Dependence:** Both cloud computing and SOA rely on network connectivity for service interactions. Services in a cloud or SOA environment communicate over networks, either within the organisation's infrastructure or across the internet. This network dependence enables distributed communication and collaboration between services.

3. **Cloud/IP WAN-Supported Invocations:** CC and SOA leverage IP-based wide area networks (WANs) for invocations between services. This means that services can be accessed and invoked remotely over the internet or private networks, leveraging the scalability and global reach of cloud infrastructures or WAN technologies.

4. **Leveraging Distributed Software Assets**: Both cloud computing and SOA enable the leveraging of distributed software assets. Services in a cloud or SOA environment can be deployed across multiple locations, data centers, or cloud regions. This distribution of software assets allows for improved fault tolerance, scalability, and availability.

5. **Producer/Consumer Model:** The producer/consumer model is a core concept in both CC and SOA. Services act as either producers or consumers of functionality. Producers provide specific services or data, while consumers utilise these services or data to fulfil their own requirements. This model facilitates loose coupling and allows for flexibility in service consumption and composition.

Other reasons why scholars may find SOA to be important when discussing CC is listed below.

Garrison and Nova (2017)

SOA's are important when discussing CC is because cloud-native applications are designed to run on modern cloud infrastructure and require certain infrastructure features to function effectively. Failure to consider elements such as container orchestration, microservices architecture, and elastic scaling infrastructure requirements can lead to poor application performance, scalability issues, and increased operational costs.

NIST (2009)
SOA, as an architectural approach, emphasizes the modularization and integration of software components into services. Cloud computing, with its ability to provide on-demand services and resources, aligns well with the principles of SOA because of its natural characteristics and capabilities.

Wang et al. (2015)
The idea of services is the main emphasis of service-oriented architecture. Services in SOA are self-contained, modular functional units intended to be interchangeable and reusable. Specific business functions are encapsulated in services, which are exposed through well specified interfaces. Synchronous request-response interactions between services, which may entail transferring data or calling processes, are the norm for communication in SOA.

PaaS and SOA emphasize agility and flexibility in application development and deployment. PaaS platforms provide tools and services that enable rapid development, deployment, and scaling of applications. Similarly, SOA supports the creation of agile and adaptable systems by decoupling services, enabling independent updates, and allowing for dynamic composition of services.

Martins et al. (2013)
Cloud solutions are tightly coupled to the proprietary technology they were created for making them strongly dependent on a single infrastructure. If software vendors deliver cloud-native applications that are tightly coupled with the cloud provider's platform, the impact of a migration can be significant on the application stack. Therefore, having an agile and scalable infrastructure, such as SOA, allows enterprises to quickly adapt to changing demands and scale resources up and down based on the workload requirements.

EDA

Another commonly used architectural method is called event driven architecture (EDA). The main emphasis in event-driven architecture is on events and how they spread across the system (Bukhsh et al, 2015). It is a loosely linked method in which services or components exchange events to interact. Events are important occurrences or changes in the system that cause other components to act or respond. Scalability, real-time responsiveness, and asynchronous connectivity are all made possible through EDA.

While loose coupling, flexibility, and modularity are goals shared by EDA and SOA, they differ in their emphasis on events and services as the main building elements. EDA is especially well suited for event-driven and real-time systems, where the ability to react quickly to occurrences is essential. Creating reusable and interoperable services that may be combined to satisfy different business objectives is the main goal of SOA, on the other hand. In the article of Bukhsh (2015) are described the similarities, differences, and other recommendations.

Mainly, EDA does not inherently emphasize reusability in the same way as SOA. It focuses more on the real-time processing of events and event-driven systems, where events are consumed and processed by specific components (Wang et al., 2015). Furthermore, EDA is based on an event-driven communication style. Components generate and publish events, and other components subscribe to these events and react accordingly. The communication is asynchronous and event-triggered, whereas in SOA communication follows a synchronous procedure.

Hence, SOA is more aligned with business domains and emphasizes reusability, while EDA focuses on leveraging real-time event processing and responsiveness. The responsive nature of EDA aligns well with the scalability and agility benefits offered by PaaS (Martins et al., 2015).

4.5 Migration strategies

Organisations migrate to the cloud for various reasons, driven by the potential benefits and opportunities it offers. For example, one of the primary motivations for companies to migrate to the cloud is the ability to scale their infrastructure and resources dynamically.

In addition to the reasons why companies migrate to the cloud, it is equally important for organisations to carefully consider and select the appropriate migration strategy that aligns with their specific needs and goals.

Ahmad et al (2018) explains the following 5 migration strategies regarding transitions to the cloud:

1. **Rehosting:** Rehosting, also known as lift-and-shift, involves migrating an application or system from an on-premises environment to the cloud without making significant changes to the underlying architecture or code. In this strategy, the application is typically moved to an IaaS platform. The main advantage of rehosting is that it allows for a quick migration with minimal modifications, but it may not take full advantage of cloud-native features and benefits.

2. **Replatform:** Replatforming involves migrating an application to a cloud platform, such as a PaaS offering, while making some modifications to optimise its performance and leverage cloud-specific capabilities. It involves taking advantage of certain cloud services, such as managed databases or serverless computing, to enhance scalability, availability, and manageability. This strategy offers a balance between speed and optimisation compared to rehosting.

3. **Repurchase:** The repurchase strategy involves replacing an existing application with a software-as-a-service (SaaS) solution. Instead of migrating the application to the cloud, the organisation opts for a cloud-based software that meets their requirements. This approach eliminates the need to manage the infrastructure and software stack, as the SaaS provider takes care of the hosting and maintenance. It offers simplicity and reduced overhead but may require adjustments in workflows and integrations.

4. **Refactor**: Refactoring, also referred to as re-architecting or cloudifying, involves making significant changes to an application's architecture and code to optimise it for cloud-native environments. It aims to leverage cloud services, scalability, and elasticity by decomposing monolithic applications into microservices or using modern architectural patterns like serverless computing. This strategy requires substantial development effort but offers greater scalability, flexibility, and cost optimisation.

5. **Rebuild:** Rebuilding entails completely redeveloping an application from scratch, often using cloud-native technologies and best practices. This approach involves taking full advantage of cloud services, microservices, containers, and serverless computing. By designing and building the application specifically for the cloud, organisations can achieve optimal scalability, resiliency, and cost efficiency. Rebuilding provides the highest level of cloud-native benefits but requires a significant investment in development resources and time.

Each migration strategy offers a different level of effort, complexity, and benefits, depending on the organisation's goals, existing application architecture, and cloud adoption strategy.

Pahl (2015) developed a migration process framework to explain the importance of architecture when moving to a PaaS platform. The most common concerns that emerged were architecture re-engineering and architecture ageing. Architecture re-engineering is the process of transforming an existing software system's architecture to address performance issues, maintainability, scalability, and other quality attributes (Khom, 2007). Architectural ageing is the steady deterioration of a software system's design over time as a result of many causes such shifting business requirements, emerging technology, and system evolution (De Lemos, 2013). Other scholars such as Fowley et al. (2018) supplement this by explaining that while some applications can be directly migrated to the cloud with minimal modifications, others may require significant changes to fully utilise the cloud provider's offerings.

Rai et al. (2015) also describe 5 phases each organisation should go through when considering a cloud migration. These phases are described as below.

Phase 1: Feasibility Study
During this phase, organisations assess the benefits and risks associated with cloud migration, considering factors like cost, scalability, security, and compliance. This helps in setting realistic expectations and determining whether cloud migration aligns with the organisation's strategic goals.

Phase 2: Requirement Analysis & Migration Planning
In this phase, organisations conduct a detailed analysis of their existing IT systems and applications to identify migration requirements and dependencies.

Phase 3: Migration Execution
The migration execution phase involves the actual movement of applications and data to the cloud environment as per the migration plan.

Phase 4: Testing & Migration Validation
Once the migration is complete, the testing and migration validation phase is conducted to ensure the successful migration of applications and services.

Phase 5: Monitoring & Maintenance
The final phase of the cloud migration process involves continuous monitoring and maintenance of the cloud environment. Organisations implement monitoring and management tools to track the performance, availability, and cost of cloud resources.

Figure 2 5-phase cloud migration model

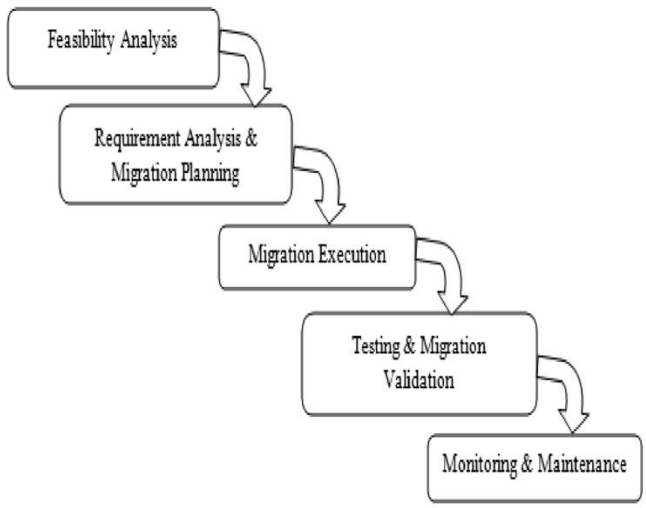


Figure 3 5-Phase Cloud Migration Model.

Note. By Rai et al.,(2015), 5-Phase Cloud Migration Model

5 Methods and Design

This overall objective of this research is to add value to the academic field of cloud computing. Next to that, the delivered theory of this research aspires to support Dyflexis B.V. and other independent software vendors (ISVs) who are seeking to transition from IaaS to PaaS.

The upcoming paragraphs of this chapter will describe the methods used to answer the main research question. Additionally, the practical techniques employed to address the research question and the rationale behind the chosen design decisions are described here. Initially, the research approach is examined, followed by a discussion of the specific research method selected for the study. Subsequently, the chapter elaborates on the procedures for sampling, data collection, and data analysis.

As a result of this project, a theory will be presented. Concerning the resilience and reliability of the data and ensuring that multiple views and sources of information are considered, the triangulation technique will be applied. In qualitative research, triangulation refers to the use of various methodologies or data sources to build a thorough comprehension of the topic (Carter et al., 2014). The proposed theory in this research is developed based on multiple sources of information along with the AWS (Amazon Web Services) framework, interviews with experts and a thorough case study of the planning software company Dyflexis. By uitilising theses multiple sources, a comprehensive theory is formed surrounding diverse aspect of the migration from IaaS to PaaS.

5.1 Research approach

To answer the research question as well as the sub-questions, a qualitative approach will be taken. Qualitative research is commonly used to investigate and comprehend a phenomena in depth (Barnham, 2015). In situations where quantitative approaches may fall short, qualitative research can give a more complete knowledge of a participant’s experiences and relationship within an organisation.

Additionally, a qualitative approach can be beneficial for exploring new areas of research where little is known about the subject (Strauss & Corbin, 1998). This covers a specific sector like moving from IaaS to PaaS. Qualitative approaches enable an extensive examination of the experiences and viewpoints of those involved in the transition.

Therefore, the methods used for this study are case studies and interviews. By conducting interviews and using case studies one can explore the risks, challenges and motivations involved in migrating to PaaS.

In addition to individually comparing the results of the interviews, this study will also analyse the results from the interviews with the outcomes of the case. This comparison is done to find commonalities and differences in the domains of risk and benefits, technical requirements, and architectural differences.

This varied comparison between the interviews and case study will provide a thorough understanding of the transition process.

The comparison identifies areas of agreement where the interview results and the Case Study are comparable. On the other hand, the comparison could also highlight areas where the interview findings and the Case Study disagree. These differences bring focus on certain elements, conditions, or challenges that exist in the specific context of the case.

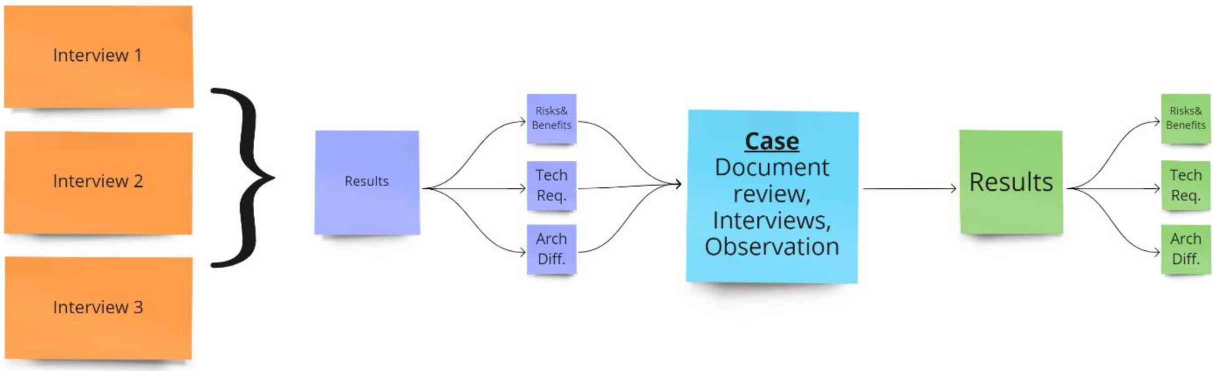


Figure 3 Approach Interviews and Case study

The decision is made to conduct interviews prior to the analysis of the case to set a strong foundation for the next stages. Accordingly, the insights gathered during the interviews help narrow down and delve into the essential factors to investigate further during the case. Ultimately, the case study will deliver findings across the three domains.

The results of the interviews will be carefully compared to the conclusions drawn from the Case Study. The three domains of risk and benefits, technical requirements, and architectural changes will be the focus of this comparison analysis. This analysis is further explained in the Discussion section.

5.2 Interviews

Interviews will be used to collect in-depth data about people's experiences, opinions, and perspectives on the architecture of Cloud Computing. Interviews will be conducted with IT (Cloud)experts and developers responsible for managing and implementing the migration, as well as stakeholders such as managers and business executives who may be impacted by the shift.

The objective of the interviews is to gather a focused and detailed understanding of cloud transitioning on an individual level. These interviews will help in comprehending real-world threats and benefits that companies may experience throughout the transition period.

To establish reliable results grounded theory and qualitative analysis will be used. According to Charmaz and Belgrave (2012) grounded theory and qualitative interviewing are ideal methods for collecting qualitative data. Qualitative interviewing is a method for gathering rich and thorough data from participants through in-depth interviews. The grounded theory method is an iterative and inductive technique to qualitative data analysis that allows for the construction of ideas based on the data itself.

In-depth interviews with relevant stakeholders, such as IT professionals, cloud specialists, or representatives from organisations that have transitioned to PaaS, are conducted during qualitative interviewing. Furthermore, qualitative interviews will help obtain extensive and nuanced data about the possible risks and benefits of switching to a PaaS platform, technical requirements for operating on PaaS, and architectural distinctions between IaaS and PaaS platforms in the context of the current transition. The interviews are semi-structured in order to allow exploration of emergent topics and the gathering of detailed insights. Open-ended questions are created to prevent any form of bias and enable participants to express their transition-related experiences, problems, and viewpoints.

The data collected through qualitative interviews can be analysed using the principles of grounded theory. According to Strauss and Corbin’s (1990) the first step is to apply open coding after the interviews have been transcribed. This involves dividing the data into smaller relevant components and providing descriptive codes to each of them.

Subsequently links and interactions will be created between the codes and categories discovered during open coding. Axial coding helps examine the interrelationships and connections between the codes and categories, resulting in a more complete theory for understanding the migration from IaaS to PaaS.

5.3 Study Sample

As mentioned before, the objective of the interview is to collect enough evidence to develop a solid theory that accurately represents the phenomenon under the research. To reach the objective it is of eminent importance that the interviewees are expected to have an expert’s understanding of cloud computing.

If the participants lack an understanding of the topic, the results could be stated as deceitful or inaccurate. Fortunately, many of the developers within Dyflexis have experienced similar migrations and all the external cloud experts understand the phenomena quite well. Through connections of Dyflexis external interviews have been scheduled with senior infrastructure project managers, DevOps Cloud Engineers, and AWS Heroes (leaders who are active and outspoken in AWS communities). All originating from or currently occupying a commanding position within a multinational company. Specific characteristics of the participants are mentioned in Appendix 1 : Background interviewees.

Interviews shall be held individually and will be carried out in-person to gain a better understanding of the participants' social and physical contexts (Strauss and Corbin, 1990). The presence of an interviewer helps participants to deliver thorough and context-rich replies and fosters a personal connection. Furthermore, visual signals, body language, and facial expressions can also convey extra meaning and depth that other interview styles may overlook.

The number of interviews for this complex situation is stated between eight and twelve interviews. However, this may vary depending on the gathered information. Throughout the interviews continuous comparison is applied, involving the comparison of new data to the previous collected data, defining similarities and differences. This technique is used to determine theoretical saturation, which is the point at which data gathering is complete and more data collection is unlikely to offer new insights or classifications (Rowlands et al., 2016).

5.4 Case study

Additionally, a comprehensive case study of Dyflexis will be done. A case study is a research approach and factual method for investigating real-world events (Alshdadi et al., 2020). The purpose of this case study will be to examine the specific path the company will follow in transitioning from IaaS to PaaS. By examining their transition experience and (expected) outcomes, significant insights and practical concerns will surface to inform the theory. Additionally, case studies are a recommended technique when there is minimal control over occurrences and the focus is on contemporary phenomena in some real-life setting (Yin, 1994).

A case study method should have congruence with philosophical considerations, inquiry approaches, and a research logic consideration (Rashid et al., 2019). According to Rashid (2019), these considerations set the fundament for conducting a case study. Here, a reason for each phase is provided on why a case study method is suitable for this research project that focuses on the transition from IaaS to PaaS.

Philosophical consideration

Another word for philosophical consideration is “theoretical thinking”, which refers to the act of deeply thinking about the underlying principles, beliefs, or theories that guide one's approach to a subject or problem (Popil, 2011).

Before Dyflexis makes the decision to transition from IaaS to PaaS, they likely will engage in theoretical thinking to guide their decisions:

- Regarding the underlying principles of the organisation, Dyflexis might value flexibility and scalability as much as the efficiency of the development teams. This results in considering an alignment of these values and a PaaS model.
- Next to that, Dyflexis may also have certain beliefs about the PaaS technology. For example, their believe that moving to PaaS will lead to long-term cost savings.

Inquiry approach
Inquiry techniques involve data collection methods including observations of the implementation and usage of the PaaS solution in the organisation, analyses of relevant documents such as contracts, agreements, and documentation related to the transition.

Data gathering methods utilised in the case study are included under inquiry approaches. This will include interviews with IT experts, managers, and decision-makers engaged in the shift. They may witness the deployment and usefulness of the PaaS solution throughout the organisation. Furthermore, the approach in this case includes the study of important papers such as contracts, agreements, and transition paperwork (blueprints).

Research logic
The general strategy and structure of the case study are referred to as research logic (Rashid et al., 2019). For example, the collection of data at various phases of the transition process to discover the elements that contribute to or interfere with an effective adoption.

The case study concerning Dyflexis will be used in combination with the AWS Cloud Adoption Framework (CAF) and the research questions. The CAF offers a systematic method to AWS cloud transition. This is applied to supplement the research and give useful insights.

The AWS CAF will serve as a guide to understanding the general principles and best practices of cloud computing, including the specific concepts and considerations related to PaaS. It describes the major stages, actions, and concerns required in a successful cloud transformation. Therefore, making it a reliable framework to use for a case study about an organisation planning to transition from IaaS to PaaS.

As mentioned before, the framework addresses a variety of elements of cloud transformation, such as business, people, governance, platform, security, and operations. It offers a comprehensive view of the migration process and guarantees that all essential factors are considered (AWS, 2016). This extensive coverage enables the case study to address the organisation's needs from numerous perspectives and assess its preparedness and success in each area during the transition.

The findings, analysis and recommendations of the case are presented in a separate case study report (Appendix 4: Case study report). The report will provide background information about the organisation, its current IaaS infrastructure, and the reasons behind the decision to move to PaaS.

By integrating the AWS framework within the case study, the results can be placed within industry standards, ensuring that the findings and insights remain relevant and applicable to a wider audience. In addition to the framework the company will participate in an AWS cloud assessment, which examines an organisation's present IT environment and identifies cloud adoption and optimisation opportunities.

Furthermore, the assessment analyses and assesses potential migration risks such as data loss, service availability, and vendor lock-in. The results of the assessment will be analysed and compared to the answers of the interviews with cloud experts. The constant comparison is done to minimize bias and increase reliability of the results (Abdelmaboud et al., 2014).

6 Results

This research aimed to explore the issues involved in the adoption of PaaS while explaining the practical impact of a migration from IaaS to PaaS. As mentioned in chapter 5 the results were obtained using the research methods case studies and interviews. Interviews were conducted with cloud experts and IT professionals, to gather their experiences, perspectives and motivation related to a PaaS adoption. The case study conducted here involved analysing a real-life scenario where an Independent Software Vendor (ISV), Dyflexis, is planning on migrating from IaaS to PaaS.

Both methods of data collection delivered several insightful results. Overlapping themes and ideas linked to the transition came forth in both the interviews and the case study. However, because they emerged more strongly using that specific approach, several themes and concepts were more thoroughly explored and described in either the findings of the interviews or the results of the case study.

6.1 Results interviews

A set of nine interviews was conducted to offer a well-rounded view on the topic. As seen in the table the interviews covered a wide range of roles, companies, industries, and experiences. For a deeper understanding of the interviewees' profiles, please refer to the detailed information in Appendix 1.

The choice to conduct this number of interviews was supported by evidence of the methodological concept theoretical saturation, as discussed in chapter 5.

Interviewee	Role	Organisation	Industry	Years experience CC
1	Lead engineer	PostNL	Logistics	15
2	IT Analyst	Dyflexis	Software Development	8
3	Devops Engineer	Chantelle Digital Lab	Apparel	12
4	Engineering Manager	Dyflexis	Software Development	12
5	Project Manager	Shell	Energy	25
6	Cloud operations engineer	EVision Industry Software	Enterprise Software	15
7	Software Engineer	Eventstore	Technology	9
8	Cloud engineer	Nike	Retail	10
9	Developer	Dyflexis	Software Development	7

Table 1 Background information interviewees

An overview is provided of the labels used for the interviews conducted (Table 2 Initial set of Labels). To prevent duplicate codes a constant comparison was made of the codes assigned to similar themes. When duplicate codes were found, these were refined by clarifying their definitions which in turn emphasized the data more effectively.

Utilising the qualitative data analysis method of axial coding, the interview outcomes were examined. The information was methodically divided into categories and subcategories, enabling a thorough comprehension of the viewpoints of the respondents. During the coding process, common themes and patterns were found throughout the interviews. Important insights were also extracted, and links between various concepts were established.

By comparing the data from the interviews, recurring themes and variations in the interviewees' experiences and perspectives were stated, regarding the switch from IaaS to PaaS. This investigation focusses on the difficulties, advantages, and factors involved in such a move.

Trade off principle	Serverless	IaaS as a base for cloud computing	Load spreading/scaling	docker	Cost reduction
Wide spread database structure	function as code	Regulatory and compliance governance	Cloud native	Containerisation	Improved productivity
Core competence driven	Compute, storage and network	Orchestration	Multi cloud architecture	risks associated	scaling operations
Scalable cloud solutions	tooling solutions	Lifecycle management	Technology solution	Vendor-lock-in	Data at rest
Flexible capacity expansion at hand	Committing to the cloud	Long term cloud decision	legacy systems	Benefits	Caching
Competitor driven decision	Long term cloud decisions	Deployment cycle diversity	Network traffic	Time to market	Messaging services
Cloud gives a competitive advantage	identifiable workloads	latency levels	rearchitecting	Control over infrastructure	API
Trade off principle	Context dependable solutions	service levels	Budget approval	Maintenance	software stack
Scalability opportunities	Commercial software	Reliability	Kubernetes	Total cost of ownership	solution
Data structure	not cloud native software.	Cost effective	in-house knowledge	Compatibility	Case based
Authorisation	Shared cloud responsibility model	Sla	Business value delivery	Middleware	Monolith application
Cached data service	PaaS as a fundament	SaaS	scalability	Deployment	Business view
Data driven architecture	Cost saving solution	Business need	Ci/cd pipeline	integration	Multi cloud
Unforeseen issues	function as a service	Risk averse	cost effective	IaaS	CFO
Monolith migrations	Variable expense	Business value	Authentication	PaaS	finance
Service driven	Auto scaling	overhead costs	Compliance	Level of abstraction	CTO
Partial migration	Featureful computing	On premise	Monolithic	Pre-configured	PaaS limitations
Cloud-native vs cloud-based	Capacity scaling	Licensing costs	Lifetime expectancy	Compliance adherence	Databrick
Breaking down the monolith	Agile	Cost benefit analysis	Roadmap	Agility	Kubernetes
Failover	Enterprise architecture	Core business focus	Aws	Flexibility	Loosely coupled components
Redundancy			data center	overhead complexity	Business reason

Table 2 Initial set of Labels

In the analysis of the interviews, a total of 121 labels were used to categorize and code the data from each separate interview. However, it became apparent that working with such a large number of labels posed challenges in terms of manageability and efficiency in establishing linkages and themes within the data.

The initial set of 121 labels was reduced to 28 labels to improve manageability and logical relationship identification. This was done by merging labels that represented similar concepts and ideas and discarding labels that did not contribute to the analysis. This led to themes and categories that included several linked labels rather than working with a huge number of individual labels.

The process of transitioning from an initial set of 121 labels to a more manageable and focused collection of 28 labels involved the implementation of the following measures:

1. Grouping of the codes based on their similarities and connection. This initial step involves axial coding, where clusters of related concepts are made.
2. The iterative process of constant comparison where new labels were created and compared to existing labels led to enhancement of the categories.
3. Examine each code's applicability and analytical significance, particularly in context of the developed category.
4. Combine codes that indicate closely related topics. For example, "Platform services availability" refer to the reliability of certain services running on the platform. This covers initial labels such as "redundancy" and "failover services".
5. Continue categorising and removing labels iteratively, until theoretical saturation is reached where more data gathering no longer alters the results.

These steps led to the following categorisation and position of the 28 labels:

Migration and Deployment

1. Data Migration
2. Simplified Deployment
3. Application Portability
4. Platform Updates and Maintenance
5. Continuous Integration and Continuous Deployment (CI/CD)

Strategic considerations

6. Risk Assessment
7. Benefits Analysis
8. Cost optimisation
9. Vendor Lock-in
10. Flexibility and Agility
11. Infrastructure abstraction

Scalability and Performance:

12. Scalability
13. Performance Optimisation
14. High Availability & Fault Tolerance
15. Load Balancing and Auto-Scaling

Resource Management and Efficiency:

16. Resource Management
17. Operational Efficiency
18. Developer Productivity
19. Automation Capabilities

Security and Compliance:

20. Security Enhancements
21. Compliance and Governance

Service and Support:

22. Service Level Agreements (SLAs)
23. Platform Services Availability
24. API and Integration Support
25. Network Configuration and Connectivity

Data and Database services:

26. Database Services
27. Monitoring and Analysis:
28. Application Performance Monitoring (APM)

To provide a more structured overview of the agreements and disagreements among the experts, the labels have been separated into three categories: agreed on, disagreed on, and neutral. The labels in the "agreed on" category have been agreed upon by the respondents as to their importance and effects.

On the other hand, labels in the "disagreed" category are those where different viewpoints or ideas were noticed among the interviewees.

Finally, the "neutral" category includes labels where there was no apparent consensus or unclear results.

The agreed topics are mostly the benefits of PaaS described in 6.1.1, which include labels such as Developer productivity, Scalability and Elasticity and Automation capabilities. Further discussed by the cloud experts was the level of infrastructure abstraction, which is described in 6.1.2.

The concepts where the experts disagreed on reside in the risks and challenges of such a transition, described in paragraph 6.1.3. The paragraph 6.1.4 is neutral but nonetheless contains important concepts mentioned by the experts, which include labels such as Benefits Analysis, risk assessment and Compliance and governance.

Out of the 28 labels mentioned, it is important to note that not all of them are covered in the discussion. The focus has been on addressing the key and significant labels where clear linkages and connections are established. These include topics such as infrastructure abstraction, high availability, and multi-cloud architecture. These labels have been prioritized due to their relevance and the meaningful insights they provide in the context of the discussion. It is important to consider that other labels may still hold significance, but the emphasis has been placed on addressing the ones with clear connections and established relevance.

Agreed on	**Disagreed on**	**Neutral opinions**
Scalability Performance Optimisation Security Enhancements Flexibility and Agility Operational Efficiency Developer Productivity Simplified Deployment Infrastructure Abstraction Load Balancing and Auto-Scaling Cost Optimisation Compliance and Governance	Vendor Lock-in Data Migration High Availability and Fault Tolerance Data Migration Multi-cloud Architecture	Risk Assessment Application Portability Benefits Analysis Resource Management Service Level Agreements Automation Capabilities Application Monitoring and Logging Platform Updates and Maintenance API and Integration Support CI/CD

Table 3 Distinguished labels

To enable a thorough analysis of the data, each label indicated a unique characteristic or thought mentioned in the interviews. The analysis identified the linkages and interconnections between the codes and categories by using these labels. This made it possible to recognise recurrent themes, common patterns, and broad ideas that appeared across the interviews. This resulted in the formation of the following concepts:

- The benefits and risks of transitioning from an IaaS model to a PaaS model
- What distinguishes the cloud delivery models from each other
- What platform-specific requirements there are when running an application on a PaaS platform

6.1.1 Benefits

The benefits and possible downsides of switching from IaaS to PaaS were discussed with a group of cloud specialists throughout the interview process. Most of the experts had similar points of views, even though their responses varied in wording. Only the most important and well-discussed benefits of transitioning are highlighted in this study. These include **Cost optimisation, Enhanced developer productivity, Scalability and Elasticity and Enhanced security capabilities**.

<u>Cost optimisation</u>

Many experts agreed on that organisations using IaaS oversee administering and maintaining the infrastructure's foundation, which includes servers, storage, and networking. PaaS platforms often offer flexible pricing models, allowing businesses to pay for resources and services based on usage, resulting in better cost optimisation. Whereas infrastructure components for IaaS sometimes have set monthly or hourly charges, regardless of how many resources are actually used.

"PaaS platforms typically follow a pay-as-you-go pricing model. This lets companies to only pay for the resources and services they use" - interviewee 8

<u>Enhanced Developer Productivity</u>

Another agreeable aspect among the experts was that the shift to PaaS enables developers to focus more on application development and less on infrastructure management. PaaS platforms provide built-in tools and services that streamline development processes, allowing developers to increase productivity and deliver applications more efficiently. All nine experts agreed on that PaaS offered built-in tools making management of applications easier and give developers insightful data on how well their work is performing, enabling them to see problems early and fix them.

"PaaS provides extensive tools and services, including rollback, deployment, configuration management, and version control, for managing the whole application lifecycle." - interviewee 2

<u>Scalability and Elasticity</u>

All experts (9) acknowledged the fact that PaaS includes built-in elasticity and scalability characteristics that enable applications to grow dynamically based on demand. By allowing companies to efficiently handle workload variations without manual intervention, performance and customer experience are enhanced.

"While some flexibility may be sacrificed, the potential for leveraging advanced platform capabilities and focusing on value-added activities can lead to increased agility and the exploration of new business opportunities." - interviewee 8

<u>Enhanced security capabilities</u>

Another fully agreed section experts discussed was that PaaS providers offer robust security measures and compliance frameworks to address security concerns and ensure regulatory compliance. They highlighted that In PaaS, the cloud provider has become more in charge of security, including that of the platform services and underlying infrastructure. To protect the platform and its individual components, the provider utilises security controls and mechanisms such authentication, access restrictions, data encryption, and vulnerability management.

"Moving to PaaS can provide advantages including improved platform security, compliance support, and simplified security administration, even if some control may be given up in terms of implementing infrastructure-level security controls. By providing tools, documentation, and built-in security measures that are in line with various criteria, PaaS providers frequently follow industry standards and laws, helping enterprises achieve compliance." - interviewee 6

6.1.2 Differences in Cloud Delivery Models

Out of the interviews it became apparent that the experts emphasized the distinction in the infrastructure and responsibility of the infrastructure between the two delivery models.
Table 4 serves as a representation of the key points discussed by the experts, allowing for a clear comparison between the two cloud service models.

"Make sure you have a clear separation of responsibilities. Storage vs Compute vs Network. This is what is delivered among all cloud providers and what differs the delivery models." interviewee 1

As seen in the header of the table all experts specifically highlight three key areas of responsibility: storage, compute, and network. Regardless of the specific cloud platform or provider chosen, these components form the foundation of the infrastructure.

These three key areas of responsibility—storage, compute, and network—reside within the broader category of "**infrastructure abstraction**, which refers to the process of abstracting (or hiding) the underlying infrastructure details and complexities from the users or developers.

With PaaS, the focus shifts from managing and configuring the infrastructure components individually (as in IaaS) to leveraging a higher-level platform that handles these responsibilities. PaaS platforms provide a managed environment where developers can focus more on application development and deployment, while the underlying storage, compute, and network infrastructure are abstracted and managed by the platform provider.

	Compute	**Storage**	**Network**
IaaS	Overprovisioning, controlling, and managing virtual servers. Self-managed containerisation and orchestration services	Control over the configuration of storage types: block, object, or file storage. Responsibility of the storage infrastructure lies with the customer	TCP/IP managed by the customer, who is responsible for the configuration of virtual networks, subnets, firewalls, and load balancers
PaaS	Automatically scaling EC2's based on demand. Messaging and queuing services are handled by the PaaS provider	The responsibility of backups, database operating systems lies with the provider	Pre-configured network settings and security features offered by the platform

Table 4 Abstraction between IaaS and PaaS

6.1.3 Risks

Opposed to the agreed benefit areas, there were some differing opinions among the experts concerning the points of Learning curve, Data migration and the absence of a Multi-cloud architecture.

Learning curve

The first disagreeing point among the experts was on how risky it was to switch from IaaS to PaaS because of the perceived knowledge gap. Half of the group of experts perceived this knowledge gap as a risk due to the participation of other parties, with specialized cloud expertise, that businesses may need to contract. They expressed concern that these third parties would not always operate in the organisation's best interests. They believed that depending on outside expertise would result in conflicts of interest and barriers in decision-making.

"There are quite a few times I've seen both AWS consultants and Azure consultants coming in and suggesting things that are not very good for the customer but very good for the bottom line of AWS or Azure". - interviewee 5

The second group of experts did not perceive the steep learning curve as a significant risk and highlighted the support mechanisms established by cloud providers, including educational materials, white papers and guidance. They believed that these resources effectively mitigate the risks associated with the knowledge gap.

"I do not see any major risks, especially in not having any experiences with PaaS. There are many reliable sources if you do not have a direct solution for the problem"- interviewee 6

Data migration

There was also a disagreement regarding the existence of risks in the process of data migration. The first group emphasized difficulties with data integrity, compatibility problems, and possible data loss during the change. These experts thought that switching platforms may put the organisation's data assets and overall operational continuity at risk.

The other group of experts did not think the data migration procedure posed any substantial threats. They emphasized the durability and dependability of the cloud providers' services and tools for data transfer. These experts stated that the risks of data transfer could be successfully reduced with careful planning, intensive testing, and adherence to best practices. They had the opinion that cloud providers give assistance and tools to guarantee a secure and easy data migration process.

Multi-cloud architecture

The final topic of disagreement among the experts was the choice of whether to use a multi-cloud strategy or host data at a single cloud provider. The first group of experts recommended hosting data at a single cloud provider due to its simplicity, streamlined management, and significant cost savings. They suggested that grouping data and services under one source facilitated faster integration, centralized management, and perhaps even better pricing via volume savings. They had the opinion that depending on a single provider would simplify management and lessen the complexity of multi-cloud settings.

Some experts believed that enterprises could achieve the freedom to run their applications on many clouds by developing cloud-native solutions, which are particularly created to utilise the benefits of cloud computing. Modern architecture patterns, such as microservices, containers, and serverless computing, are frequently used in the development of cloud-native applications. They believed that this kind of flexibility allows organisations to take advantage of the strengths and offerings of different cloud platforms, mitigating risks associated with vendor lock-in and providing greater resilience to potential service disruptions.

"I'm always a big fan of having at least two vendors playing, and that's good as well for failover, automatic redundancy, etc."- interviewee 2

6.1.4 Requirements for ISVs

Regarding the specific technical requirements for an independent software provider's application, the cloud specialists did not offer a conclusive solution. Instead, they emphasized that these requirements would vary depending on the business goals of the organisation. In other words, there is no one-size-fits-all answer, and the technical needs of the application should align with the objectives of the business. This agreement among the cloud experts relates to the area of "Long-Term-Cloud Decision".

Although the experts did not provide explicit answers, they did delve into the crucial aspects of PaaS. They highlighted significant features such as deployment and configuration, scalability and load balancing, and platform APIs and services. These elements play a vital role in PaaS, as they contribute to the effective management and optimisation of applications on cloud platforms.

As mentioned before ISVs need scalability and elasticity to manage changing workloads and guarantee outstanding performance of their product. The application should make use of the PaaS platform's scalability capabilities, such as auto-scaling and load balancing, throughout the move. This may involve defining resource thresholds, configuring scaling policies, and implementing dynamic scaling depending on measurements given by the platform. To do so, some ISV's need to make architectural changes to their current infrastructure.

"If you have this big monolith running somewhere and want to migrate it to the cloud. Well, that is going to cost a lot of money and cause you a lot of problems" interviewee 1

The majority of the experts agreed on that by breaking down a monolithic application into loosely coupled components, each component can be individually scaled based on demand, allowing for more efficient resource utilisation. Additionally, with loosely coupled components, it becomes easier to introduce new functionalities, make updates, or fix issues without impacting the entire application. This modular approach promotes flexibility, agility, and scalability, which aligns well with the capabilities provided by PaaS platforms. However, elaboration of breaking down the monolith was not further discussed during the interviews, since they all mentioned these that this, once again, depends on the business objectives and the way the organisation/application is structured.

6.1.5 Summary

The image illustrated here summarizes the considerations according to the cloud experts when transitioning from IaaS to PaaS.

Making the switch from IaaS to PaaS has benefits including cost optimisation, increased development productivity, and higher scalability and elasticity.

However, organisations must deal with the risks associated with data transfer, the learning curve, and the complexity of a multi-cloud architecture.

Technical decisions are made in accordance with the business' goals, and architectural changes are based on long-term cloud ambitions.

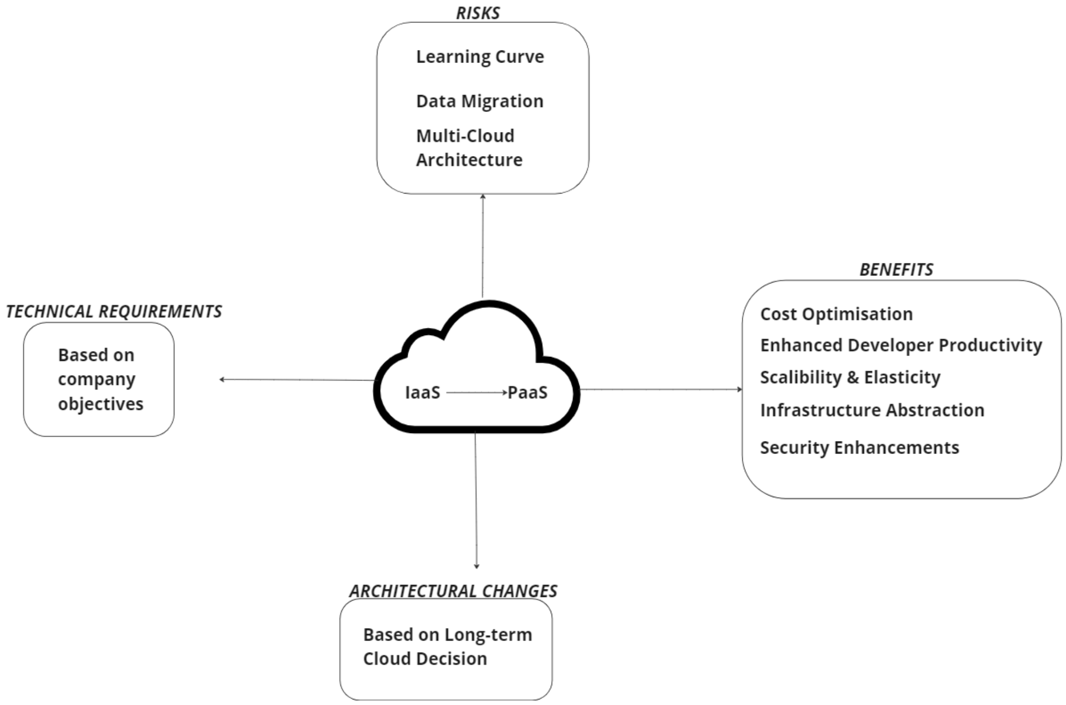


Figure 4 Visualisation of interview findings

6.2 Results Case Study

To learn more about a specific company's shift from IaaS to PaaS, a case study of Dyflexis was done in addition to the interviews. The goal of the case study was to look at the precise route that Dyflexis would take throughout the migration and to pinpoint the important factors that would influence their decision-making. Documentation review, interviews, and observations of the company's existing infrastructure and operations were mainly used to gather the findings.

The interviews involved engaging directly with stakeholders closely associated with the migration. Three interviews were done altogether since the answers of the interviews were similar with findings of the case study observations and document analysis. Throughout this chapter some statements and information may be referenced as "personal communication". This tag is applied because the documents and sources are not publicly available.

The interviews within the case study were conducted in a relatively unstructured manner. As a result, the points gathered from the interviews were combined and integrated with the findings from the document review. These points are described as below.

- The motivation to transition from IaaS to PaaS. These results outline the specific benefits or drivers that led to the consideration of PaaS, such as increased agility, scalability, cost-efficiency, improved developer productivity and improved time to market **(6.2.1).**
- The expected benefits Dyflexis aims to achieve with transitioning to PaaS **(6.2.2)**
- Identification of key sections for assessing cloud readiness **(6.2.3)**
- The applied migration strategy. The strategies discussed here involved an explanation of the phases the organisation will go through with the application during the migration process. These steps include how the components of the application will be transferred to the new platform **(6.2.4).**

During the document review phase, relevant materials such as organisational documents, technical specifications, architectural plans and past communications provided insights into the motivation behind the transition, required changes, and migration strategy.

The case study gave the difficulties and possibilities particular to Dyflexis a real-world setting and for a more thorough examination of them. The results of the case study integrated well with the data of the interviews with cloud experts. Specifically, emphasizing the important elements SaaS providers need to take into account during such a transition. A comparison of the case study and interviews is stated in the next chapter.

All the findings and results obtained from the conducted case study are thoroughly explained and documented in the case study report of Dyflexis. The document presents and analyses the outcomes, insights, and conclusions derived from the case study investigation.

6.2.1 Drivers behind the Transition

To fully utilise the advantages of the platform model and make informed choices, it is crucial to pinpoint and examine the precise drivers behind the shift to PaaS (L. Van Donkersgoed, personal communication, April 18th, 2023). In the case of Dyflexis, the drivers identified include the ability to quickly respond to customer demands, overcoming limitations posed by infrastructure for rapid business/application growth, and addressing modularization and overprovisioning of resources (P.

Lubbers, personal communication, April 25th, 2023). These drivers align with three key categories: scalability and agility, development and deployment, and cost optimisation.

Dyflexis aims to reap benefits in each of these three key categories. In terms of scalability, during peak periods of high demand for scheduling or time-tracking features, the PaaS platform can automatically allocate additional computing resources to ensure smooth and responsive user experiences (M. Polak, personal communication, March 7th, 2022). In the current Infrastructure scalability and agility require manual provisioning and configuration of additional virtual machines and servers to accommodate increased demand (P. Lubbers, personal communication, March 9th, 2022).

For development and deployment, PaaS offers a more streamlined and efficient environment (Cohen, 2013), enabling Dyflexis to focus on application development rather than spending additional time and resources in managing the underlying infrastructure.

Finally, PaaS provides the flexibility to dynamically allocate resources based on demand, avoiding the need for overprovisioning of infrastructure resources (Yu et al., 2011). This also underlines the pay-as-you-go model of PaaS allowing the organisation to pay only for the resources they actually use.

6.2.2 Benefits Dyflexis

This paragraph describes the benefits Dyflexis aims to achieve transitioning to PaaS.

These advantages empower Dyflexis to deliver a more reliable, scalable, and cost-effective application while enabling its developers to focus on core business logic and innovation. The following benefits are established through interviews with the ICT architect of Dyflexis (P. Lubbers, personal communication, May 16th, 2023).

1. **High user satisfaction**: Moving to PaaS can enhance the user experience by providing a more stable and reliable platform. PaaS providers often handle infrastructure management, ensuring high availability and performance. This leads to improved user satisfaction as the application is consistently accessible and responsive.

2. **Developer productivity:** PaaS platforms offer pre-configured development frameworks, tools, and services that simplify the development process (Shu-Qing and Jie-Bin, 2010). By abstracting away infrastructure concerns, developers can focus more on writing application code and delivering features. This increases developer productivity and allows them to rapidly iterate and innovate.

3. **Scalable opportunities:** PaaS enables Dyflexis to easily scale its application as user demand grows. PaaS providers typically offer automatic scalability features, allowing the application to handle increased workloads without manual intervention (Ardagna et al., 2012). This scalability ensures the system can handle peak usage periods and accommodates business growth effectively.

4. **Event-driven service integration** instead of custom integration solutions: PaaS often provides integration capabilities and pre-built services, including event-driven services and message brokers. Dyflexis can leverage these services to facilitate seamless integration with other systems and applications (Ardagna et al., 2012). Uitilising the event-driven service

integration provided by the PaaS platform reduces the need for custom integration solutions (Van Eyk et al., 2018), saving time and effort in building and maintaining complex integrations.

5. **Simplified development and deployment:** PaaS streamlines the development and deployment processes (Cohen, 2013). With pre-configured environments and managed services, Dyflexis can focus on writing application code and rely on the PaaS provider for handling infrastructure setup, configuration, and deployment. This simplifies the development workflow and reduces the time and effort required for initial setup and ongoing maintenance.

6. **Usage-based billing:** PaaS platforms typically follow a usage-based billing model, charging organisations based on the resources consumed (Mohammed & Zeebaree, 2021). This allows Dyflexis to align costs with actual usage, optimising resource allocation and cost efficiency. It also provides flexibility, as costs can scale up or down based on application demand and usage patterns.

6.2.3 Key sections for assessing cloud readiness

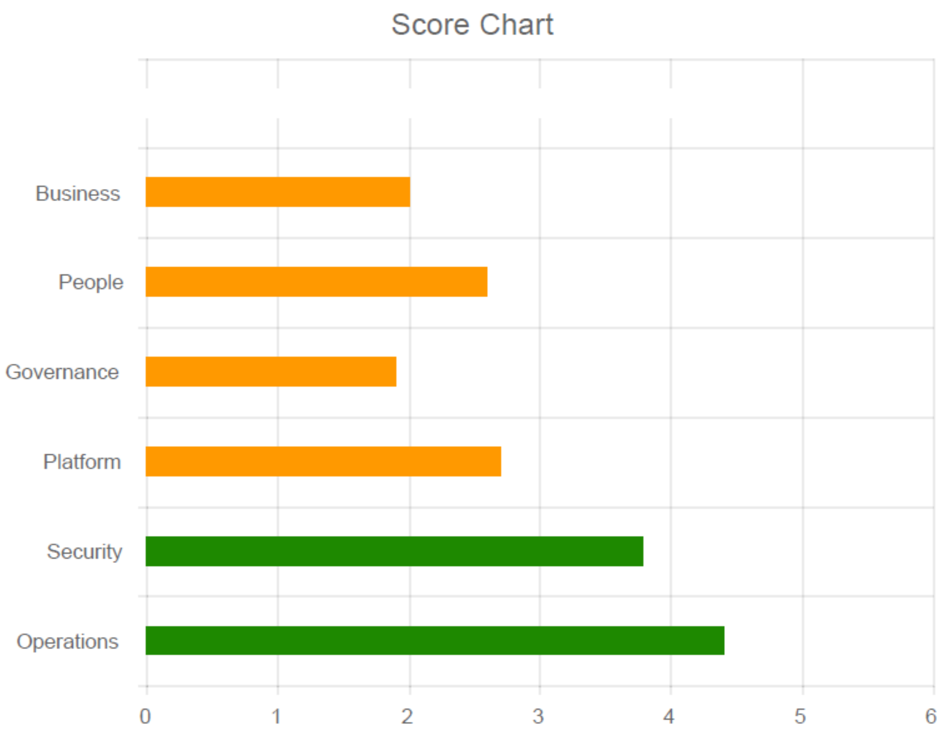
Dyflexis went through an AWS Cloud Adoption Framework (CAF) Assessment, which serves as a vital starting point for companies considering a migration to the public cloud. It helps them understand the benefits and potential risks associated with such a transition, enabling informed decision-making and planning (AWS, 2022).

The AWS readiness assessment serves as a vital starting point for companies considering a migration to the public cloud. It helps them understand the benefits and potential risks associated with such a transition, enabling informed decision-making and planning (AWS, 2022).

As shown in Figure 5, Dyflexis received a relatively low score in the perspectives **Business, People, Governance, and platform.** This suggests that these areas may require further attention and improvement in terms of the AWS cloud readiness assessment conducted for Dyflexis.

On the other hand, Dyflexis obtained a higher score in the perspectives of Security and Operations. This indicates that Dyflexis has demonstrated a relatively stronger readiness in these specific areas.

Figure 5 Results AWS Assessment



Note. By AWS (2023). AWS Cloud readiness assessment

The AWS CAF Assessment provides several perspectives for organisations considering cloud adoption. These perspectives include the areas around Business, People, Governance, Operations, Platform and Security. All areas play a vital role in the overall cloud readiness and migration. However, due to time limitations and the scope of the research, the decision to focus on the perspectives Security and Platform is made to prioritize immediate improvements and mitigate potential risks associated with these aspects of the cloud infrastructure.

Additionally, the areas Platform and Security have common topics that are in line with the scope of this study. For example, the platform engineering section delves into the architecture of the cloud environment. This includes the evaluation of network configurations, infrastructure management processes and Infrastructure as Code (IaC) implementation.

The transition from IaaS to PaaS requires a thorough understanding of the security measures provided by PaaS. These include the following:

- Understanding the characteristics of security responsibilities between Dyflexis and AWS becomes crucial to avoid gaps or overlaps in security controls.
- Examining the security measures around data protection, encryption, and access controls provided by the PaaS services ensures that sensitive data remains adequately protected.
- IAM and network security practices may differ in a PaaS environment compared to IaaS. By examining how these aspects are integrated into the PaaS services, Dyflexis can ensure that access controls and network configurations are appropriately managed.

Security
The security perspective examines the security and privacy considerations associated with cloud adoption. This includes evaluating data protection measures, identity and access management, network security, and compliance requirements.

Dyflexis obtained a score between 3 and 4 in the security section, which indicates a relatively high level of readiness in terms of the assessed security criteria. It suggests that Dyflexis has implemented robust security measures and practices, demonstrating a strong commitment to safeguarding their cloud environment.

While these results indicate a high level of readiness in the security section, it's important to note that here too may still be areas for improvement or specific recommendations provided within the assessment. For example, Dyflexis can strengthen access controls by implementing strict user authentication mechanisms, such as multi-factor authentication, to prevent unauthorized access to sensitive data. Additionally, they can enhance data encryption practices by adopting encryption at rest and encryption in transit techniques, ensuring that data remains protected both while stored and during transmission.

Platform
The platform perspective centers around the technical aspects of cloud adoption. It evaluates an organisation's existing IT infrastructure, applications, and data architecture to determine the best cloud platform and services that align with their requirements.

Dyflexis obtained a score between 2 and 3 in the platform section indicating a moderate level of readiness in terms of the assessed criteria. It suggests that there are areas within the platform aspect that may require further attention, improvements, and optimisation. For example, Dyflexis can invest in platform engineering by optimising their infrastructure to handle increased workloads and ensure high availability. This may involve implementing load balancing mechanisms, optimising network configurations, and leveraging caching techniques to improve performance and scalability.

Recommendation CI/CD pipeline
Another improvement area within the platform section is structuring the overall deployment process. Dyflexis received a low score in this specific area.

Dyflexis can implement CI/CD pipelines using tools like AWS Code Pipeline or Jenkins. This enables them to automate the build, test, and deployment processes, ensuring that code changes are continuously integrated, tested, and deployed to production. This allows for faster feedback loops, reduces the risk of manual errors, and improves overall agility.

6.2.4 Migration Strategies Dyflexis

In the case of Dyflexis, it is the company's opinion to use a combination of migration strategies for their transition. The combination of strategies includes the Lift & Shift, Refactoring + Rebuilding and Retiring. The choice for applying this specific combination lies in the way Dyflexis is structured as an organisation and the functionalities of the application.

For simplicity reasons the steps are divided into the phases below, where each phase describes the actions, the organisation will take.

1. **Assessment and Goal Setting**

- Assess the existing IT infrastructure to determine its suitability for migration to the desired PaaS platform.
- Align migration goals with the overall strategic vision of the business.
- Utilise the Objective and Key Results (OKR) technique to set clear goals and track progress throughout the migration process.
- Identify challenges, such as tight coupled features that will lead to interference and complexity when making changes.

In this first phase the organisation keeps mind that the overall objective is to migrate from a monolith to a more modular architecture.

2. **Lift and Shift**

- Start by migrating the monolithic application to the PaaS environment using the lift and shift approach.
- Package and move the entire application stack, including code, dependencies, and configuration, to the PaaS platform.
- Maintain the current infrastructure layer, allowing Dyflexis to continue managing and configuring the underlying infrastructure.
- Benefit from having the full infrastructure at a single vendor, reducing the cognitive load and enabling the use of vendor platform features directly.

3. **Refactoring and Rebuilding**

- Apply the strangler pattern to gradually refactor the monolithic application into smaller, independent microservices.
- Identify specific components and functionalities within the monolith to extract and convert into separate microservices.
- Develop the microservices using PaaS-specific features and services to leverage the capabilities of the platform. Rebuild certain components based on a cost-benefit analysis, ensuring that the benefits of PaaS are maximized.

In the third chapter of the case study report the backend architecture of the Dyflexis application is broadly described. A notable process within the current backend architecture of the application is the presence of a manual configuration involved in subscribing to and sending events from the event broker.

Based on this manual process it is evident for Dyflexis to adjust its application architecture or modify the application's code to fully leverage the benefits offered by PaaS, such as containerisation and message queuing. By doing so, Dyflexis can effectively utilise these PaaS services, which can enhance scalability, manage resources efficiently, and improve overall performance of the application, instead of relying on manual configurations within the application.

4. **Retiring**

- The Retiring phase focuses on decommissioning or retiring components or functionalities that are no longer needed or have been replaced.
- As the migration progresses and new components are introduced, the older components that have been replaced or seen as obsolete can be safely retired.
- This phase involves carefully planning the retirement process, ensuring that dependencies are managed, data is properly migrated or archived, and the impact on other systems or processes is minimized.
- Retire the original monolithic components as more functionalities are handled by the microservices over time.

6.2.5 Summary

The transition to PaaS can result in benefits such as high user satisfaction, improved developer productivity, and enhanced scalability. However, the case study discovered a risk in the form of a high rate of customer churn, due the switch of cloud provider.

Technical requirements included implementing CI/CD pipelines for ensuring efficient code updates and utilizing Infrastructure as Code, to manage and provision resources in a consistent and automated manner.

Regarding the architectural changes, the case study highlighted the importance of breaking down the monolithic architecture, promoting modularization and flexibility. Additionally, rearchitecting the code base was a necessary step to align with the PaaS environment's architecture and capabilities.

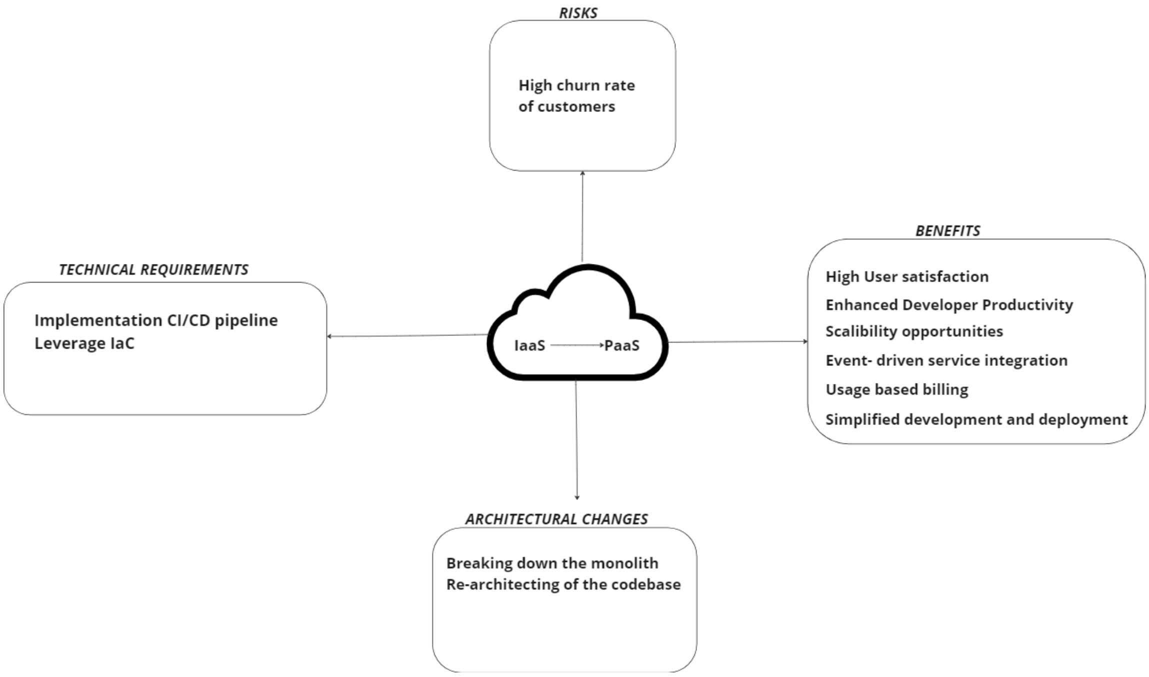


Figure 6 Visualisation case study finding

7 Discussion

This section examines the broader implications and significance of transitioning from IaaS to PaaS, providing valuable insights on the associated risks and challenges.

Additionally, in this discussion comes forward how the shift to PaaS may influence the broader field of cloud computing, specifically how PaaS benefits can enhance the competitiveness and agility of ISVs in the cloud computing environment.
To understand the purpose and effects of this transformation, this section will entail examination of the observed results and the advantages, difficulties, and potential trade-offs related to the transition from IaaS to PaaS. As part of this research, several variables, including scalability, cost-effectiveness, development efficiency, and the overall influence on organisational operations, are examined.

The first paragraph, 7.1, looks at the benefits of both the interview results and case study results and compares this with existing literature. The paragraph that follows, 7.2, reflects on the risks and how this compares to existing articles or frameworks. Finally, the last paragraphs, 7.3 and 7.4, respectively explain the architectural differences and encountered migration strategies.
The table below presents a comparison between interviews and case study showcasing areas of overlap and divergence in the results.

	Benefits	**Risks**	**Technical Requirements**	**Architectural differences**
Interviews	Cost optimisation Scalability and Elasticity Enhanced Developer productivity Infrastructure abstraction Security Enhancements	Learning Curve Data migration Multi-cloud architecture	Long-Term Cloud Decision	Based on company objectives
Case Study	High user satisfaction Enhanced Developer productivity Scalability opportunities Event-driven service integration Usage-based billing Simplified Development and deployment	High churn rate of customers	Implementation CI/CD pipeline Leverage IaC	Breaking down the monolith Re-architecting of codebase

Table 5 Comparison Results Interviews & Case Study

7.1 Benefits of the transition

During the interviews and case study came forward that it is essential to conduct a thorough risk and benefits analysis before undertaking the transition. The potential benefits were clearly discussed and emphasised in both the case study and the interviews. As observed in Table 6, certain benefits emerged during the interviews that were absent in the case study, and conversely, some concepts were present in the case study but not in the interviews.

The benefits common to both the interviews and case study included Cost optimisation, Scalability and Elasticity, Enhanced Developer productivity, Security Enhancements, and Infrastructure abstraction. The results of a successful cloud adoption and deployment is influenced by all these variables.

Nonetheless, there were also distinct benefits in the case study that were not mentioned during the interviews with cloud experts. The benefits that occurred exclusively in the case study are identified as High user satisfaction and Event-driven service integration.

Commercial organisations often provide frameworks and methodologies to assist businesses in evaluating whether to adopt cloud computing or not. However, in addition, the academic world offers valuable tools that contribute to decision-making in technology adoption. One such tool is the TOE framework, described in chapter 4.5, which is widely used in academic research. The TOE framework establishes a solid foundation for exploring and understanding the various aspects that drive the adoption and implementation of technological innovations, such as cloud computing.

The organisational factor of "slack" within the TOE framework, refers to the availability of unused resources within an organisation. By applying the TOE framework and conducting analysis prior to migration, organisations can identify and analyse the existing slack or unused resources more clearly. This analysis helps organisations understand the potential areas where resources can be optimised or reallocated, ensuring more efficient resource utilisation during and after the migration process.

According to Borgman (2013) the technology and organisational context factors of the TOE framework play a bigger and more significant role in the decision-making process of whether organisations adopt cloud computing. Organisations assess factors such as the scalability, flexibility, security, reliability, interoperability, and performance of cloud computing solutions.

Common benefits

These common benefits are mentioned both throughout the interviews and during the case study process. These findings highlight the positive effect a transition from IaaS to PaaS can have on organisations. These consistent benefits are proven reliable due to their appearance in both real-world interviews and in-depth caste study analyses. This, in turn, can be referenced to by future researchers studying this particular topic.

Cost optimisation

During the interviews cost optimisation emerged as a key advantage, as PaaS offers a more streamlined and cost-effective model by eliminating the need for managing infrastructure and providing scalable resources on demand. According to Chavan and Kulkarni (2015), they highlight the cost-saving benefits of PaaS by explaining that it eliminates the necessity of upfront investments and lowers expenses related to hardware procurement and management.

As listed in the previous chapter one of the benefits stated by Dyflexis included usage-based billing, where the organisation will only pay for the resources, they actually use. This benefit leads to potential cost savings, especially during periods of low demand.

Security enhancements

Another fully agreed section experts discussed was that PaaS providers offers better security. Scholars such as Mohammed & Zeebaree (2021) do not specifically focus and compare security aspects of PaaS but emphasize the responsibility PaaS takes away from customers. By relying on the expertise and security measures of PaaS providers, customers can offload some of the security burdens.

Improved scalability and elasticity

In the case study, organisations tend to overprovision resources in relation to the infrastructure.

Both the case study and the interviews delivered overlapping results regarding the scalability and elasticity of PaaS. PaaS providers, such as AWS, provide Load balancing and Auto scaling features for applications that have unpredictable or changing workloads.

Additionally, the acknowledgement of the elasticity features in PaaS can be found in the characteristics of CC described by the NIST (2009):

" Rapid elasticity is another benefit of cloud computing, which enables users to swiftly scale resources up or down in response to shifting demands."

Enhanced developer productivity

The results in the interviews explain that PaaS enhances developer productivity by abstracting infrastructure concerns and enabling focus on application development and innovation. The case study described the same benefit the company aims to achieve through pre-configured PaaS tools.

In chapter 4.2, it is mentioned that Dalbhanjan (2015) discusses in his article how PaaS providers offer various high-level abstraction services that enable developers to concentrate on the development process rather than the underlying infrastructure.

The experts highlighted during the interviews that this results in an enhanced time to market, which is also perceived as a primary motivator for Dyflexis's transition from IaaS to PaaS, as observed in the case study.

Infrastructure abstraction

Chapter 6.1.2 displays a clear distinction between the cloud delivery models. In addition to cloud experts, academics such as Steffen Kachele et al (2013) have also contributed to the understanding of cloud delivery models by characterising a separation based on network, storage, and computing components. In PaaS, these characterisations will all be handled by the PaaS provider, abstracting away the responsibility for the customer.

Computing services primarily focus on providing mechanisms for running applications within the cloud environment. This includes the availability of virtual machines, containers, and serverless computing platforms that allow users to deploy and execute their software applications.

Storage services are essential for persistent data storage within the cloud. They offer dependable and expandable data storage and access options. These services frequently include capabilities like data replication, backup, and durability to guarantee the reliability and accessibility of stored data.

Network services include systems that enables connectivity and data transfer between users, programs, and virtual machines. Some examples are load balancers, virtual private networks (VPNs), firewalls, and other networking elements that assure safe and effective data transfer of network services.

Diverting benefits

Several benefits were identified in the case study but did not clearly come forward during the interviews. These benefits included High user satisfaction and Event driven integration.

High user satisfaction

In the case study the benefit "High user satisfaction" was mentioned because the perspectives of the end customers of Dyflexis was taken into consideration. In the case study High user satisfaction was a result of the driver "The ability to respond to customers' demand quickly".

This benefit demonstrated that after transitioning to PaaS, end-users of Dyflexis will be more satisfied with the enhanced system. This is due to the new services offered by the PaaS platform such as automated scaling, improved application responsiveness and faster deployment of new features.

This example can be linked to previous benefits including scalability and elasticity. The ability to handle increased traffic without compromising performance enhances the overall user experience and satisfaction.

Event driven integration

Event-driven integration is an approach to connecting and coordinating various applications, services, or systems in a distributed environment. This is highlighted more in the case study rather than in the interviews because Dyflexis extensively focused on real-time responsiveness and scalability, whereas the interviews had a broader scope and did not delve deeply into specific benefits.

In paragraph 4.4 Buksh (2015) mentions that to achieve a scalable infrastructure EDA needs to be integrated. When EDA is integrated into a PaaS environment, it can enhance the platform's inherent scalability capabilities. EDA allows for more dynamic and responsive resource allocation based on event-driven triggers.

This powerful architectural pattern, EDA, provides a range of advantages that can create growth opportunities for ISVs and allow them to stay competitive in this evolving technology landscape.

7.2 Risks

Learning curve

During the interviews, several risks associated with transitioning from IaaS to PaaS were identified. The first risk mentioned was the learning curve, which refers to the challenge of acquiring new knowledge and skills required to effectively work with the PaaS environment.

The organisational context of the TOE framework involves the internal factors within an organisation that influence the decision to adopt cloud computing. This includes aspects such as the organisation's IT strategy, resources, skills, and capabilities.

PaaS often introduces new tools and technologies that may require specific skills and capabilities. Evaluating the organisation's current skill set is essential to determine if employees have the expertise to work with the PaaS platform effectively. Training and upskilling may be necessary to bridge any knowledge gaps.

High churn rate of customers

During the case study of Dyflexis not many risks were explicitly identified by the organisation. However, a potential risk emerged regarding the expected churn of customers if Dyflexis decided to migrate to the specific cloud provider AWS.

This emphasizes the significance of assessing and managing the potential risks associated with customer retention and satisfaction during the migration process. It serves as a reminder that customer communication, transparency, and support are crucial elements to address potential concerns and mitigate the risk of customer churn.

Multi-cloud architecture

As mentioned in the previous chapter there has been a notable disagreement among cloud experts regarding the choice between a single cloud architecture and a multi-cloud architecture. Using a single cloud provider can simplify the management and integration of cloud services. Combining cloud services with a single provider result in substantial cost savings through volume discounts and streamlined billing processes (AWS,2019).

Vendor lock-in

On the other hand, relying on multiple cloud providers mitigates the risk of being tied to a single vendor and gives organisations more flexibility to switch providers if necessary. As mentioned by Repschagel et al. (2013) the main obstacles identified for accepting PaaS were concerns related to vendor lock-in and the perceived lack of control over the underlying infrastructure.

While some experts expressed concerns about the learning curve, vendor lock-in, and limited customisation, others had a more optimistic view and believed that these risks could be effectively managed. These differing perspectives emphasize the complexity of evaluating risks in such transitions and highlight the importance of considering a variety of viewpoints and experiences when making decisions about adopting PaaS.

7.3 Technical requirements

During the interviews the experts did not offer conclusive technical requirements organisations should have when migrating to PaaS. This sheds light on the fact that migrating to PaaS is a complex process that can vary significantly based on the organisation's unique needs, existing IT infrastructure, and business objectives.

Many of the technical requirements for ISVs became clear during the case study, rather than in the interviews. These requirements included establishment of a robust CI/CD pipeline and leveraging Infrastructure as Code.

CI/CD pipeline

Another requirement highlighted in the case study is the importance of implementing a Continuous Integration/Continuous Deployment (CI/CD) pipeline. This involves automating the build, testing, and deployment processes to ensure a streamlined and efficient software delivery lifecycle within the PaaS environment.

During the interviews, the establishment of a Robust CI/CD pipeline was mentioned, although it was not further elaborated upon as the experts emphasized that its implementation and specifics are highly dependent on the unique circumstances of each case. This suggests that the design and implementation of a CI/CD pipeline should be customised to meet the organisation's specific needs and requirements when migrating to PaaS.

Leveraging IaC

Finally, the case study also highlighted the significance of leveraging infrastructure as code (IaC). This approach involves managing and provisioning infrastructure resources programmatically using code, allowing for greater consistency and scalability in the deployment and configuration of the PaaS environment. With IaC, changes to the infrastructure are version-controlled, allowing for easy rollback by reverting to a previous version of the code (Operational Excellence - AWS Well-Architected Framework, 2020). The evaluation of these technological factors helps organisations determine if cloud computing aligns with their technical requirements and if the benefits outweigh any potential risks or drawbacks.

7.4 Architectural differences

During the interviews, the experts did not provide a detailed description of an ideal architecture or any changes in architecture when transitioning to PaaS. In the case study, on the other hand, steps were outlined to split the monolithic application into loosely coupled services.

Interviews

In chapter 6.1.2 the responsibility of the infrastructure is highlighted. While infrastructure abstraction itself was not considered a direct benefit, it does serve to achieve certain benefits and facilitate architectural changes.

In the traditional IaaS environment, ISVs are responsible for managing the underlying operating system, including tasks such as patching, updates, and security configurations. However, when moving to PaaS, operating system management becomes abstracted by the platform provider.

The degree of ownership and control over infrastructure components is the primary area where IaaS and PaaS differ from one another in terms of accountability. In PaaS, the provider abstracts more of the infrastructure-related activities, enabling the client to concentrate on the development and deployment of applications (Mohammed & Zeebaree, 2021). In IaaS, the customer is more responsible for managing and protecting the infrastructure.

Case Study

Breaking down the monolith

In the case study report it is described that both a monolithic architecture and loosely coupled components are included in the Dyflexis application. This means that the application consists of a combination of tightly integrated components within a monolithic structure, as well as loosely coupled components that operate independently.

By breaking down the monolithic application each service can be developed, deployed, and managed independently, allowing for faster iteration and innovation.

This decoupling also enables the application to take advantage of the scalability features provided by the PaaS platform, such as automatic scaling and load balancing, which can optimise resource utilisation and enhance performance.

Furthermore, it is demonstrated that monolithic designs make it difficult to grow or change individual sections separately since all parts and functionalities are strongly integrated. Thus, when transitioning to a more scalable and flexible environment like PaaS, it is beneficial to move away from the monolithic approach and adopt a loosely coupled component-based architecture.

Finally, the split into loosely coupled services facilitates easier maintenance and updates. Instead of making changes to the entire monolith, developers can focus on specific services, reducing the risk of unintended consequences and enabling more efficient testing and deployment processes.

NIST (2009) suggests that the concept of having loosely coupled services, which are independent and modular components, running on an agile and scalable infrastructure is a key idea behind cloud computing. They further suggest that this architecture has the potential to transform enterprises by enabling them to become interconnected nodes in the cloud.

The move to loosely coupled services aligns with the principles of and Event Driven and microservices architecture, which also promotes modularity, resilience, and scalability. This architecture allows

Dyflexis and other ISVs to take advantage of PaaS offerings that are designed to support microservices-based applications.

If the ISV's application uses an EDA, the move to PaaS should take into account the platform's event processing capabilities. To efficiently handle and process events, the application needs to communicate with the platform's event brokers, messaging services, or event hubs. This involves setting up event subscriptions, controlling event delivery assurances, and enhancing event-driven processes in accordance with the capabilities of the platform (Cugola et al., 2001).

Reengineering of the application

The case study identified the need for ISVs to reengineer their applications to align with the PaaS environment, if certain components of the application are integrated in a way that prevents the effective utilisation of PaaS services. In such cases, reengineering of the application's codebase becomes imperative to leverage the full capabilities and benefits offered by the platform. According to Pahl (2015) reengineering the existing codebase to adapt to the cloud environment is one of the primary concerns during the migration to the cloud.

7.5 Migration strategies

The case study employed a combination of migration strategies to facilitate their migration from Iaas to PaaS. The Dyflexis case study is unique since it primarily focuses on the transition from IaaS to PaaS, in contrast to many publications in the literature that start with an on-premises starting point.

This differentiation highlights the importance of adapting migration strategies based on the organisation's existing cloud environment and specific migration goals.

Additionally, Dyflexis integrated in their cloud migration strategy, the 5-phase model, with specific emphasis on the Feasibility Study and Requirement Analysis & Migration Planning phases.

For example, in the case study report the company combined prerequired steps within the migration process, such as determining long term business goals based on their Objective Key Results (OKRs), assessing the IT infrastructure and the alignment with the desired platform.

This mix-and-match strategy allows them to leverage the benefits of different migration methods, aligning each application with the most suitable approach for a successful transition.

8 Conclusion

While the public cloud offers a rich feature set and benefits in terms of PaaS, there are certain scenarios where businesses may be hesitant to give up full control over their data and infrastructure due to compliance, data retention, or cost considerations (Heachert et al., 2021). In addition, the cloud experts highlighted that the choice between adopting PaaS services or maintaining more control over infrastructure and data should be based on several key factors, including specific business objectives, long-term cost considerations, IT Skills and Expertise and cloud strategy alignments.

The interviews acknowledged that by embracing PaaS organisation stand to gain increased scalability, agility and cost-efficiency. Another notable observation during the interviews is that all the experts unanimously agreed on the benefits of PaaS. And that the opinions differed more when it came to identifying the potential risks associated with transitioning to PaaS.

The other method used in this research, case study, demonstrated a collection of results. When compared to the interviews, the identified benefits were consistent, while the risks were not specifically identified. This difference could be linked to the uniqueness of the transition from IaaS to PaaS. Other unique findings during the case study included the organisation's strategic decision to leverage a combination of migration strategies and architectural changes to split the monolith into smaller independent services, leaning towards a more Event Driven Architecture.

During this research it became clear that there are many factors which need to be taken into consideration when transitioning to PaaS. The term readiness is used widely in the adoption of cloud computing and relates to an organisation's ability and readiness to effectively incorporate new procedures, technologies, and adjustments related to cloud computing (Loebbecke et al., 2012).

For particular situations, such as the current, organisations may find it necessary to use a combination of academic frameworks such as TOE to carry out thorough evaluations of their level of cloud adoption readiness. Additionally, commercial solutions like AWS CAF can be used to accelerate the planning and execution of the cloud adoption process. This involves assessing how well the organisation's IT architecture, and current capabilities match with the goals of cloud adoption. By using both academic and commercial resources businesses can gain valuable insights and guidance to ensure a well-informed and successful transition to the cloud, specifically PaaS.

This research proved that although both IaaS and PaaS provide flexibility and scalability opportunities, the management of it tremendously differs. IaaS gives the user more control and requires them to either manually scale resources or set up auto-scaling policies. When using PaaS, the platform provider manages elasticity by dynamically scaling the resources in response to application demand (Mohammed & Zeebaree, 2021). Users can focus more on application development because of PaaS's responsibility shift, which additionally reduces the operational pressure of infrastructure administration.

Overall, this study explained that the outcome of a transition from IaaS to PaaS strongly depends on careful planning, a detailed evaluation of the current infrastructure and a clear understanding of the organisation’s needs. By exploring these areas the study explained that organisations are able to identify benefits including scalability, cost optimisation and security enhancements.

On the other hand, the study identified areas which may create risks for organisations that are planning on migrating to PaaS. These risks include the learning curve associated with this new platform, the presence of a single cloud provider and the data migration risks during the transition.

Other considerations that came forth when migrating from IaaS to PaaS was the current infrastructure of the application. Specifically, the process of how events or components communicate with each other and the dependency between them. As mentioned throughout this study, factors strongly depend on the objectives of the business and the characteristics of the application. The study emphasizes that breaking down a large application into smaller more manageable components is necessary to fully leverage PaaS. However, in order to determine how much loose coupling and cohesion is required (Abreu & Goulão, 2001), is dependent on the specific case.

8.1 Future research

Both interviews and the case study have delivered valuable insights about the challenges, benefits, and considerations involved in the transition. Future research in this field should take the following into consideration to determine whether the findings are supported by evidence and represent practical outcomes.

To determine the existence of specific technical requirements and the influence of architectural changes on transitioning to the public cloud, it is advised to conduct multiple case studies across various industries. Different industries might have distinct characteristics, compliance requirements, and challenges when it comes to cloud adoption.

Therefore, the examination of multiple case studies allows for a deeper understanding of factors that influence transitions across different industries. Furthermore, analysing multiple case studies can uncover industry boundaries which in return might lead to generalizable strategies or industry specific approaches.

Another point to consider is expanding the pool of interview candidates. By interviewing stakeholders from various departments, such as finance, operations, or management, researchers can gain insights into the broader organisational perspective. These stakeholders can provide valuable insights into financial implications, resource allocation, organisational readiness, and the overall impact on the business.

For example, cost can be a crucial factor in any technology transition. Interviewing stakeholders involved in financial decision-making allows for a deeper understanding of the cost implications associated with transitioning to higher cloud abstractions.

8.2 Limitations

While this research involving ten (9) interviews and a case study provided valuable insights, it is important to acknowledge its limitations. Some of the limitations include the following:

1. Resource Constraints: Conducting interviews and case studies requires time, effort, and resources. With limited resources, it may not be feasible to gather data from a larger sample size or conduct multiple case studies, which could have provided more comprehensive insights.

2. Difficulty in Establishing Causality: The small sample size and qualitative nature of the research may make it challenging to establish causality between variables or identify clear cause-and-effect relationships. The findings may provide correlations or associations but may not demonstrate a definitive causal link.
   For example, the research might find a positive relationship between adopting PaaS and improved efficiency in software development, but it cannot definitively establish that PaaS adoption directly causes the observed improvements. Additionally, due to the small sample size, it may be challenging to claim that these benefits apply universally.

3. Contextual Specificity: The findings from a single case study may be specific to that context, making it challenging to generalize the results to other settings or organisations. The uniqueness of the case study may limit the transferability of the findings to different scenarios. For example, in the workforce management sector, where Dyflexis works, there may be particular compliance rules, labor laws, or industry standards. These industry-specific criteria would need to be taken into account when moving to PaaS, and the platform would need to be checked to make sure it does.

Acknowledgement

As they say a thesis is one of the hardest things…and yet it is done many times.

First off, I would like to extend my gratitude to both Drs. Bas Kruiswijk and Dr. Werner Heijstek for their valuable contributions, specifically their critical feedback and structured guidance. Additionally, their quick responses and professionalism have made this journey enlightening and rewarding.

Furthermore, my appreciation goes out to the entire Dyflexis team for accommodating me within their busy yet remarkably flexible environment. I am thankful for Pascal Lubbers, whose technical knowledge and experience has helped me find clarification in both technical and strategic topics throughout this project.

I also wish to acknowledge Michael Abspoel for his role in connecting me with the cloud experts who have contributed greatly to this study.

A final appreciation goes out to you, as reader, whether you are just skimming or digging deep, I hope that you found both delight and knowledge inside these pages. May this thesis incite interest, raise questioning, and leave a legacy of newly acquired wisdom.

Kindly,

Jamal Thijssen

References

Li, A., Yang, X., Kandula, S., & Zhang, M. (2010, November). CloudCmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (pp. 1-14). https://dl.acm.org/doi/abs/10.1145/1879141.1879143

Wulf, F., Lindner, T., Strahringer, S., & Westner, M. (2021). IaaS, PaaS, or SaaS? The Why of Cloud Computing Delivery Model Selection: Vignettes on the Post-Adoption of Cloud Computing. In *Proceedings of the 54th Hawaii International Conference on System Sciences, 2021* (pp. 6285-6294).https://opus4.kobv.de/opus4-oth-regensburg/frontdoor/index/index/docId/791

Cohen, B. (2013). PaaS: New opportunities for cloud application development. *Computer*, *46*(09), 97-100. https://www.computer.org/csdl/magazine/co/2013/09/mco2013090097/13rRUB6SpSY

Ahmad, N., Naveed, Q. N., & Hoda, N. (2018, November). Strategy and procedures for Migration to the Cloud Computing. In 2018 IEEE 5th international conference on engineering technologies and applied sciences (ICETAS) (pp. 1-5). IEEE. https://ieeexplore.ieee.org/abstract/document/8629101

Dantas, J., Khazaei, H., & Litoiu, M. (2022, July). Application Deployment Strategies for Reducing the Cold Start Delay of AWS Lambda. In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)* (pp. 1-10). IEEE. https://ieeexplore.ieee.org/abstract/document/9860368

Quint, P. C., & Kratzke, N. (2016). Overcome vendor lock-in by integrating already available container technologies towards transferability in cloud computing for smes. Proceedings of CLOUD COMPUTING, 2016. https://www.researchgate.net/publication/292980185_Overcome_Vendor_Lock-In_by_Integrating_Already_Available_Container_Technologies_-_Towards_Transferability_in_Cloud_Computing_for_SMEs

Loebbecke, C., Thomas, B., & Ullrich, T. (2012). Assessing Cloud Readiness at Continental AG. *MIS Quarterly Executive*, *11*(1). https://www.researchgate.net/publication/290309238_Assessing_cloud_readiness_at_continental_AG

Niknejad, N., Ismail, W., Ghani, I., Nazari, B., & Bahari, M. (2020). Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation. Information Systems, 91, 101491. https://www.sciencedirect.com/science/article/abs/pii/S0306437920300028

Ogunlolu, I., & Rajanen, D. (2019). Cloud computing adoption in organizations: a literature review and a unifying model. https://aisel.aisnet.org/isd2014/proceedings2019/ManagingISD/4/

Opara-Martins, J., Sahandi, R., & Tian, F. (2014, November). Critical review of vendor lock-in and its impact on adoption of cloud computing. In *International conference on information society (i-Society 2014)* (pp. 92-97). IEEE. https://ieeexplore.ieee.org/abstract/document/7009018

Raines, G. (2009). Cloud computing and SOA. MITRE, white paper, Oct. https://apps.dtic.mil/sti/citations/AD1107392

Carrasco, J., Durán, F., & Pimentel, E. (2018). Runtime migration of applications in a trans-cloud environment. In *Service-Oriented Computing–ICSOC 2017 Workshops: ASOCA, ISyCC, WESOACS, and Satellite Events, Málaga, Spain, November 13–16, 2017, Revised Selected Papers* (pp. 55-66). Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-319-91764-1_5

Rai, R., Sahoo, G., & Mehfuz, S. (2015). Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration. SpringerPlus, 4, 1-12.

Linthicum, D. S. (2017). Cloud-native applications and cloud migration: The good, the bad, and the points between. *IEEE Cloud Computing*, *4*(5), 12-14. https://ieeexplore.ieee.org/abstract/document/8125545

Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. Automation in Construction, 122, 103441. https://www.sciencedirect.com/science/article/pii/S0926580520310219

Abdelzahir Abdelmaboud , Dayang N.A. Jawawi , Imran Ghani , Abubakar Elsafi , Barbara Kitchenham , Quality of Service Approaches in Cloud Computing: A Systematic Mapping Study, The Journal of Systems & Software (2014), doi: 10.1016/j.jss.2014.12.01 https://ieeexplore.ieee.org/abstract/document/9337100

H. P. Borgman, B. Bahli, H. Heier and F. Schewski, "Cloudrise: Exploring Cloud Computing Adoption and Governance with the TOE Framework," 2013 46th Hawaii International Conference on System Sciences, Wailea, HI, USA, 2013, pp. 4425-4435, doi: 10.1109/HICSS.2013.132. https://ieeexplore.ieee.org/abstract/document/6480378

Saraswat, M., & Tripathi, R. C. (2020, December). Cloud computing: Comparison and analysis of cloud service providers-AWs, Microsoft and Google. In 2020 9th international conference system modeling and advancement in research trends (SMART) (pp. 281-285). IEEE.

Dalbhanjan, P. (2015). Overview of deployment options on aws. *Amazon Whitepapers*. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf

Lee, S. Y., Tang, D., Chen, T., & Chu, W. C. C. (2012, July). A QoS assurance middleware model for enterprise cloud computing. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops* (pp. 322-327). IEEE. https://ieeexplore.ieee.org/abstract/document/6341595

Lan, L., Wang, B., Zhang, L., Shi, R., & Li, F. (2015). An Event-driven Service-oriented Architecture for the Internet of Things Service Execution. International Journal of Online Engineering, 11(2). http://dx.doi.org/10.3991/ijoe.v11i2.3842

Z. A. Bukhsh, M. van Sinderen and P. M. Singh, "SOA and EDA: A comparative study: Similarities, differences and conceptual guidelines on their usage," 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), Colmar, France, 2015, pp. 213-220. https://ieeexplore.ieee.org/abstract/document/7517937

A. Khajeh-Hosseini, D. Greenwood and I. Sommerville, "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, USA, 2010, pp. 450-457, doi: 10.1109/CLOUD.2010.37. https://ieeexplore.ieee.org/abstract/document/5557962

Freet, D., Agrawal, R., John, S., & Walker, J. J. (2015, October). Cloud forensics challenges from a service model standpoint: IaaS, PaaS and SaaS. In *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems* (pp. 148-155). https://dl.acm.org/doi/abs/10.1145/2857218.2857253

Zhelev, S., & Rozeva, A. (2019, November). Using microservices and event driven architecture for big data stream processing. In AIP Conference Proceedings (Vol. 2172, No. 1, p. 090010). AIP Publishing LLC. https://pubs.aip.org/aip/acp/article/2172/1/090010/598251/Using-microservices-and-event-driven-architecture

Lehrig, S., Eikerling, H., & Becker, S. (2015, May). Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics. In Proceedings of the 11th international ACM SIGSOFT conference on quality of software architectures (pp. 83-92). https://dl.acm.org/doi/abs/10.1145/2737182.2737185

Shared Responsibility Model. (n.d.-b). https://aws.amazon.com/compliance/shared-responsibility-model/

Kächele, S., Spann, C., Hauck, F. J., & Domaschka, J. (2013, December). Beyond IaaS and PaaS: An extended cloud taxonomy for computation, storage and networking. In 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (pp. 75-82). IEEE. https://ieeexplore.ieee.org/abstract/document/6809342

Ahmad, M. I. (2015). Unified theory of acceptance and use of technology (UTAUT). *LinkedIn Pulse*, 179-211.

Pogarcic, I., Krnjak, D., & Ozanic, D. (2012). Business Benefits from the Virtualization of an ICT Infrastructure. *International Journal of Engineering Business Management*, *4*(Godište 2012), 4-42. https://hrcak.srce.hr/160693

Rogers, E. M. (2002). Diffusion of preventive innovations. *Addictive behaviors*, *27*(6), 989-993. https://www.sciencedirect.com/science/article/abs/pii/S0306460302003003

Muñoz-Escoí, F. D., & Bernabéu-Aubán, J. M. (2017). A survey on elasticity management in PaaS systems. Computing, 99, 617-656. https://link.springer.com/article/10.1007/s00607-016-0507-8

G. Cugola, E. Di Nitto and A. Fuggetta, "The JEDI event-based infrastructure and its application to the development of the OPSS WFMS," in IEEE Transactions on Software Engineering, vol. 27, no. 9, pp. 827-850, Sept. 2001, doi: 10.1109/32.950318. https://ieeexplore.ieee.org/abstract/document/950318

Cruz-Jesus, F., Pinheiro, A., & Oliveira, T. (2019). Understanding CRM adoption stages: empirical analysis building on the TOE framework. *Computers in Industry*, *109*, 1-13. https://www.sciencedirect.com/science/article/abs/pii/S0166361518305438

Chavan, P., & Kulkarni, G. (2013). PaaS cloud. International Journal of Computer Science and Information Security (IJCSIS), 1(1). https://www.researchgate.net/publication/258255287_PaaS_Cloud?enrichId=rgreq-aa290ef76fa0b8ff11daeafef381cf55-

XXX&enrichSource=Y292ZXJQYWdlOzI1ODI1NTI4NztBUzo5ODc4Mzg0NTE1ODkxM0AxNDAwNTYzMzEyMjE1&el=1_x_2&_esc=publicationCoverPdf

Palankar, M. R., Iamnitchi, A., Ripeanu, M., & Garfinkel, S. (2008, June). Amazon S3 for science grids: a viable solution?. In *Proceedings of the 2008 international workshop on Data-aware distributed computing* (pp. 55-64). https://dl.acm.org/doi/abs/10.1145/1383519.1383526

Mell, P., & Grance, T. (2009). Effectively and securely using the cloud computing paradigm. NIST, Information Technology Laboratory, 2(8), 304-311.

Operational Excellence - AWS Well-Architected Framework. (2020, July 2). https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.pillar.operationalExcellence.en.html

C. Pahl and H. Xiong, "Migration to PaaS clouds - Migration process and architectural concerns," *2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems*, Eindhoven, Netherlands, 2013, pp. 86-91, doi: 10.1109/MESOCA.2013.6632740. https://ieeexplore.ieee.org/abstract/document/6632740

Ansar, M., Ashraf, M. W., & Fatima, M. (2018). Data Migration in Cloud: A Systematic Review. *Am. Sci. Res. J. Eng. Technol. Sci.(ASRJETS)*, *48*, 73-89. http://asrjetsjournal.org/

Fowley, F., Elango, D. M., Magar, H., & Pahl, C. (2018). The benefits of using experimental exploration for cloud migration analysis and planning. In Cloud Computing and Service Science: 7th International Conference, CLOSER 2017, Porto, Portugal, April 24–26, 2017, Revised Selected Papers 7 (pp. 157-176). Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-319-94959-8_9

Heuchert, S., Rimal, B. P., Reisslein, M., & Wang, Y. (2021). Design of a small-scale and failure-resistant IaaS cloud using OpenStack. Applied Computing and Informatics. https://www.emerald.com/insight/content/doi/10.1108/ACI-04-2021-0094/full/html

Charmaz, K., & Belgrave, L. (2012). Qualitative interviewing and grounded theory analysis. *The SAGE handbook of interview research: The complexity of the craft*, *2*, 347-365. https://books.google.nl/books?hl=en&lr=&id=VCFsZsvZdwkC&oi=fnd&pg=PA347&dq=charmaz+and+belgrave+grounded+theory&ots=aR_gQyJvGe&sig=yhBVEdqjp4-o3fAZYrSJayNN54U&redir_esc=y#v=onepage&q=charmaz%20and%20belgrave%20grounded%20theory&f=false

Yu, H., Li, X., Zhang, S., & Shi, Y. (2011, October). An adaptive approach to resource provisioning in PaaS. In *2011 Eighth Web Information Systems and Applications Conference* (pp. 52-57). IEEE. https://ieeexplore.ieee.org/abstract/document/6093602

Z. Shu-Qing and X. Jie-Bin, "The Improvement of PaaS Platform," *2010 First International Conference on Networking and Distributed Computing*, Hangzhou, China, 2010, pp. 156-159, doi: 10.1109/ICNDC.2010.40. https://ieeexplore.ieee.org/abstract/document/5645419

C. A. Ardagna, E. Damiani, F. Frati, D. Rebeccani and M. Ughetti, "Scalability Patterns for Platform-as-a-Service," *2012 IEEE Fifth International Conference on Cloud Computing*, Honolulu, HI, USA, 2012, pp. 718-725, doi: 10.1109/CLOUD.2012.41. https://ieeexplore.ieee.org/abstract/document/6253571

E. van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uță and A. Iosup, "Serverless is More: From PaaS to Present Cloud Computing," in *IEEE Internet Computing*, vol. 22, no. 5, pp. 8-17, Sep./Oct. 2018, doi: 10.1109/MIC.2018.053681358. https://ieeexplore.ieee.org/abstract/document/8481652

Appendices

Appendix 1 : Background interviewees

Interviewee 1	
Position	Lead Engineer
Organisation	PostNL
Industry	Logistics
Experience with cloud computing	This expert has many experiences with migrations, specifically from on-premise to IaaS. This individual has demonstrated exceptional technical expertise and a strong commitment to sharing their knowledge and expertise with the AWS community, making him one of a select few AWS Heroes in The Netherlands.
Technical knowledge	Cloud Computing, Operations Management, Software Development, Project Management, Mobile Applications, Network Security.
Lessons Learned	Cloud computing is essentially distinguishable by Compute, Storage and Network.

Interviewee 2	
Position	ICT Specialist
Organisation	Dyflexis
Industry	Workforce Management Software
Experience with cloud computing	This individual is an expert in Javascripting, back-end technologies and infrastructure design.
Technical knowledge	Cloud Computing, Operations Management, Software Development, Project Management, Customer Service, Process Improvement, Network Architecture
Lessons Learned	The higher the level of abstraction, the more extensive the use of services for containerisation and orchestration.

Interviewee 3	
Position	Developer
Organisation	Dyflexis
Industry	Workforce Management Software
Experience with cloud computing	This individual has some experiences with migrations from on-premises to Private clouds. His responsibilities within previous migrations were associated with workload prioritization.
Technical knowledge	Cloud Computing, Operations Management, Java, Backend Development
Lessons Learned	The technical decisions are made based on the direction of where the business intends to go.

Interviewee 4	
Position	Devops Engineer
Organisation	Chantelle Digital lab
Industry	Apparel
Experience with cloud computing	This individual has many experiences with AWS migrations and is an expert in creating and managing cloud infrastructures of web applications.
Technical knowledge	Cloud Computing, Software architect, Java, Backend Development, Data Analytics, Platform optimisation.
Lessons Learned	For every business issue, there is almost always a technical solution.

Interviewee 5	
Position	Engineering manager
Organisation	Dyflexis
Industry	Workforce Management Software
Experience with cloud computing	This individual is a highly skilled and experienced full stack lead developer, using the latest technologies to create innovative solutions for a wide variety of business problems, including previous migrations within the current organisation.

Technical knowledge	Object Oriented Programming, Devops, Ansible, AWS, JavaScript, Docker, Domain Driven Design, Software architecting
Lessons Learned	Cloud solutions are essential for providing the necessary resources and infrastructure to handle the intricacies of complex software systems while ensuring optimal performance and usability for end-users.

Interviewee 6	
Position	Senior infrastructure Project Manager
Organisation	Shell
Industry	Energy
Experience with cloud computing	This person is a strategic IT specialist with more than 25 years of broad, global experience in managing big-scale projects. directing varied project teams while providing architectural control to meet commercial objectives. Driven by the need for quicker cloud and hybrid workload transitions, DevOps adoption methods.
Technical knowledge	IT Strategy, Service Delivery, Strategy, Cloud Computing, Data Center.
Lessons Learned	Cloud computing relies on a cost benefit analysis to determine the advantages of migrating to cloud-based solutions. This analysis helps organisations evaluate the potential cost savings, scalability, and efficiency gains compared to traditional on-premises infrastructure.

Interviewee 7	
Position	Cloud Operations Engineer
Organisation	EVision Industry Software
Industry	Enterprise software
Experience with cloud computing	This AWS Solution Architect has an abundance of experience in creating extremely effective cloud systems and carrying out successful migrations. His skills include negotiating challenging migration procedures, creating scalable architectures, and putting into practice successful tactics to optimise cloud settings for increased productivity and performance.
Technical knowledge	Cloud Computing, Application Administration, Linux System Engineer, Platform optimisation, SQL, Virtualization.
Lessons Learned	With its capacity to analyse massive volumes of data, automate procedures, and make wise

	judgments, artificial intelligence (AI) is set to play a big role in cloud solutions and revolutionize several sectors. Organisations may achieve new levels of productivity, creativity, and personalisation to advance their businesses by uitilising the potential of AI in cloud settings.

Interviewee 8	
Position	Software Engineer
Organisation	Event Store
Industry	Technology
Experience with cloud computing	This developer has a solid understanding of cloud technologies and architectures, he has successfully designed and implemented cloud-based solutions for various projects. His proficiency in leveraging cloud services, such as virtualization, containerisation, and serverless computing, enables him to optimise performance, scalability, and cost-efficiency in cloud environments.
Technical knowledge	Cloud Computing, Software architect, Java, Backend Development, Data Analytics, Platform optimisation.
Lessons Learned	It is imperative to focus on the significance of user-centric design and continuous feedback loops to continuously improve and refine cloud-based applications.

Interviewee 9	
Position	Senior Devops cloud engineer
Organisation	Nike
Industry	Retail
Experience with cloud computing	This person has practical expertise configuring AWS Single Sign-On (SSO) to work with the organisation's Google Account to streamline user access management. Using CI/CD services, they have successfully managed a faster deployment process for several projects, providing seamless changes from staging to production settings. They have also shown adept at uitilising Terraform to create and maintain the business' AWS infrastructure, concentrating on developing a secure network and launching various goods.

Technical knowledge	Cloud Computing, Software architect, Java, Backend Development, Data Analytics, Platform optimisation, Kubernetes, CI/CD.
Lessons Learned	Transitioning to a higher level of abstraction in cloud computing, such as PaaS, can limit certain aspects of flexibility, but it also opens opportunities for growth and innovation within the organisation. While some flexibility may be sacrificed, the potential for leveraging advanced platform capabilities and focusing on value-added activities can lead to increased agility and the exploration of new business opportunities.

Appendix 2: Interview questions

Demographic questions

1. How many years of experience do you have in the cloud industry?
2. In which industry have you gained experience with cloud migrations?
3. What has been your role(s) in the project(s) associated with cloud migrations?
4. What is your current job title/role in the cloud domain?

Risk and benefits

5. Can you share an example of a challenging cloud project or migration you have worked on?
6. Did your company identify a compelling business reason for moving to a PaaS platform?
7. From your expertise, what are the key challenges/risks that organisations typically face during the transition from IaaS to PaaS?
8. How can they mitigate those risks?
9. What benefits did the company expect to achieve from this cloud transition?
10. What considerations should organisations keep in mind when evaluating whether a transition from IaaS to PaaS is the right choice for their specific needs and objectives?

Technical requirements

11. Can you share examples of specific tools or technologies you have found valuable in the transition from IaaS to PaaS
12. What are the key considerations for organisations when choosing between public, private, or hybrid cloud solutions?
13. What are the essential factors to evaluate when selecting a cloud service provider?
14. How does cloud computing facilitate scalability and high availability for applications?
15. What are the emerging trends and advancements in cloud computing that organisations should pay attention to?
16. what are the key technical requirements you consider to ensure compatibility and seamless integration with existing systems and applications?
17. Can you provide insights into the process of identifying and addressing any potential security vulnerabilities or compliance considerations when migrating from IaaS to PaaS?

Architectural differences

18. How do you approach the rearchitecting process to align the application with the technical capabilities and limitations of the PaaS environment?
19. Are there any best practices or methodologies you follow?
20. How does the shift from IaaS to PaaS impact the application architecture? (Particularly in terms of scalability, fault tolerance, and performance optimisation)
21. What architectural considerations should organisations keep in mind when designing applications for a PaaS environment?
22. Are there any specific architectural patterns or design principles that are more commonly used in PaaS environments compared to IaaS?

Appendix 3: Case study questions

1. What does your company aim to achieve with this transition?
2. How much do you trust the hosting provider with your data?
3. Are your end-users compliant with this transition? (in terms of Public cloud, Cloud Vendor)
4. What are the potential risks or trade-offs associated with the transition from IaaS to PaaS?
5. How do you plan to mitigate those risks and ensure minimal disruption to the existing operations?
6. Can you describe the current IaaS infrastructure in terms of its scale, components, and key functionalities?
7. How does the shift to PaaS affect the development of microservices-based architectures compared to traditional monolithic architectures?
8. How do you envision the new PaaS architecture improving the efficiency, scalability, or performance of the backend process compared to the current setup?
9. What are the requirements for monitoring and managing the platform?
10. What are the limitations of the platform that may impact future expansion?

Appendix 4: Case study report

Case Study Report Dyflexis

By Jamal Thijssen

July 14, 2023

Abstract

Assessments help businesses make wise decisions, reduce risks, and improve their migration strategy (Dan Sullivan, 2013). The Dyflexis case study report details the subsequent migration processes performed to switch from IaaS (Infrastructure as a Service) to PaaS (Platform as a Service) and includes the results of the AWS readiness assessment. The goal of this report is to present a real-world example and gather insights into the risks, advantages, technological needs, and architectural differences involved with moving from IaaS to PaaS. The results of the hands-on assessment focus on whether the firm is prepared to utilise the cloud. To simplify the process within the time and scope provided, the study focused on the platform and security areas while excluding business people and governance issues.

Thoroughly evaluating the potential costs and benefits associated with the transition is crucial before making any decisions (J. Crumley, personal communication, April 5, 2023). By considering both the financial aspects and the technical expertise, organisations can make informed decisions and maximize the benefits of transitioning to PaaS.

The choice to switch from IaaS to PaaS was influenced by several important variables. The automated resource provisioning and scaling capabilities of PaaS were one of Dyflexis' key motivations as it looked to decrease resource waste, optimise costs, and remove human management tasks. Enhancing developer productivity was another motivation, since it would free up the team to concentrate on application development and innovation rather than infrastructure administration. Dyflexis wants to give developers a simplified and effective platform, resulting in increased productivity and quicker software development, therefore it is planning on switching to a PaaS environment. The Dyflexis case study report contributes to a greater knowledge of the advantages and motivations for moving to PaaS by providing insightful information on the assessment findings, migration procedures, and driving forces.

Table of Contents

Introduction

The case study report begins with describing Dyflexis's current IaaS infrastructure, with a particular focus on the backend. The report highlights the drivers behind Dyflexis's decision to transition to a higher abstraction level, such as the need to respond rapidly to customer demands and the scalability opportunities offered by PaaS.

The report further discusses the AWS readiness assessment conducted by Dyflexis, with specific attention given to the platform and security sections. This assessment aimed to evaluate the preparedness of Dyflexis to migrate to AWS PaaS. The platform section emphasized the importance of investing in platform engineering capabilities and implementing best practices for designing and managing the platform architecture. The security section highlighted the need for data classification, data governance policies, and encryption techniques to protect sensitive information.

Regarding the migration steps, the report outlines the process of transitioning the infrastructure of different components of the Dyflexis application. This involves carefully migrating data, configurations, and dependencies from the existing IaaS environment to the PaaS platform.

The case study highlights the motivations for the migration, the assessment of AWS readiness, and the steps involved in transitioning the infrastructure. This report intends to serve as a valuable resource for both Dyflexis and other Independent Software Vendors (ISV's) considering a similar transition and showcases the benefits and considerations involved in adopting a higher abstraction level for enhanced scalability and customer responsiveness.

Method

The overall study follows a qualitative approach. Hence, methods used in this case study include data collection, procedure analyses and direct observations. Multiple sources of data were used to enhance the validity of the findings. Data collection also includes the review of existing documentation associated with the current IaaS infrastructure of the organisation, which consists out of the network architecture, storage configuration and computing resources.

Other data collection methods are semi-structured interviews with IT managers, Infrastructure administrator and Cloud experts. The interviews are designed to gather in-depth information about the company's goals, challenges with the present IaaS infrastructure and expectations from the PaaS transition. Relevant documents were collected to gain further insights in the transition process such as migration plans, assessment reports and roadmaps.

The collected data, including interview transcripts and document review findings, were analysed using thematic analysis. The analysis involves systematically organising and interpreting qualitative data, such as interview transcripts, focus group discussions, or written documents, to uncover underlying themes and gain a deeper understanding of the research topic (Castleberry and Nolen, 2018). Part of this analysis were identifying recurring themes, patterns, and insights related to the migration process, challenges, and outcomes.

The case study technique described above aims to offer a full knowledge of the IaaS to PaaS conversion process. By employing a combination of AWS assessments, interviews with experts and architects, and a comprehensive literature review, robust results can be obtained for an area that is relatively unknown, such as the transition from IaaS to PaaS.

Findings

This chapter presents the findings of the case study. The findings are organised based on the research questions and themes that emerged from the data analysis of the conducted interviews. Throughout this chapter quotes or citations from interviews and external experts are used to support the findings.

Business Domain

Dyflexis resides in the business domain application development, specifically the design of workforce management software solutions. These software solutions are intended to help organisations improve personnel scheduling, time registration, labour cost control, and other workforce management operations. Shift planning, time tracking, absence management, employee communication, labour forecasting, and reporting and analytics are common aspects of their program.

To fully utilise the advantages of the platform model and make informed choices, it is crucial to pinpoint and examine the precise drivers behind the shift to PaaS.

These drivers showcased in *Table 6* describe the key drivers that prompted the migration from IaaS to PaaS, the challenges that may be encountered during the transition, and the outcomes achieved.

DYFLEXIS			
Driver/motivation	Challenge/risk/obstacles	Process/Solution	(Expected) Result/Benefit
Ability to respond quickly to customer's demand	Possible high churn rate because of vendor dependency	Retarget lost customers: showcase the enhancements and benefits of the new platform. Make a strong argument for their return.	High user satisfaction Retained customer base Improved customer loyalty
Rapid business/application growth restricted by infrastructure	Competitive disadvantage Learning curve gap	External consultancy In-house training and certification	No unscheduled downtime Focus development on application design instead of infrastructure
Sync vs Async (events)	Extra infrastructure required. Event broker, puller DB transactions require additional code	Setup an event broker on the new platform	Scalable opportunities

Containerisation	Currently, within deployment application packing takes place within the code	Kubernetes and docker services:	Predictability, Redundancy, ease of use,
Event Driven Architecture (EDA)	Compatibility and integration considerations	Conduct thorough testing and proof-of-concept exercises	Event-driven service integration instead of custom integration solutions
Modularization; Splitting the monolith	Knowledge gap, cost insights, limiting complexity	Follow 3-step phase (Ch. 5)	Simplified Development and Deployment
Overprovisioning of resources	Limited customisation	Enable Elastic scaling Leverage resource monitoring tools and services	Usage based billing: pay only for the resources you consume instead of having fixed expenses Gaining insights into the application's capacity requirements

Table 6 Overview Key Drivers

Recognising the relationship between the drivers and their outcomes with the categories of scalability, cost optimisation, and efficient development and deployment is essential. It highlights the importance of considering these aspects from the beginning to ensure a well-planned and successful transition to PaaS.

For each category mentioned prior, the following explanations and examples are provided to illustrate its role and impact in the transition from IaaS to PaaS:

1. Scalability: With PaaS, organisations can leverage the scalability and flexibility of the platform to rapidly scale their application resources based on customer needs. For example, during peak periods of high demand for scheduling or time-tracking features, the PaaS platform can automatically allocate additional computing resources to ensure smooth and responsive user experiences.

2. Development and deployment: PaaS enables faster development and deployment of new features and functionalities, ISV's to quickly adapt to market trends and customer requirements. For instance, when expanding their software to support new industries or regions, the PaaS platform provides the necessary tools and infrastructure to accelerate the development and deployment cycles, which in

return enables rapid business growth.

3. Cost optimisation: PaaS provides the flexibility to dynamically allocate resources based on demand, avoiding the need for overprovisioning of infrastructure resources. In the context of WFM software, this means that the application can scale up or down automatically based on workload fluctuations. For example, during periods of low usage, the PaaS platform can reduce the allocated resources to optimise costs, while during high-demand periods, it can scale up resources to ensure smooth operation without the need for manual intervention. This examples also underlines the pay-as-you-go model of PaaS allowing the organisation to pay only for the resources they actually use.

Backend infrastructure

Describing the backend of the application before transitioning is crucial because it provides a clear understanding of the existing infrastructure, dependencies, and requirements. This also enables the identification of specific components that need to be migrated first, helps assess the compatibility of the backend with the target PaaS platform, and guides the development of a comprehensive migration plan.

The current state of the backend infrastructure is as described in *Figure 7*.

In an event-driven context, the backend process can be described as follows: When an end user requests the program, an event is triggered. This event is then stored in the database table, capturing the relevant information. Simultaneously, the program saves the status of the user's request to the MySQL database. This process is commonly referred to as the "transactional outbox" pattern in software architecture. This pattern ensures reliable and consistent communication through events between different services or components within a system (de Oliveira Rosa et al, 2020). In the domain of workforce management, this pattern can be applied to ensure the dependable and consistent handling of events associated with employee scheduling, time tracking, shift changes, and other workforce-related actions. The transactional outbox pattern provides a robust foundation for maintaining data integrity and facilitating seamless event-driven interactions within the workforce management system.

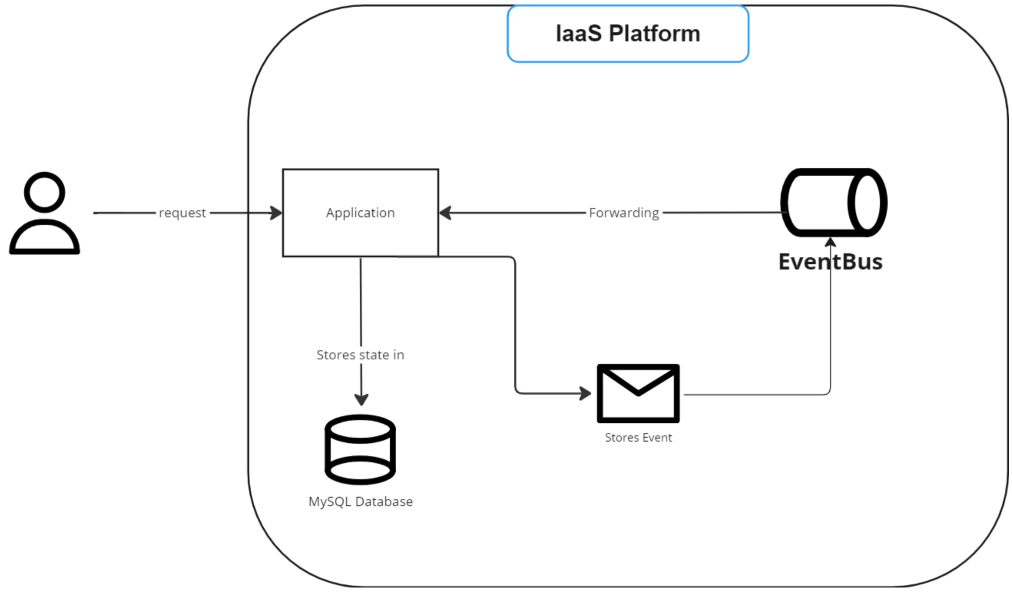


Figure 7 Current Backend infrastructure

An integrated service known as a message relay pulls the event from the tables and hands it over to the event broker. The program then subscribes to this event at the event broker and handles incoming events. At the current moment, there is a manual configuration involved in subscribing to and sending events from the event broker within the application. Instead of relying on manual configurations within the application, PaaS services provide more efficient solutions where you can enhance event subscription, automate event handling and coherently integrate different components of the application.

An example of a frequently used event broker service in PaaS is called AWS EventBridge. With AWS event bridge rules can be created that decide how events from different sources are routed and handled. Furthermore, automatic trigger actions in reaction to events, such as sending alerts via additional integrations such as Amazon SNS (Simple Notification Service), may be performed. *Figure 8* visualizes the desired flow of the backend within PaaS.

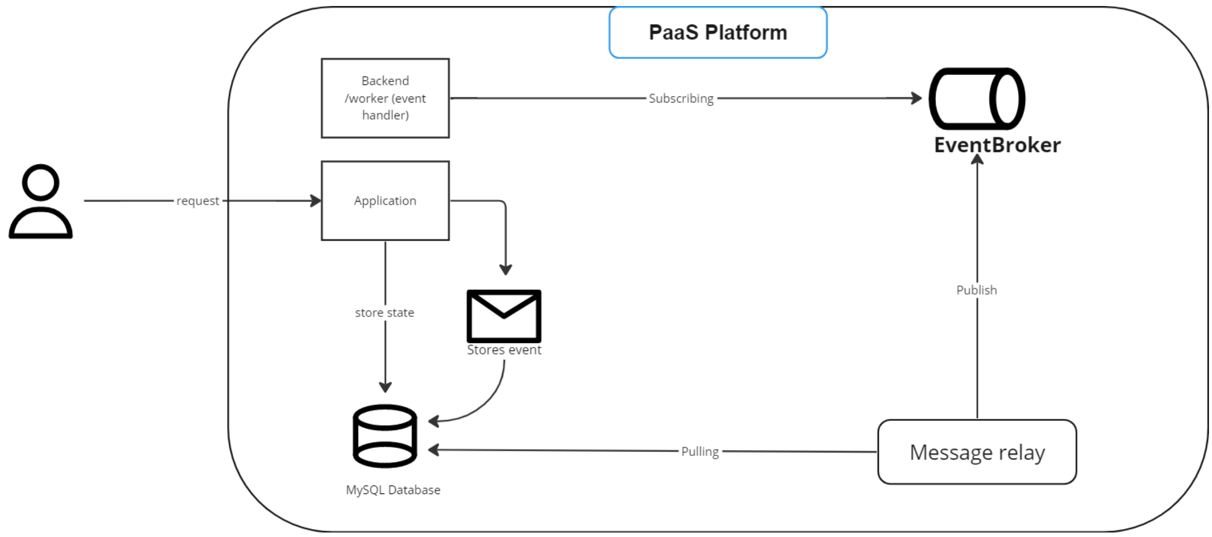


Figure 8 Desired Backend Infrastructure

The primary distinction between an event broker and an event bus is their amount of centralization and controversy.

An event broker is a centralized component that handles event routing, delivery, and administration, similar to a message broker. It usually includes extra features like event persistence, message queuing, and complex routing capabilities. Apache Kafka, RabbitMQ, and AWS EventBridge are examples of event brokers.

An event bus is a communication pattern or notion that specifies how events are shared between components. Message queues, publish-subscribe systems, and event-driven frameworks are some of the technologies that may be used to create it. An event bus can be built as an independent component inside a system or as part of an event broker.

AWS Cloud Transformation Maturity Model

As mentioned in the introduction, Dyflexis participated in an AWS cloud readiness assessment. This assessment was taken on May 23rd 2023. The representatives of Dyflexis in this assessment were Pascal Lubbers and Jamal Thijssen. The assessment explains and illustrates both the strong and weak sections of the company across the six AWS Cloud Adoption Framework (CAF) perspectives. The perspectives consist out of the sections Business, People, Governance, Platform, Security and Operations.

The AWS readiness assessment serves as a vital starting point for companies considering a migration to the public cloud. It helps them understand the benefits and potential risks associated with such a transition, enabling informed decision-making and planning (AWS, 2022).

As shown in Figure 3, Dyflexis received a relatively low score in the perspectives Business, People, Governance and platform. This suggests that these areas may require further attention and improvement in terms of the AWS cloud readiness assessment conducted for Dyflexis.

On the other hand, Dyflexis obtained a higher score in the perspectives of Security and Operations. This indicates that Dyflexis has demonstrated a relatively stronger readiness in these specific areas.

It is worth noting that while the Business, People, and Governance perspectives received a lower score, they should not be overlooked. These areas play a vital role in the overall cloud readiness and successful migration. However, due to time limitations and the scope of the assessment, the decision to focus on Security and Platform is made to prioritize immediate improvements and mitigate potential risks associated with these critical aspects of the cloud infrastructure.

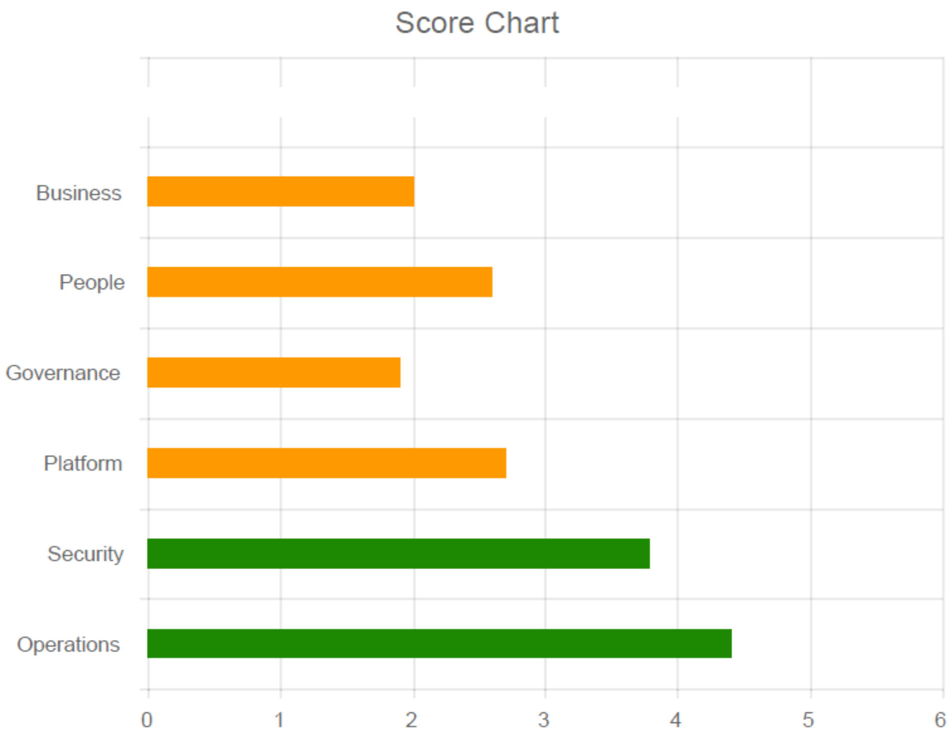


Figure 9 Score Chart AWS CAF

A low score in the business perspective indicates that Dyflexis has challenges related to aligning their business objectives with the AWS cloud platform. It suggests they may need to focus on understanding how AWS can support their specific business requirements, such as cost optimisation, scalability, or agility.

A low score in the people perspective suggests that the company may lack the necessary skills, knowledge, or resources to effectively utilise AWS. This could include gaps in expertise with AWS services, training needs for employees, or a lack of a dedicated team to manage and optimise AWS resources.

A low score in the governance perspective indicates that the company may have deficiencies in their AWS governance practices. It suggests they may need to establish or improve policies, processes, and controls for managing their AWS environment effectively. This could include areas such as compliance, security policies, or resource management.

Platform

In the AWS readiness framework, a score between 2 and 3 in the platform section indicates a moderate level of readiness in terms of the assessed criteria. It suggests that there are areas within the platform aspect that may require further attention, improvements, or optimisation.

The score between 2 and 3 signifies that while there might be some existing strengths and capabilities in the platform section, there is still room for enhancement. For example, Dyflexis can invest in platform engineering by optimizing their infrastructure to handle increased workloads and ensure high availability. This may involve implementing load balancing mechanisms, optimizing network configurations, and leveraging caching techniques to improve performance and scalability.

Another improvement area is structuring the overall deployment process. Dyflexis can implement CI/CD pipelines using tools like AWS CodePipeline or Jenkins. This enables them to automate the build, test, and deployment processes, ensuring that code changes are continuously integrated, tested, and deployed to production. This allows for faster feedback loops, reduces the risk of manual errors, and improves overall agility.

"The deployment of updates and even the applications strongly depend on having a good CI/CD pipeline" - J. Crumley

To fully understand the implications in the platform section, it is important to refer to the specific criteria and indicators used within the AWS readiness framework. These criteria further assess various aspects related to platform engineering, infrastructure scalability, performance optimisation, automation, and other relevant factors.

By analysing the detailed assessment results and recommendations associated with the platform section, organisation, such as Dyflexis can identify specific areas for improvement and take appropriate actions to enhance their platform's readiness for the cloud environment.

Security

In the AWS readiness framework, a score between 3 and 4 in the security section indicates a relatively high level of readiness in terms of the assessed security criteria. It suggests that Dyflexis has implemented robust security measures and practices, demonstrating a strong commitment to safeguarding their cloud environment.

A score between 3 and 4 indicates that the organisation has established effective security controls, policies, and procedures to protect their infrastructure, applications, and data within the AWS environment. It indicates a good understanding and implementation of security best practices, compliance requirements, and risk management strategies.

This also demonstrates that Dyflexis has taken significant steps to ensure the confidentiality, integrity, and availability of their resources in the cloud. It suggests that they have invested in security technologies, implemented appropriate access controls, encryption mechanisms, monitoring systems, and incident response processes.

While these results indicate a high level of readiness in the security section, it's important to note that here too may still be areas for improvement or specific recommendations provided within the assessment. For example, Dyflexis can strengthen access controls by implementing strict user authentication mechanisms, such as multi-factor authentication, to prevent unauthorized access to sensitive data. Additionally, they can enhance data encryption practices by adopting encryption at rest and encryption in transit techniques, ensuring that data remains protected both while stored and during transmission.

Furthermore, to enhance overall security posture, regular security assessments and vulnerability scanning should be conducted to identify and address any potential weaknesses in the infrastructure and applications. Dyflexis can also invest in security training and awareness programs to ensure that employees understand and follow best practices for data protection.

A score between 3 and 4 in the security section of the AWS readiness framework is a positive indication of the organisation's commitment to maintaining a secure cloud environment.

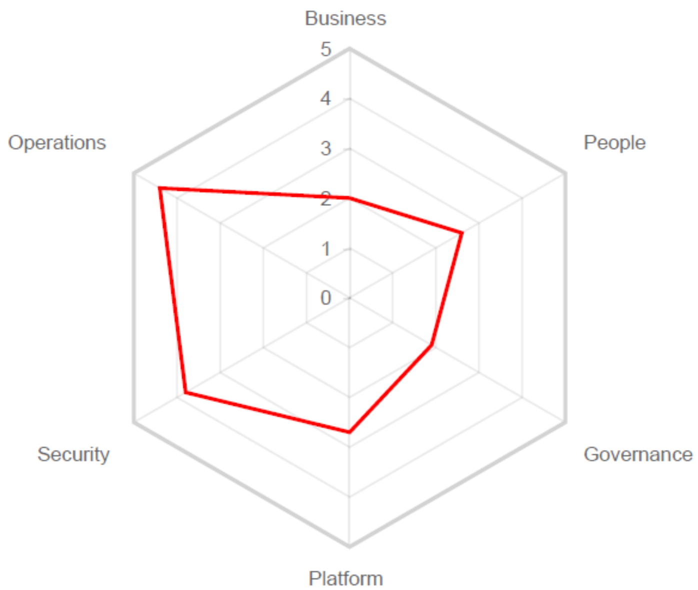


Figure 10 Radar Chart

Migration strategy

In the case of Dyflexis, there are two strategies available for migrating from one cloud provider to another:

1. The Lift & Shift,
2. The Strangle pattern

The first strategy involves lifting the existing infrastructure and applications from the current cloud provider and shifting them as-is to the new cloud environment. However, alongside the migration, the opportunity is taken to modernize and optimise the applications, making use of the features and capabilities offered by the new cloud platform. This strategy allows for a relatively straightforward migration process while also enhancing the application's performance, scalability, and efficiency. For these reasons this is the chosen approach by Dyflexis.

The second strategy, strangle pattern, is an alternative approach. In this strategy, instead of migrating all applications at once, a gradual transition is followed. This approach allows for a step-by-step migration, enabling the organisation to leverage the benefits of both the new and existing cloud providers in a multi-cloud setup.

In a hybrid or multi-cloud environment, organisations need to manage and maintain infrastructure and services across different cloud platforms (Alshammari et al., 2017). This involves additional operational and management overhead, as well as potential costs associated with data transfer, network connectivity, and licensing agreements with multiple providers. On the other hand, the lift and shift allows organisations to take advantage of the cost efficiencies and scalability offered by the new cloud platform without the complexities of managing a hybrid or multi-cloud environment.

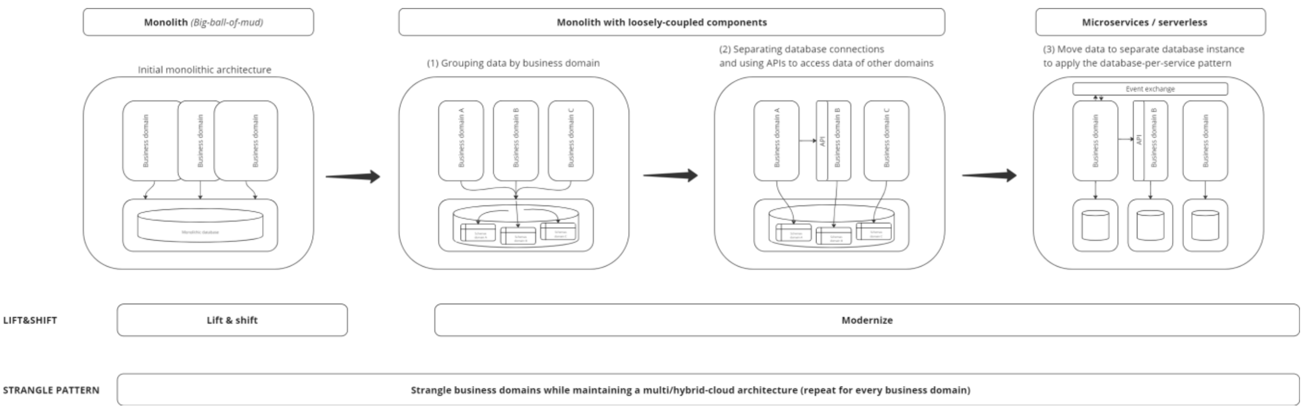


Figure 11 Migration approach

Outlining the high-level steps involved in moving from a monolithic architecture to a cloud-native environment is necessary before implementing the migration plan. These actions will act as a road map to direct the migration process and guarantee an efficient transition.

High-level Steps for Transitioning from Monolith to Cloud-Native Architecture:

1. Define Long-term Business Goals: Determine the organisation's long-term objectives and strategic vision for the cloud migration.
2. Establish OKRs: Define Objectives and Key Results (OKRs) based on the identified business goals to guide the migration process.
3. Prioritize Investments: Use OKRs to prioritize where to invest resources for the migration, focusing on high-value areas.
4. Visualize Value Chain with Wardley Maps: Utilise Wardley maps to visualize the value chain of offerings from a user perspective, aligning it with user needs.
5. Create Global Map of Business Domain: Employ techniques like Event Storming and Bounded Context Canvases to create a comprehensive global map of the business domain.
6. Assess Subdomains: Compare subdomains with the target architecture and determine whether they can be replatformed, rearchitected, or rebuilt.
7. Move Features to Cloud-Native Infrastructure: Based on the assessment, move features within their well-defined boundaries to the new cloud-native infrastructure.
8. Map Implementation against Target Architecture: Assess the implementation of each context against the target architecture and current coding standards.
9. Decide on Feature Treatment: Based on the assessment, decide whether to rebuild, rearchitect, or replatform the features within their respective contexts.
10. Evaluate Monolith's Role: As high-value features are migrated, evaluate the monolith's role, considering whether it should be retained as a support service or retired based on cost-benefit analysis.
11. Completion of Migration: Over time, the cloud-native architecture will support all core features, reducing dependency on the monolith. The migration is considered complete when the monolith primarily contains supporting features.

By following these high-level steps, the organisation can systematically transition from a monolithic architecture to a cloud-native infrastructure, effectively leveraging the benefits of cloud computing, enhancing scalability, and optimizing cost and resource allocation.

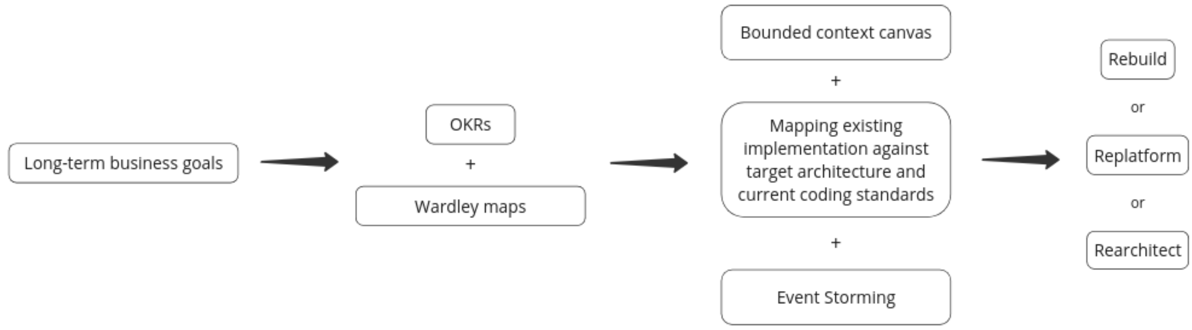


Figure 12 High Level steps

This cycle provides a systematic approach to migration projects. It enables Dyflexis to set priorities and focus on various system components, enabling a planned and calculated approach to (future) migration.

As mentioned before, when it comes to modernizing the architecture or modern application development, the second strategy of strangle pattern often considered more suitable. This approach recognises the need for a gradual transformation by building new components or services alongside the existing ones, eventually replacing or "strangling" the functionality of the legacy system.

Modern application development refers to the approach of building and deploying applications using contemporary methodologies, architectures, and technologies. It aims to address the challenges posed by complex and rapidly evolving business requirements, as well as advancements in cloud computing and software development practices.

In the context of application architecture, the term "monolith" refers to a traditional approach where the entire application is built as a single, tightly coupled unit. In a monolithic architecture, all components and functionalities are tightly integrated, making it challenging to scale or modify individual parts independently. However, when transitioning to a more scalable and flexible environment like PaaS, it is beneficial to move away from the monolithic approach and adopt a loosely coupled component-based architecture.

Loosely coupled components refer to individual modules or services within an application that are designed to be independent and have minimal dependencies on each other. These components can be developed, deployed, and managed separately, enabling better scalability, maintainability, and flexibility.

By breaking down the monolithic application into loosely coupled components, each component can be individually scaled based on demand, allowing for more efficient resource uitilisation. Additionally, with loosely coupled components, it becomes easier to introduce new functionalities, make updates, or fix issues without impacting the entire application. This modular approach promotes flexibility, agility, and scalability, which aligns well with the capabilities provided by PaaS platforms.

The strangler pattern can be applied in a tranisition process using three main phases: the monolithic phase, containerisation phase, and maintaining phase. This means identifying boundaries and splitting off subdomains piece-by-piece. While doing it's possible to have a hybrid architecture that mixes a monolithic architecture and service-oriented architecture (microservices). Once all domains are strangled and have well-defined boundaries encapsulated in services, the monolithic architecture can be removed (Figure 6).

Monolithic phase

1. Identify Service Boundaries: Analyse the existing monolithic application to identify potential microservices and define their boundaries based on business capabilities, domain-driven design principles, and technical dependencies.
2. Define APIs and Contracts: Design and document clear interfaces and contracts for each microservice, enabling loose coupling and independent development.
3. Data Management Strategy: Analyse data requirements and determine an appropriate data management strategy, such as a shared database or separate databases per service, to ensure data consistency and availability.

Containerisation provides a lightweight and efficient way to package applications and their dependencies, enabling consistent and reliable deployment across different environments. Containers encapsulate the application and all its dependencies, ensuring that it runs consistently regardless of the underlying infrastructure (Khan, 2017).

Container orchestration platforms should support declarative configuration and infrastructure-as-code principles. This allows developers to define the desired state of the application infrastructure using code, simplifying deployment and ensuring consistency across environments.

Additionally, experts highlight the importance of efficient resource uitilisation and cost optimisation. Container orchestration platforms should have features to efficiently manage and allocate resources, such as scaling down idle services or optimizing resource allocation based on workload patterns.

Containerisation phase

1. Containerisation: Package each microservice as a containerised application using technologies like Docker, enabling easy deployment, scalability, and isolation.
2. Implement Infrastructure as Code: Utilise infrastructure-as-code tools (e.g., AWS CloudFormation) to define the required infrastructure resources, such as load balancers, autoscaling groups, and network configurations, making deployments repeatable and automated.
3. Service Orchestration: Implement service discovery mechanisms (e.g., AWS Elastic Load Balancer, AWS Service Discovery) to enable dynamic service registration, routing, and load balancing.
4. Event-Driven Architecture: Leverage event-driven patterns and messaging systems (e.g., Amazon SNS, Amazon SQS, or Apache Kafka) to enable asynchronous communication and decoupling between services.
5. CI/CD Pipeline: Establish a robust CI/CD pipeline to automate the build, test, and deployment processes for each microservice, ensuring frequent and reliable releases.

Another key characteristic of containerisation is fault tolerance and self-healing capabilities. The orchestration platform should have mechanisms in place to detect and automatically recover from failures, such as restarting failed containers or redistributing workloads to healthy instances.

Maintaining phase

1. Monitoring and Observability: Implement monitoring and observability tools (e.g., AWS CloudWatch). Observing the monolith with loosely coupled components is no different than

monitoring the initial monolith. Errors, exceptions and useful information is logged, aggregated and displayed in a readable way using software like Prometheus, Grafana or Datadog.
2. Training and Upskilling: Invest in training and upskilling your IT team to familiarize them with the new PaaS environment. This includes understanding the platform's concepts, features, and best practices for application deployment and management.

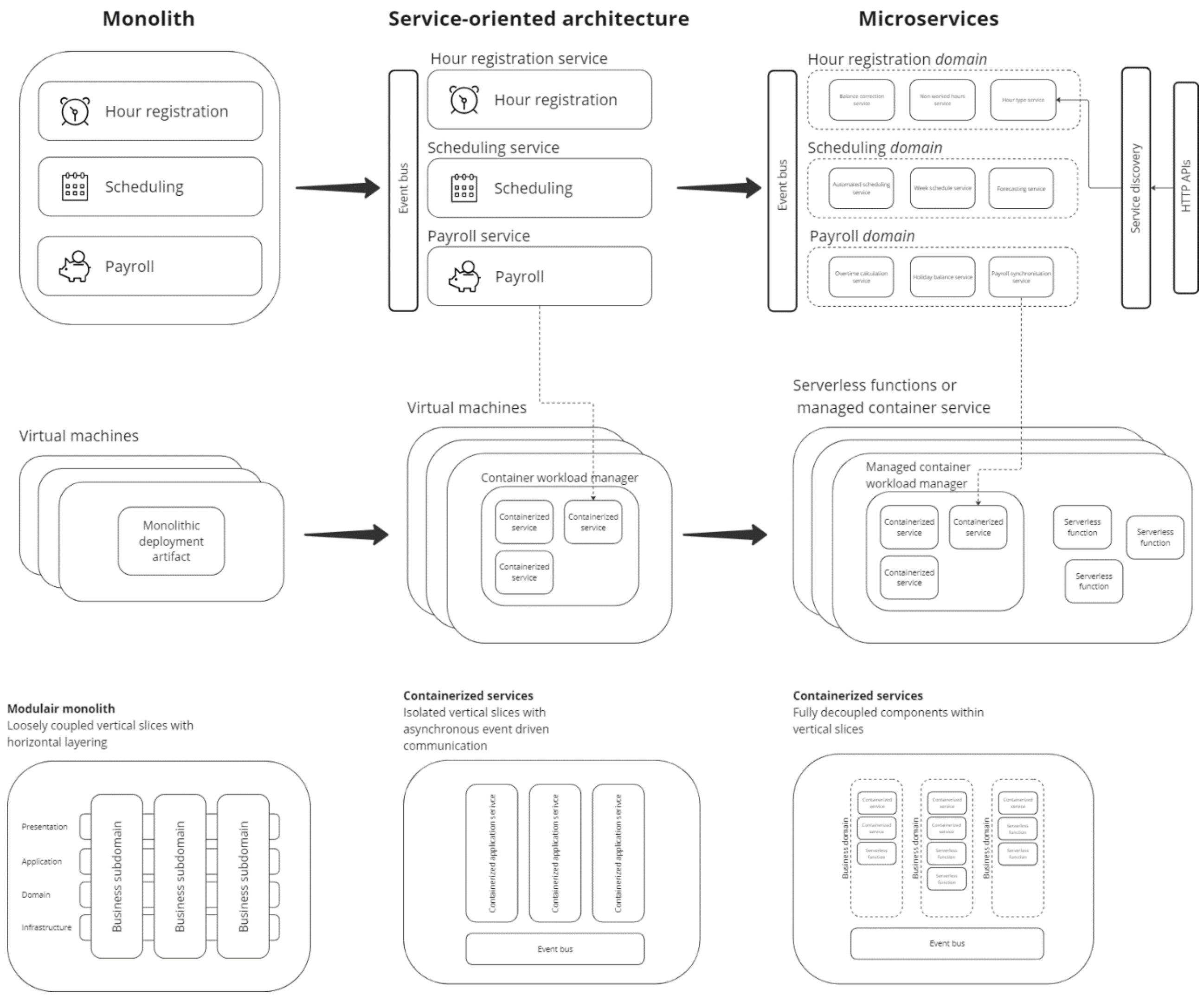


Figure 13 Monolith to Microservices

Data is a critical aspect of any application, and migrating it seamlessly to the PaaS platform is crucial. The following steps involves planning and executing the data migration strategy, ensuring data integrity, and minimizing downtime during the transition.

1. Analyse existing database and group data (tables, views, stored procedures, foreign keys, triggers, etc.) by business domain
2. Use separate database connection on application level to access the data of the separated business domains. Business domains that need data that is now owned by another business domain must use the service's API to access the data

3. To migrate to database-per-service architecture: move data to separate database instance

Role of CTO in the migration process

"It really is a cost benefit analysis on risk, business value you're delivering. And really where you want to put your focus on...." - L. Van Donkersgoed

The citation emphasizes the importance of conducting a cost benefit analysis, evaluating risks, and determining the business value of cloud solutions. The CTO plays a significant role in assessing these factors and making strategic decisions about where to invest resources and focus efforts.

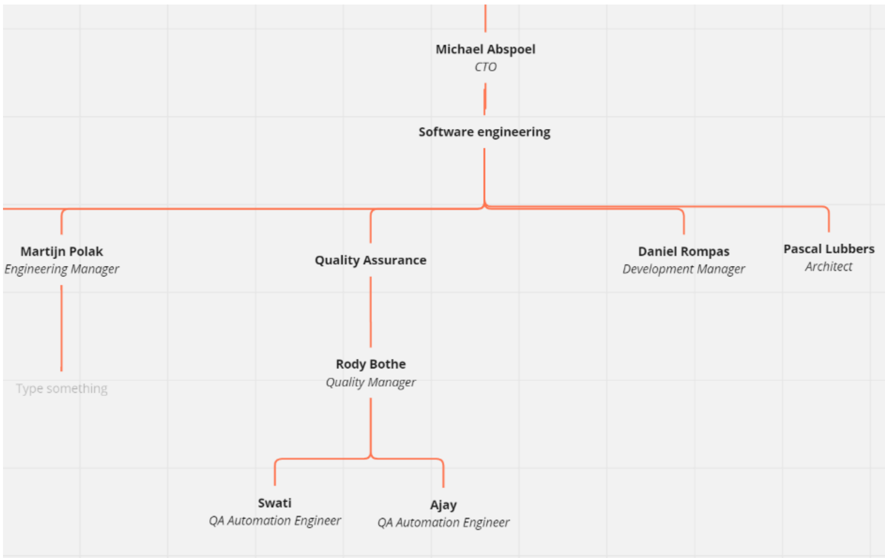
The CTO, as a senior technology executive, is responsible for overseeing the organisation's technology strategy and ensuring alignment with business objectives (Adler and Ferdows, 2001). Their role extends to evaluating and implementing technological changes, including migrations.

The CTO is in charge of creating the overall plan and roadmap for the transition from IaaS to PaaS. He reviews the business needs, weighs the advantages and drawbacks of PaaS adoption, and establishes the migration's strategic goals and objectives. The CTO is seen as Head of the development department, which is build up out of multiple teams (Figure X)

The architect and other development teams offer advice on how to successfully leverage PaaS services, optimise resource consumption, ensure scalability, and adopt best practices for security and performance. Accordingly, the CTO reviews and analyses the new PaaS-based infrastructure and application architecture on migration-related risks such as data security and privacy, flexibility, performance and costs.

A in-depth cost comparison is performed between the current IaaS costs and the projected costs of PaaS adoption. He considers factors such as scalability, resource uitilisation, and efficiency gains offered by PaaS. This analysis helps determine if PaaS will result in cost savings or improved cost efficiency in the long run.

Other expenses associated with the migration are one-time costs, including xpenses such as rearchitecting applications, data migration, training, and any required consulting or professional services. These costs need to be factored into the overall analysis to determine the total cost of migration.


Michael Abspoel
CTO
Software engineering
Martijn Polak
Engineering Manager
Quality Assurance
Daniel Rompas
Development Manager
Pascal Lubbers
Architect
Type something
Rody Bothe
Quality Manager
Swati
QA Automation Engineer
Ajay
QA Automation Engineer

Analysis

In this chapter, the report provides recommendations specifically focused on the AWS assessment, with a particular emphasis on the platform and security sections. These recommendations aim to guide Dyflexis in improving their infrastructure based on the findings from the assessment. Figure 7 describes a heatmap of the sections. The green sections indicate a high level of cloud adoption readiness, whereas the yellow sections imply that additional preparation is recommended.

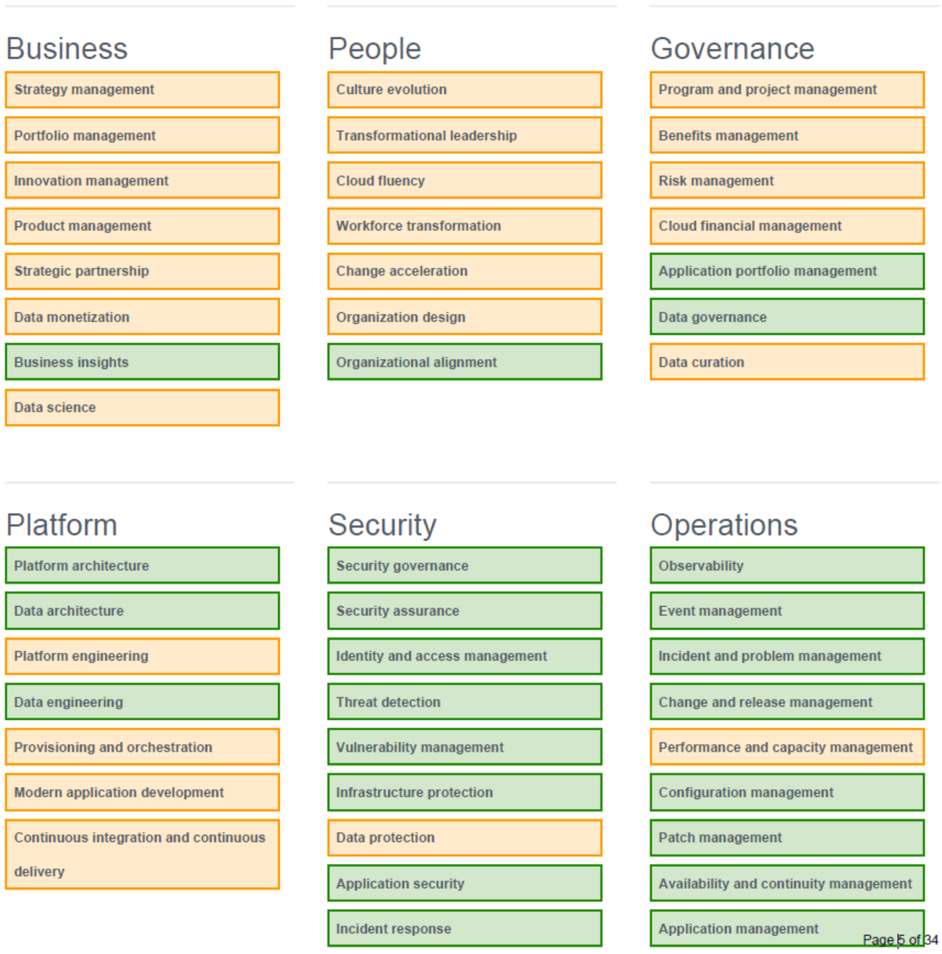


Figure 14 Heatmap CAF Sections

As mentioned before given time constraints and the scope of the study, the emphasis will be placed on the sections **platform, and security** of the AWS Cloud Readiness Assessment, while the aspects related to business people, operations and governance will be excluded.

Alshdadi et al. (2020) discovered through platform evaluation that several legacy systems employed by the institutions needed significant changes or adjustments to interface seamlessly with the cloud environment. A smooth transfer to the cloud was made possible by the institutions' decision to devote resources for the required system improvements and architectural modifications.

The assessment of the platform section helps find any holes or restrictions that could prevent a seamless shift to the cloud. Organisations may decide on essential upgrades, system integrations, and future scalability difficulties by knowing the strengths and limitations of the present platform.

The platform part also examines the potential for interoperability and integration between the organisation's current systems and the cloud environment. This assessment helps determine the feasibility of migrating critical applications and data to the cloud without disrupting ongoing operations. It also assists in outlining the necessary steps to ensure a seamless integration process.

Dyflexis can allocate resources and expertise towards developing strong platform engineering capabilities. This includes building a team of skilled engineers dedicated to maintaining and improving the infrastructure supporting their software. By investing in platform engineering, Dyflexis can ensure a robust and scalable infrastructure that can handle increasing demands and evolving customer needs.

Furthermore, Dyflexis can adopt industry best practices when designing, building, and managing their platform architecture. This involves following architectural principles, such as modularity, scalability, and fault tolerance. By adhering to these practices, Dyflexis can create a platform that is flexible, reliable, and capable of supporting the software's performance and growth.

On the other hand, Chen et al. (2022) found that assessing security measures was of foremost importance to address the unique security challenges faced by ISVs in the cloud environment. One of the remarkable findings in this article were that the ISVs had not implemented robust access controls for their cloud-based software. This lack of proper access controls posed a significant security risk, potentially allowing unauthorized access to sensitive customer data and intellectual property.

As a remedy the organisation was compelled to implement strict access control measures, including role-based access control (RBAC) and multi-factor authentication, to protect customer data and prevent unauthorized access.

According to the article by Chen, there are several key areas in which Dyflexis can make improvements related to data classification, data governance, and encryption techniques.

Data Classification: Dyflexis can conduct a thorough data classification process, identifying and categorizing data based on its sensitivity and criticality. By classifying data, Dyflexis can determine the appropriate level of security controls needed to protect it. This enables the implementation of targeted security measures and ensures that sensitive data receives the highest level of protection.

Data Governance: Dyflexis can establish robust data governance policies and procedures to govern the handling, storage, and access to data. This includes defining clear roles and responsibilities, implementing access controls and permissions, and establishing data retention and deletion policies. By enforcing strong data governance practices, Dyflexis can maintain data integrity, privacy, and compliance with relevant regulations.

Encryption Techniques: Dyflexis can implement encryption techniques to protect data from unauthorized access. This includes encryption at rest, which ensures that data stored in databases or on storage devices remains encrypted when not in use. Additionally, encryption in transit can be implemented to secure data as it moves between systems or networks. By leveraging encryption, Dyflexis can safeguard sensitive information and prevent unauthorized disclosure or interception.

Based on the literature, AWS evaluation, and conversations with cloud specialists, the following suggestions are summarized and suggested for Dyflexis to place itself in a more mature stage of cloud readiness:

Platform section

- Invest in platform engineering capabilities to ensure a robust and scalable infrastructure for the software.
- Implement best practices for designing, building, and managing the platform architecture.
- Implement infrastructure-as-code practices to enable automated provisioning and configuration of resources.
- Utilise infrastructure orchestration tools, such as AWS CloudFormation or Terraform, to manage infrastructure deployments consistently.
- Adopt modern application development methodologies, such as microservices or serverless architectures, to enhance scalability and flexibility.
- Leverage containerisation technologies, such as Docker, to enable portability and easier deployment
- Establish a CI/CD pipeline to automate the build, test, and deployment processes
- Implement version control systems, such as Git, to manage source code and track changes.

Security section

- Classify data based on its sensitivity and criticality to determine appropriate security controls.
- Implement data governance policies and procedures to ensure proper handling, storage, and access to data.
- implement encryption techniques, such as encryption at rest and encryption in transit, to protect data from unauthorized access.

Noticeable is the length of recommendations per section. The recommendations are made merely on the sections where Dyflexis has scored below average.

"Make sure you have a clear separation of responsibilities. Storage vs Compute vs Network."
L. Van Donkersgoed

A clear separation of responsibilities within the Dyflexis infrastructure has been established (table 2). This separation acknowledges the distinct roles and functions of different components in the infrastructure, namely storage, compute, and network.

By separating these responsibilities, Dyflexis ensures that each component can be managed independently, optimizing resource allocation and performance. Storage is dedicated to handling persistent data storage and retrieval, compute focuses on running applications and processing tasks, and network facilitates communication and data transfer between different components.

This separation of responsibilities brings several benefits. Firstly, it allows for scalability and flexibility, as each component can be scaled independently based on its specific requirements. For example, if the application requires additional compute resources, scaling can be focused solely on the compute component without affecting storage or network. This granular scalability ensures efficient resource uitilisation and cost optimisation.

Secondly, a clear separation of responsibilities enhances fault tolerance and resiliency. By isolating storage, compute, and network functionalities, failures or disruptions in one component are less likely to impact the others. This improves system reliability and minimizes the risk of single points of failure.

Lastly, the separation of responsibilities enables efficient management and maintenance. Different teams or roles can be assigned to handle specific components, ensuring specialized expertise and streamlined operations. It also simplifies troubleshooting and debugging processes, as issues can be isolated to a particular component.

	Compute	**Storage**	**Network**
IaaS	Overprovisioning, controlling and managing virtual servers	Control over the configuration of storage types: block, object or file storage. Responsibility of the storage infrastructure lies with the customer	TCP/IP managed by external partner, who is responsible for the configuration of virtual networks, subnets, firewalls and load balancers
PaaS	Automatically scaling EC2's based on demand	The responsibility of backups, database operating systems lies with the provider	Pre-configured network settings and security features offered by the platform

Table 7 Distinction IaaS and PaaS

Conclusion

In conclusion, the comparison between IaaS and PaaS highlights the advantages of PaaS in abstracting away underlying infrastructure administration and empowering developers to focus on their applications. While adopting a PaaS model may necessitate upskilling or retraining of IT employees, the potential benefits of enhanced modularity, flexibility, and scalability are significant. The adoption of loosely coupled systems allows for seamless integration with external services and systems, fostering interoperability and leveraging the diverse ecosystem of cloud services and technologies.

However, it is essential to strike the right balance of loose coupling to avoid hindering system performance and maintainability. A careful assessment of system requirements and dependencies is crucial in establishing effective loose coupling. Fortunately, organisations have access to various tools, frameworks, and assessments to guide them towards their current state of cloud readiness and enable the adoption of cloud-native practices.

In this cloud-native era, embracing a microservices architecture aligns well with cloud principles, offering auto-scaling, load balancing, service discovery, and enabling continuous integration and continuous deployment (CI/CD). By leveraging these practices, organisations can streamline development and deployment processes, ensuring they are well-equipped to harness the full potential of cloud computing and drive innovation in the ever-evolving digital landscape.

References

de Oliveira Rosa, T., Daniel, J. F. L., Guerra, E. M., & Goldman, A. (2020, July). A method for architectural trade-off analysis based on patterns: evaluating microservices structural attributes. In *Proceedings of the European Conference on Pattern Languages of Programs 2020* (pp. 1-8). https://dl.acm.org/doi/abs/10.1145/3424771.3424809

Adler, P. S., & Ferdows, K. (1990). The chief technology officer. *California Management Review*, *32*(3), 55-62. https://journals.sagepub.com/doi/abs/10.2307/41166617?journalCode=cmra

Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds?. Currents in pharmacy teaching and learning, 10(6), 807-815.,

ISO 690, https://www.sciencedirect.com/science/article/abs/pii/S1877129717300606

M. M. Alshammari, A. A. Alwan, A. Nordin and I. F. Al-Shaikhli, "Disaster recovery in single-cloud and multi-cloud environments: Issues and challenges," *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Salmabad, Bahrain, 2017, pp. 1-7, doi: 10.1109/ICETAS.2017.8277868. https://ieeexplore.ieee.org/abstract/document/8277868

F. Brito e Abreu and M. Goulao, "Coupling and cohesion as modularization drivers: are we being over-persuaded?," Proceedings Fifth European Conference on Software Maintenance and Reengineering, Lisbon, Portugal, 2001, pp. 47-57, doi: 10.1109/CSMR.2001.914968. https://ieeexplore.ieee.org/abstract/document/914968

Khan, A. (2017). Key characteristics of a container orchestration platform to enable a modern application. IEEE cloud Computing, 4(5), 42-48. https://ieeexplore.ieee.org/abstract/document/8125559

Alshdadi, A. A., AlGhamdi, R., Alassafi, M. O., Alfakeeh, A. S., & Alsulami, M. H. (2020). A validation of a cloud migration readiness assessment instrument: case studies. SN Applied Sciences, 2, 1-12. https://link.springer.com/article/10.1007/s42452-020-3162-9