Universiteit
Leiden
The Netherlands

Integrating Multi-Party Computation in Data Warehouses for Secure Healthcare Analytics: An Architectural and Performance Study

J.H. Meijer (s4000757)

Supervisors:
Prof. dr. M.R. Spruit & Drs. E. Roorda

Bachelor Thesis– Informatica & Economie

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

July 17, 2026

Abstract

This research sets out to investigate how Multi-Party Computation (MPC) can be integrated into a data warehouse system to enable secure, privacy-preserving analyses. This research is being carried out in collaboration with RIGA, a foundation dedicated to improving care, health and well-being in the WSD region (Westland, Schieland and Delfland) through data-driven interventions. A literature review is first conducted, followed by interviews with professionals from the RIGA and Linksight. Through architectural modelling and performance testing, this study demonstrates that such a system can be viable, if it accounts well for performance issues of the technology. By proactively computing computationally intensive queries during off-peak hours, constructing pre-calculated data cubes and serving cached results, the system maintains practical utility for data analysts without compromising security. Performance evaluations of the Linksight system confirmed the heavy computational overhead and poor scalability inherent to MPC technology.

Acknowledgements

Before you lies the bachelor thesis "Integrating Multi-Party Computation in Data Warehouses for Secure Healthcare Analytics: An Architectural and Performance Study". This thesis was written as part of the study Computer Science & Economics at Leiden University. The research was conducted from January to July 2026 in collaboration with RIGA.

I would like to express my sincere gratitude to everyone who supported me throughout this journey. First and foremost, I want to thank my supervisors, Marco Spruit and Els Roorda for their guidance and support. I would also like to extend my thanks to Michelle van Delden and Jacinta van de Grint for overseeing my time within RIGA and providing me with the resources and feedback necessary to complete this thesis. Furthermore, I am grateful to the professionals from both RIGA and Linksight who generously gave their time to be interviewed and share their thoughts.

Finally, I want to thank my family, friends, partner and pets for their patience, encouragement, and support during the writing process.

Contents

1 Introduction

At a time when the healthcare sector is increasingly dependent on data sharing, it is vital that these exchanges are secure and protect privacy. Traditional methods often fall short in this regard, as data collaborations require data to be centrally available at an individual level. Such a vulnerability makes the system fragile and susceptible to the growing threat of cyberattacks and data breaches. Multi-Party Computation (MPC) offers a solution to this: a technique that enables organisations to perform joint analyses at an individual level without sharing their underlying data with one another.

This research is being carried out in collaboration with RIGA, a foundation dedicated to improving care, health and well-being in the WSD region (Westland, Schieland and Delfland) through data-driven interventions. Within this organisation, work is underway on the Joint Regional Information Platform (GRIP), a data workshop designed to provide secure access to up-to-date information on healthcare usage in the region. Currently, such data is limited and often only available with a delay via national sources such as Statistics Netherlands (CBS), which makes it difficult to respond in a timely manner to developments in healthcare demand.

GRIP's current infrastructure consists of an MPC implementation from the software provider Linksight, which is connected to an Azure Lakehouse structure via custom Python scripts. In this setup, data is staged and partially transformed through these scripts. However, achieving a fully automated and consistent transformation of this raw data into usable analytical data cubes remains a significant challenge; it is currently a manual, labour-intensive process that is confusing for data analysts and poorly scalable.

However, little is known about the combination of MPC and data warehousing, both in the literature and in practice. This thesis therefore focuses on the question: How can Multi-Party Computation be integrated into a data warehouse system to enable secure, privacy-preserving analyses? To this end, a literature review is first conducted, followed by interviews with professionals from RIGA and Linksight. Based on this, an architectural design will be drawn up, which aims to combine all requirements as much as possible with the current implementation. Finally, the existing MPC infrastructure will be tested for performance and scalability.

2 Background

2.1 Data warehouses

2.1.1 Historical development

In the early 1970s, the first calls were made for a system to support business decision-making. (Sprague Jr & Watson, 1996) The reasoning was that if a separate database containing a wealth of information about a company could be created, it could then be used for analysis and data-driven decision-making. This concept was later developed by Bill Inmon, whose research into this concept in 1992 led to the book 'Building the Data Warehouse' (W. Inmon, 1996). The so-called 'father of the data warehouse' (W. H. Inmon, Strauss, & Neushloss, 2010) described that a data warehouse had to meet four characteristics: subject-oriented, integrated, time-variant and non-volatile. He also argued that such an analytical system must remain separate from existing databases, as to ensure data processes would not collide with each other.

Around four years after the first publication of Inmon's book, Ralph Kimball published his own take on the data warehouse. In contrast to a 'top-down' approach, his book 'The Data Warehouse Toolkit' (Kimball, 1996) describes a 'bottom-up' method. The differences between those is expanded upon in section 2.1.2. Central to Kimball's approach was the concept of thinking in terms of so-called 'data cubes' and dimensions. A data cube allows a measure, such as a count of consultations, to be aggregated along several dimensions at once, such as time, region, and care type, and can be sliced or rolled up along any of them. The ideas implemented were initially conceived in the 1960s during a collaboration between General Mills (in Minneapolis) and Dartmouth University (Power, 2007).

In 1993, the term Online Analytics Processing (or OLAP for short) was coined by E.F. Codd (Codd, 1993). It describes the difference between systems designed to handle real-time data (OLTP or Online Transaction Processing) and those that are build purely for analytic purposes. OLAP systems usually store information in data cubes (also called OLAP cubes) and contain optimizations for better facilitating complex read queries, like storing data by column instead of by row.

Data warehouse architecture continued to evolve after Inmon and Kimball. Inmon himself, for example, introduced DW 2.0 (Data Warehouse 2.0) in 2008, which placed an emphasis on integrating the data lifecycle (W. Inmon, Strauss, & Neushloss, 2008b) and supporting unstructured data (W. Inmon, Strauss, & Neushloss, 2008a). As an alternative to the aforementioned approaches, the DataVault (later followed by a 2.0 version) was devised by Dan Linstedt (W. Inmon, Linstedt, & Levins, 2019) (Linstedt & Olschimke, 2016). In this approach, data was split into hubs (containing static information such as IDs), satellites (containing changing data) and links between the two. Today, a number of different data warehouse structures have emerged from this body of thought, such as the Hub-and-Spoke Data Mart Architecture, the Enterprise Warehouse with Operational Data Store, and the Distributed Data Warehouse Architecture (Sen & Sinha, 2005).

Although there is no global measurement of this, studies show that in some markets there is a strong preference for the 'bottom-up' approach (Hwang, Ku, Yen, & Cheng, 2004). The market size for data warehouses is around 30 billion dollars (Wadhwani, 2026), with an expected growth to over 60 billion dollars by 2032 (Singh, 2025). As the digital transition continues, data warehouses are becoming an integral part of the healthcare system. Despite the growing use of clinical data warehouses, there remains a lack of comprehensive understanding regarding their specific operational benefits and limitations (Lyu, Craig, O'Reilly, & Taniar, 2025).

2.1.2 Top-down vs bottom-up structures

Inmon and Kimball are best remembered for proposing two opposing strategies for how such a system should be build: top-down and bottom-up.

Inmon argued that a top-down approach, where datasets are first combined and then split up into smaller branches, would work best. First, a central, organisation-wide data warehouse containing all the information should first be built. Data in this data warehouse had to be stored in a normalised form, specifically in 3NF; this involves breaking the information down into smaller tables that are linked together via keys. This method of storage reduces redundant storage of repeated information and improves data integrity. This created a 'Single Source of Truth', from which smaller offshoots known as 'datamarts' could be created. These mini-data warehouses specialise in information about specific sectors (W. Inmon, 1996). The strength of this approach lies in enterprise-wide consistency: because every datamart is sourced from the same core, conflicts between different two out of sync datamarts is avoided. The main disadvantage is that the central warehouse must largely be designed and populated before any individual use case can be served, which tends to make top-down projects slower to deliver initial value and harder to adapt iteratively. This generally translates to high set-up and maintenance costs (Gath, 2024).

Kimball's bottom-up approach inverts this sequence. Subject- or department-specific data marts are built first, each modelled dimensionally around facts and data cubes and organised around shared dimensions. These individual data marts are then combined into a central data warehouse. Because each data mart can be delivered independently, this approach lends itself to incremental, use-case-driven development: value can be delivered for one analytical question without first having to remodel the underlying organization of the data (Kimball, 1996). Its drawback is the lack of strict governance of shared dimensions across marts, which risks bottom-up warehouses to drift into inconsistent definitions over time (Gath, 2024).

2.1.3 ETL vs. ELT

Populating a data warehouse from one or more source systems requires moving data through three conceptual stages: extracting it from its source, transforming it into a usable form, and loading it into its destination. The order in which these stages occur gives rise to two distinct strategies: ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) (Seenivasan, 2022).

In the traditional ETL approach, data is transformed before it is loaded into the warehouse. After extraction from the source system, data passes through an intermediate staging area, where it is cleaned and restructured. When loading the information into the data warehouse, it is validated against a target schema. This keeps the warehouse itself relatively lightweight and consistent, since transformations are handled by an external system and data is standardized upfront. Depending on the implementation, data formats are commonly provided by a Master Data or Reference Data system (Loshin, 2010).

With the rise of cloud computing, so has the popularity of moving data using ELT (Berisha, Mëziu, & Shabani, 2022). It shifts the burden of transforming data to the data warehouse, ensuring that the source systems can stay small. Since performance scaling is cheap and easy in cloud environments, data warehouses can scale up their compute if needed to handle large incoming datasets. If the raw data is also stored separately, new transformations can be done on the data without having to restart the data pipeline. This approach can cause issues when dealing with

sensitive information or storage limits, as it requires the data warehouse to read, process and (temporarily) store raw information.

2.1.4 Performance optimization

Because complex analytical queries are computationally expensive, data warehouses utilize internal mechanisms to optimize performance and prevent system bottlenecks.

To avoid executing heavy queries repeatedly, systems serve previously calculated results from a cache. Static caching relies on a Time-To-Live (TTL) constraint, securely storing and serving results for a predetermined duration before they expire. Alternatively, materialized views monitors the underlying source data for changes, automatically invalidating the cache only when the specific dependent datasets are updated (Hassan et al., 2022).

The Workload Manager acts as an orchestrator, dynamically queuing and allocating compute resources (CPU/memory) based on query priority and expected size. Job schedulers automate routine tasks, such as triggering ETL pipelines or pre-computing massive aggregations during off-peak hours to conserve daytime computing power (Gupta, Mehta, Wang, & Dayal, 2009).

2.2 Multi-Party Computation

Another key concept for this research is (Secure) Multi-Party Computing (MPC). This refers to cryptographic systems that enable multiple parties to perform a computation jointly without revealing individual inputs. Data is first encrypted on a row-level basis and entered into the network. A protocol takes care of joins, filters and any other calculations on the data in the network using advanced encryption algorithms. Only the final, aggregated outcome of these computations is then decrypted and revealed to the user. This ensures that the source data remains confidential to both external and internal parties (Zhao et al., 2019). Due to this property, MPC is widely used in privacy-sensitive environments; for example, the Dutch government recommends the technology to comply with GDPR legislation (Bennink & Drukarch, 2023).

The concept originated in the 1980s from a thought experiment on 'Mental Poker' (Shamir, Rivest, & Adleman, 1981), but was formally defined in Yao's Millionaire's Problem of 1982 (A. C. Yao, 1982). This problem describes two millionaires who wish to determine who is the richest, without revealing their exact wealth. Yao proved at the time that a solution existed, although more efficient methods have since been developed (Y. Yao, Zhang, Song, Zhang, & Wang, 2023).

In 1986, Yao formulated a protocol that works for arbitrary mathematical functions (A. C.-C. Yao, 1986), which was soon extended to multiple parties: the birth of MPC (Goldreich, Micali, & Wigderson, 1987). Due to the limited computing power available at the time, it remained a purely academic subject for a long time. It was not until 2004 that FairplayMP (Ben-David, Nisan, & Pinkas, 2008) appeared, a working system based on Yao's protocol. The first large-scale, commercial application followed in 2008 in Danish auctions for sugar beet contracts. This implementation demonstrated the practical value of MPC by both safeguarding bidders' privacy and reducing operational costs (Bogetoft et al., 2009).

Since then, the adoption of MPC has accelerated thanks to improved protocols, increased computing power and stricter privacy legislation. Nowadays, the technology is primarily used for securing private keys (such as in crypto wallets) and analysing sensitive data, including medical or financial information (Zhao et al., 2019).

To understand the practical applications of MPC, it is essential to distinguish between the foundational protocols that enable secure computation. While numerous variations and optimizations exist, most modern implementations rely on two primary cryptographic paradigms: Garbled Circuits and Secret Sharing.

2.2.1 Garbled Circuits

Stemming directly from Yao's early work (A. C.-C. Yao, 1986), Garbled Circuits (GC) allow two parties to securely evaluate a boolean circuit. In this protocol, one party (the garbler) encrypts the logic gates of the circuit, while the other party (the evaluator) computes the result using Oblivious Transfer (OT) to retrieve the necessary keys without revealing their inputs. While highly effective for functions with low circuit depth, GC traditionally scales poorly when introduced to a larger number of parties and requires significant network bandwidth (Evans, Kolesnikov, & Rosulek, 2018).

2.2.2 Secret Sharing

For computations involving three or more parties, Secret Sharing schemes, such as Shamir's Secret Sharing (SSS) (Shamir, 1979), are more commonly deployed. In this approach, a secret value is divided into multiple 'shares' and distributed among the participants. No individual share reveals information about the original secret; the data can only be reconstructed when a predefined threshold of shares is combined. Protocols based on SSS, such as the BGW (Ben-Or, Goldwasser, and Wigderson) protocol (Gilad & Yehuda, 2013), are highly optimized for arithmetic operations like addition and multiplication. This makes them particularly advantageous for executing complex database queries, aggregating metrics, and performing statistical analyses within distributed systems.

2.2.3 Advanced and Hybrid Protocols

Recent advancements have led to the development of highly efficient protocols like SPDZ (Damgård, Pastro, Smart, & Zakarias, 2012). SPDZ utilizes a preprocessing phase, often based on Somewhat Homomorphic Encryption (SHE), to generate correlated randomness. This allows the "online phase" to be remarkably fast and lightweight, even in environments with malicious adversaries. Such hybrid approaches are increasingly favoured in enterprise settings where analytical performance and scalability over large datasets are critical.

3 Methodology

3.1 Interviews

Semi-structured interviews form the basis for gaining an understanding of the requirements and wishes for integrating MPC technology with data warehouse principles. Interviews were conducted to explore the ideas regarding the requirements and wishes for a data warehouse based on MPC technology. To accomplish this, two interviews were carried out with focus groups from RIGA, and one interview was conducted with a Product Owner from Linksight. In this way, practical requirements can be properly aligned with technical requirements and limitations.

In the first set of interviews, RIGA staff were asked about three topics: the limitations they faced with the current working method, functional requirements for a new system, and any non-functional requirements. They were also asked to sketch out their ideas for an architecture on paper. Furthermore, these interviews were conducted in focus groups of two people each, so that staff could arrive at new insights together.

In addition, an interview was conducted with the Product Owner of Linksight's MPC system. The focus of this interview was on the (future) capabilities of the system and the limitations of MPC technology and Linksight's implementation of it. The feasibility of the ideas that emerged from previous discussions was also assessed.

Where necessary, follow-up correspondence was used to clarify and refine the points discussed during these sessions.

3.1.1 Analysis of interviews

For the purposes of text analysis, all interviews were recorded with the participants' consent and manually transcribed with speaker labels. The interviews were then coded in Atlas.ti. Relevant passages and ideas were assigned appropriate codes using 'open coding' and grouped together using 'axial coding'. Finally, the most important aspects were selected via 'selective coding' for use in further analysis.

3.2 System architecture

Based on existing theory and insights gained from the interviews conducted, a possible architecture has been outlined for a system based on the C4 model (levels 1 and 2). The various components of the system build on the current implementation of MPC by Linksight.

3.3 Testing of Linksight v3.14.1

To measure the performance of Linksight's MPC technology, various analyses are carried out in the development environment, each focusing on a different feature or protocol. The speed at which the analyses are performed is measured as the size of the datasets and the number of parties involved in the analysis increase. A set of analyses has also been devised to test the speed of filters across different data types (text, date-time and option). To measure speed, only the duration of the analysis on MPC itself is considered, excluding any processing time for checks. To prevent latency, all data stations are installed on a single Hetzner EX44 server with an Intel® Core™ i5-13500

processor and 64 GB of DDR4 memory. Data stations are connected to each other via an internal network and are not individually limited in the amount of computing power they use.

A synthetic dataset was generated for this experiment, consisting of a main table of residents (Table 1) and a related sub-table of GP network alerts (Table 2). A random number of alerts has been linked to each resident in the main table. In practice, this means that some residents might have 0 alerts while others have 8 for example.

In order to evaluate the system at different scales, datasets were generated for populations of 100, 1.000, 10.000, 100.000 and 1.000.000. In each case, the overall ratio of population to reports was 1:4 (meaning that a dataset for a population of 100 contains a total of 400 reports).

Ultimately, each dataset is distributed across 2, 3, 4 or 5 data stations. When distributed across n stations, each station is assigned a random $\frac{1}{n}$th portion of the data. Because this allocation is made with replacement, data records may overlap across multiple data stations. This is useful, since some of the testing queries try to find overlapping data. Not having this overlap would cause queries to exit early and not touch all datastations. The total number of rows in circulation always remains the same.

Table 1: Dataformat for main table (residents)

Column name	**Data type**	**Remark**
hash_bsn4	String	Primary key
geboortejaar	Integer	

Table 2: Data format for sub-table (GP alerts)

Column name	**Data type**	**Remark**
hash_bsn4	String	Reference key
netwerkmonitor_id	String	Primary key
urgentie_optie_kort	Option (String)	Format: U1, U2, ... U20
urgentie_tekst_kort	String	Format: U1, U2, ... U20
urgentie_optie_lang	Option (String)	Format: Deze melding is van niveau x
urgentie_tekst_lang	String	Format: Deze melding is van niveau x
urgentie_getal	Integer	Format: 1, 2, ... 20
urgentie_float	Float	Format: $\frac{x}{20}$
vraag_datumtijd	DateTime	Between 2020-01-01 00:00:00 and 2025-12-31 23:59:59

The following queries have been created to test the system's various protocols:

- Statistics and filters: How many residents at data station A with a report where [FILTER] applies have at least one report at $n-1$ other data stations? This query is combined with the following options for [FILTER]:
  - No filter
  - `urgentie_optie_kort == "U1"`

- `urgentie_tekst_kort == "U1"`
- `urgentie_optie_lang == "Deze melding is van niveau 1"`
- `urgentie_tekst_lang == "Deze melding is van niveau 1"`
- `vraag_datumtijd` $\geq$ `2025-01-01 00:00:00`

Internally, this query uses a custom protocol based on homomorphic encryption and secret sharing.

- Group by: How many residents are registered per year of birth in a federated dataset comprising n data stations? This method is implemented using a different custom protocol using homomorphic encryption and secret sharing. Also known as Secure Histograms.
- Deduplication of a list: How many unique residents are there, and at how many data stations are they registered, in a federated dataset comprising n data stations? The feature uses just homomorphic encryption.

4 Results

4.1 Interviews

During interviews with RIGA staff, a clear need emerged for a data warehouse based on MPC technology. At present, there is no clear and comprehensive integration of the Linksight system for Multi-Party Computation with their data warehouse environment. Despite generally limited experience with MPC, there is a strong recognition of the value in combining both concepts; both the privacy and legal-technical reasons for using MPC and the scalability and iterative nature of the data warehouse are proving to be important.

There are many ideas about which components an ideal implementation of a data warehouse based on MPC technology should include. The introduction of the so-called 'Workbench' or 'Workspace' is often cited as a positive development. While originally discussed as a way to transform diverse incoming datasets into standardized formats, later corrections by the interviewees redefined its purpose. Because uniform datasets are strictly enforced at the point of upload, the Workbench is not intended for initial standardization. Instead, the stakeholders specified three primary functions for this component:

1. The ability to supply support tables for data enrichment and easier, flexible grouping.
2. Utilizing hierarchical date functions (e.g., selecting, grouping, and filtering by year, quarter, or month).
3. Performing (date) calculations, such as determining age or the duration between a start and reference date.

The notion is that enriching data in this way will simplify analyses. In addition, a component that provides insight into data quality is important. In particular, the completeness of the data is vital for the usability of analysis results. Timeliness, on the other hand, is less important, as registrations often lag behind in practice.

The team also anticipates a number of challenges and bottlenecks. For instance, the interviewees still have many concerns regarding the speed and scalability of Linksight. Multi-Party Computation involves a significant amount of overhead, which greatly increases the processing time of a query. As this process is also highly computationally intensive, all servers and data stations must meet the correct specifications to prevent (even) slower analyses. One underperforming data station could cause the entire analysis to almost grind to a halt. Interviewee C also believes that servers that are too slow could have financial implications: "Suppose the analysis speed becomes too limiting. At some point, we would have to ask all organisations for more RAM or storage. Those are the consequences when we talk about performance. And, of course, there is a cost involved."

MPC technology also supports the implementation and enforcement of privacy measures to ensure compliance with legal requirements, in line with existing legislation. Measures such as k-anonymity (concealing results when fewer than k individuals remain) and rounding results to the nearest multiple of five were mentioned as ways of keeping the results as (pseudo)anonymous as possible. Later experimentation by one of the interviewees led to the discovery that k-anonymity could be circumvented in certain instances. If $A + B + C = 102$ and $A + B = 100$, one can reason that $C = 102 - 100 = 2$. He found that during a small test on real world data, 14 out of 74 analyses (almost 19%) had rest values below the k-anonymity threshold of $k = 10$.

Furthermore, the Linksight system does not provide access to the content of individual rows: results are, after all, only returned as aggregated data. Insights into the data are currently limited to totals, averages and standard deviations. There is still disagreement within RIGA regarding the latter option, as some believe this would be too revealing. The same issue applies to any median and min-max options. Choices made in this regard can significantly hinder analyses.

From Linksight's perspective, there appear to be few technical objections to implementing such a system. In principle, they operate as technology-agnostic as possible: new protocols, database types and other technologies could be added to their software relatively easily. The features they develop are primarily demand-driven; provided they have sufficient time and resources, they are willing to make the necessary changes to their system. RIGA's concerns regarding the speed of the system and network could be largely resolved with the help of a dedicated 'helper station'. This is a server with extra processing power, which can facilitate communication via MPC between data stations. This would allow more calculations to take place in plaintext, which should lead to significant improvements in processing speed. In theory, a standard data station that does not deliver data itself for a specific query could also act as a helper station. Some protocols even require such a helper station to facilitate necessary calculations. Furthermore, running queries during off-peak hours could potentially lead to faster processing.

Lastly, Linksight has heard from other clients that some healthcare institutions prefer a data pipeline over manual uploads. This could save staff time, as new information could be synchronised automatically. At the same time, there are also organisations which, due to the sensitive nature of the information, prefer to share it manually with their data centre.

4.2 System architecture

To give concrete form to the insights gained from the interviews, a conceptual system architecture has been developed for a bottom-up data warehouse based on MPC technology. The focus here is on the parts of the system that are new or have changed. These have been elaborated in a C4 model. In this model, a conceptual system is depicted at two levels of abstraction: system context

(level 1) and container level (level 2) for the data stations and the data warehouse engine.

Figure 1 illustrates the overall structure of the data warehouse. Above is a traditional data warehouse, which connected to the Data Warehouse Engine illustrates a simple bottom-up design. Below is a system using the Linksight system. The design allows for them to live alongside each other. If desired, usage of just the Linksight system is also possible. Both systems are supported by a Reference Data Management system, which is responsible for providing data structures and support tables to the engine, the data marts and the data stations. These are supplied by an administrative staff member, in consultation with the data providers and RIGA's data analysts. Using the supplied data structures, standardisation of the databases on the data stations is enforced remotely. This reduces the workload of the data stewards at the supplying parties, as they currently still have to attach these specifications themselves when uploading data. Because this normalisation already forces the data into 3NF form and there is no access to the rows of the data stations, a bottom-up model has been chosen. While data stations and federated datasets function as de facto datamarts, it is important to note that row-level analyses (like joins and filters) between two encrypted/private datasets can only happen within the Linksight network.

Due to the design of the Linksight system, queries must be submitted directly to a data station. Any further communication between the data stations is handled by the data stations themselves. (Provided the governance rules permit it) requests can also be made to a single data station without involving Multi-Party Computation. Should an auxiliary station be required, this can be achieved by adding an empty data station under the management of RIGA. When an auxiliary station is required for a transaction, it can be dynamically selected from an existing data station or the helper station (depending on availability). This ensures that communication is not forced to take a longer detour when this is not necessary. It also ensures that the auxiliary station can be used for transactions that require significant computing power.

4.2.1 Data station

Within a data station, there are various stages designed to ensure data quality, security and privacy. Figure 2 zooms in on Data Station 1 (from Figure 1). The data follows a clear data flow. First, the datasets are automatically supplied by a data system or manually prepared and uploaded by a data steward. Based on the data structures from Reference Data Management, the upload is approved or rejected. A conscious decision has been made to use an ETL process whereby the data must already be correctly transformed before being supplied. To limit the amount of sensitive information on the data station as much as possible, it is important that this is (pseudo)anonymised or removed before the upload. Incorrect uploads are therefore rejected more quickly, as it is possible to check immediately whether the datasets have the correct structure (rather than them first having to be transformed). Furthermore, this reduces the amount of overhead required for data transformation, particularly given that the existing data formats of contributing parties can vary considerably.

Following a successful upload, the dataset is stored in a staging database. As the supplied support tables may change in the meantime, it is important that new tables can be constructed based on the latest valid upload. These transformations are carried out in the 'Workbench', again based on information supplied by the RDS system. In this component, any optimised variants and indexes of the table are also created, insofar as the production database system does not already facilitate this. Given the current implementation by Linksight, the databases run on SQLite. The current version of the data (both upload and support tables) is maintained and shared with the

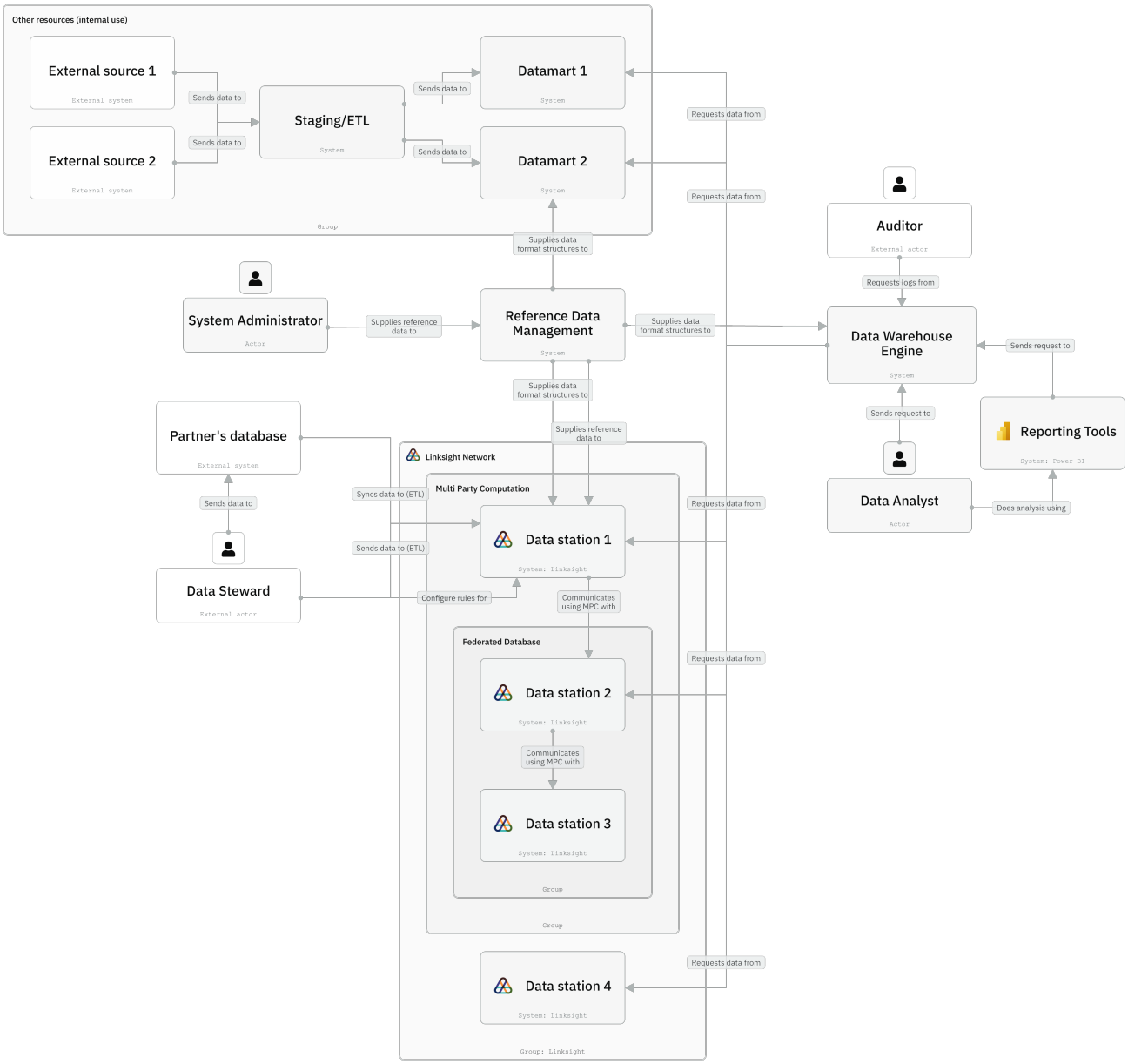

Figure 1: C4-model at level 1

engine for caching purposes.

When a query is received from the data warehouse engine, it is first checked and processed by the Privacy & Governance Enforcer. The rules governing this are established in consultation with the other parties involved in a data partnership. This module ensures that queries attempting to perform unauthorised operations are blocked. This is also the layer in which any privacy measures, such as k-anonymity and rounding, take place. Communication then takes place with other data stations and the organisation's own database via the Cryptography Manager. This ensures that everything is correctly encrypted where necessary and facilitates communication via an appropriate MPC protocol.

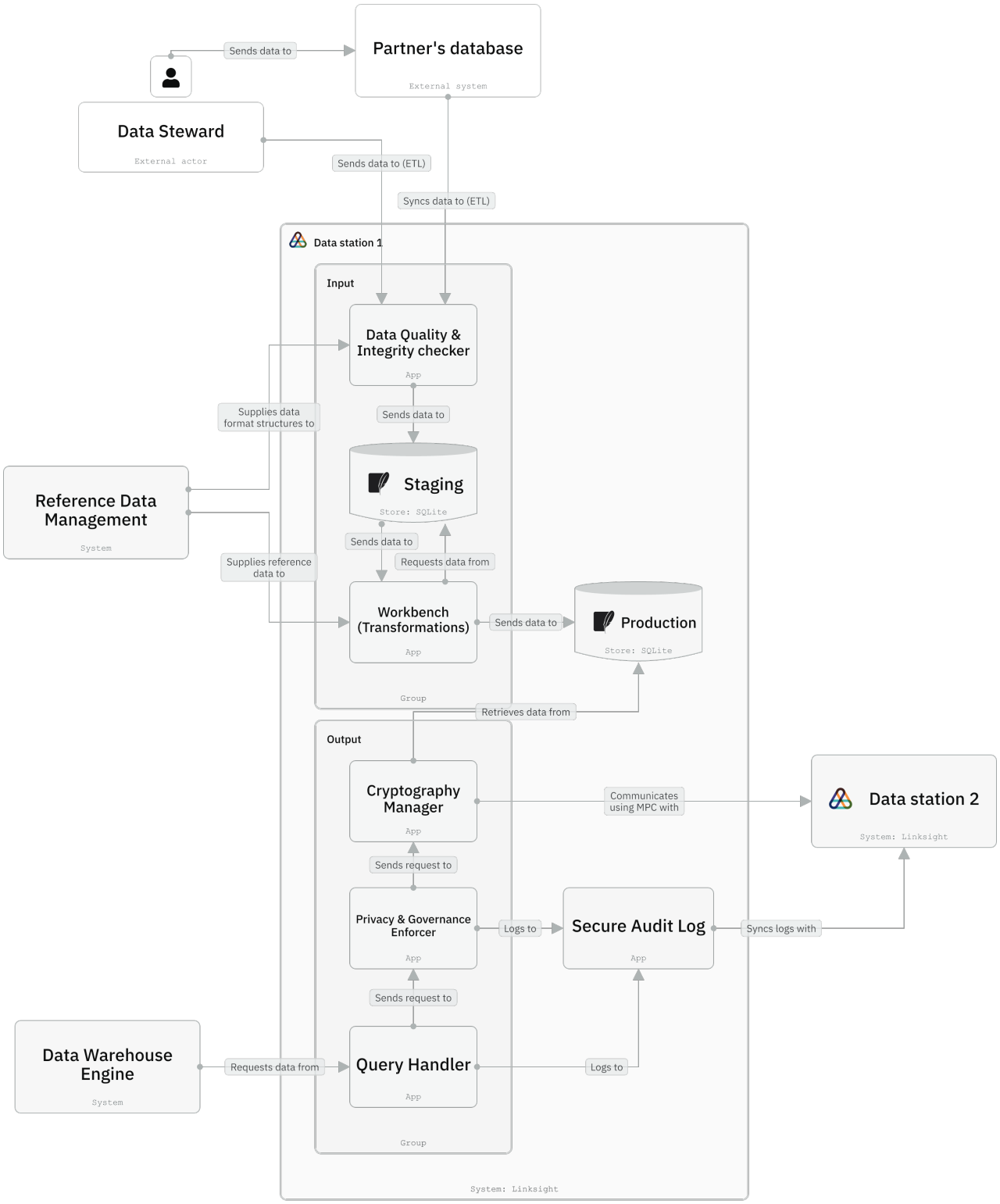

Figure 2: C4-model at level 2, zoomed into the data station

Finally, changes and queries are logged in the tamper-proof audit log. This log is shared with all other data stations within the same data collaboration, ensuring that all participating parties have a clear overview of the use of the data stations. Queries are linked to metadata from the engine (such as the research for which a query was made and whether the result has already been processed by the data warehouse engine's cache). It is, of course, important for the security and reliability of the system that any requests not handled by the data stations but processed by the cache are also logged in the audit log.

4.2.2 Data warehouse engine

The final model, figure 3, shows the engine that coordinates the data warehouse. A data analyst uses an interface to communicate with the API and Semantic Gateway in order to run queries. Reporting tools like Power BI can also use this entry point for requesting data. These queries can also be set up as recurring queries in the Job Scheduler: this allows a query to be run repeatedly according to a schedule, ensuring that the latest information is already stored in the cache.

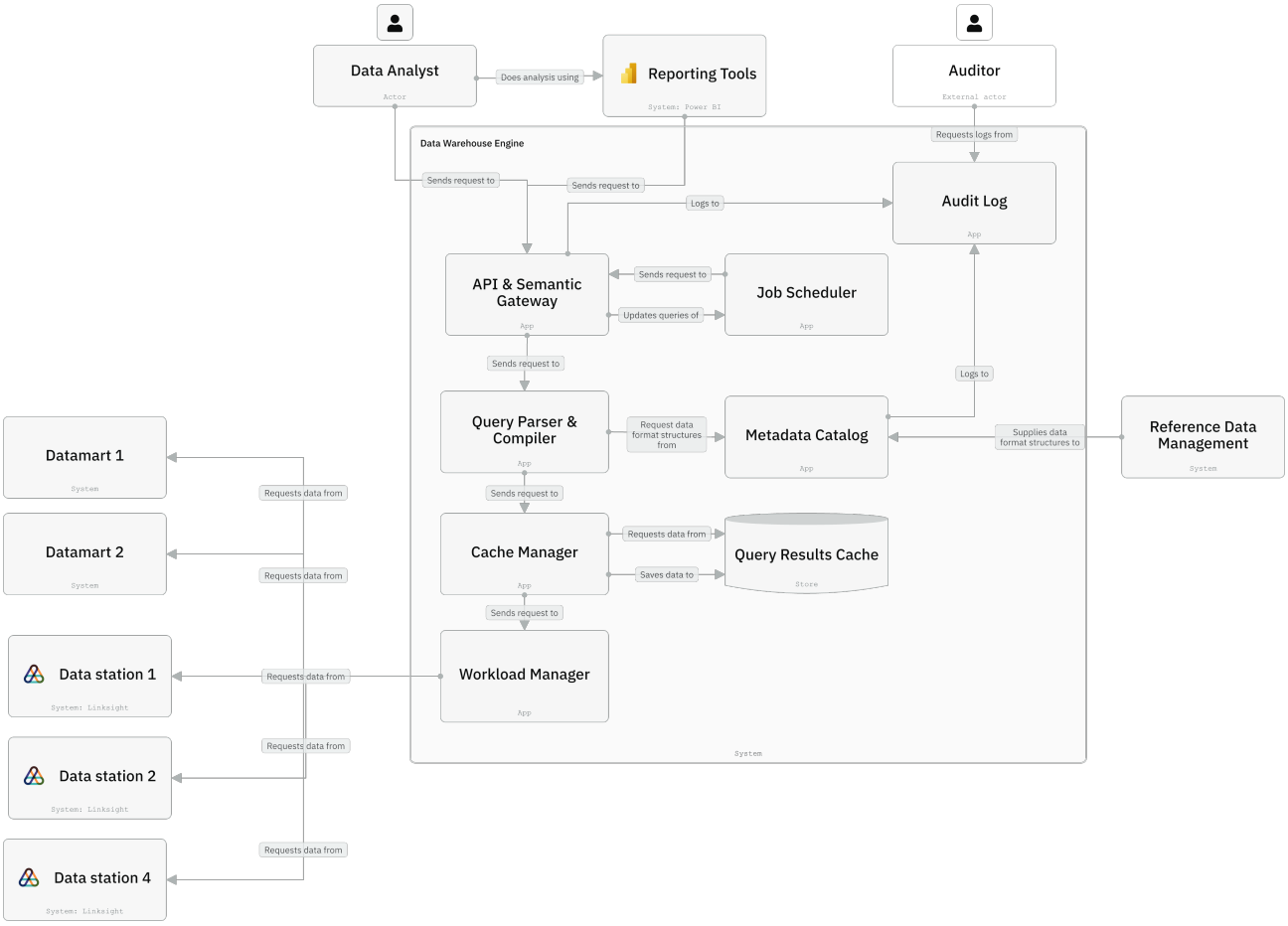

Figure 3: C4-model at level 2, zoomed into the data warehouse engine

When a query is received, it is first checked by the Query Parser & Compiler. To do this, this component uses the Metadata Catalogue, which contains all the metadata relating to the data

marts and data stations in the system. Data formats and statistics about the tables are also stored there. If a query is incorrect or will not return any results, it can be rejected at the outset.

The query is then forwarded to the Cache Manager. Depending on the capabilities of the systems behind the data marts and data stations, the cache is set up based on a dependency graph or a static Time-To-Live. In the case of a dependency graph, the version of the datasets used is taken into account when building the cache. If the required datasets are still of the same version and a result is available in the cache, the result is returned from the cache. This way, the lifecycle of the data is automatically taken into account to avoid unnecessary stale data. If this mechanism is not possible for any reason, results are cached using a TTL. Results are then stored for a fixed period and served rather than calculated. Regardless of whether the results are retrieved from data stations/data marts or from the cache, the request is logged both in the engine and in the data marts and stations that form part of the query.

The execution of queries is coordinated by the Workload Manager. Depending on the expected size and priority, a smart decision is made as to when a task is actually executed. For example, non-critical, computationally intensive tasks are scheduled for the night-time process in order to conserve computing power during the day for more important queries.

Because the architecture strictly prevents row-level data extraction, generating OLAP cubes via a traditional, centralized server is impossible. The generation of these multidimensional cubes can still be achieved using the aforementioned components of the engine:

1. Have the data analyst create the queries necessary to capture all the dimensions of the datasets using the known data structures.

2. Submit these queries to the Job Scheduler, which functions as the scheduled trigger for generating the cube.

3. When triggered, the queries are sent forward to the Cache Manager and Workload Manager like any other analysis.

4. If there isn't a valid cache available for a request, they are sent over to the responsible data stations. The aggregation calculations are performed using the implemented MPC protocols, ensuring the data remains private. These results are cached in the Cache Manager.

5. Within the presentation layer, the individual cached calculations can be retrieved through the API & Semantic Gateway to reconstruct the OLAP cubes.

6. Any missing calculations can be queried on the fly and added to the Job Scheduler for later use.

4.3 Testing of Linksight v3.14.1

4.3.1 Filters

Figure 4 shows the processing time of the queries plotted against the size of the dataset. Each point on the graph represents a different filter, which is especially visible for the dataset size of 10^6 citizens. The points are coloured according to the number of participating data stations (variable `split`).

Whilst the processing time for datasets of up to 10,000 inhabitants remains relatively constant at around 10 seconds, a (more than) exponential increase can be seen beyond this point. Whilst the processing time doubles to 20 seconds for 100,000 inhabitants, it rises to approximately 1,600 seconds, or 30 minutes, for a population of 1,000,000.

Furthermore, a pattern can be observed in the distribution of query processing speeds: for datasets containing around 100,000 residents, it appears more efficient to distribute the data across more data stations rather than fewer. The variation in processing times also appears to be much less pronounced around this size.

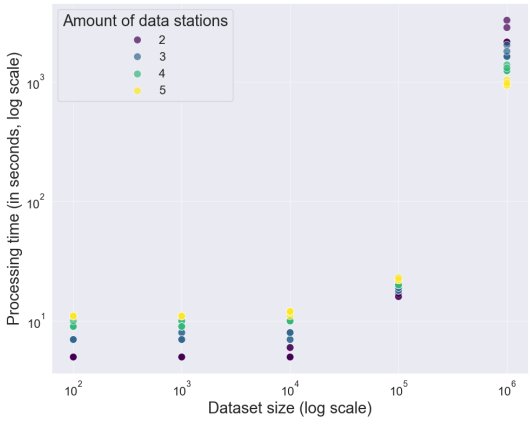

Figure 4: Processing time vs. dataset size, grouped by amount of data stations for filters

Figure 5 compares the relative speed of analyses using the various filters with the same analysis without a filter. There is a separate graph for each number of data stations, with a margin of ± 1 second. This margin has been included because the processing time in the Linksight system is rounded to the nearest second.

As the dataset grows, the filters have an increasingly significant impact on processing speed. From 100,000 residents onwards, the use of filters results in a clear reduction in processing time. With a dataset of 1,000,000 rows in the main table and two data stations, this generally leads to a 35% reduction in processing time. The more data stations are introduced, the lower this gain becomes. With five data stations, the query ran only 10% faster for most filters. Although all filters operate at roughly the same speed, this does not appear to be the case for filtering by `vraag_datumtijd`. On average, this is about half as fast as other filters.

When creating the Spearman correlation heatmap in Figure 6, the filters (a categorical variable) were converted into so-called dummy indicators. This means that each filter was transformed into a standalone binary variable. Whenever a filter is active, that variable resolves to `true` while all others remain `false`. Since only one filter is active at a time and each filter is active an equal number of times, the correlation for the other 5 filters is always -0.20 (or $-\frac{1}{5}$).

It can be seen that the dataset size is strongly correlated with processing time ($+0.81$), with a slightly positive correlation between speed and the number of data stations ($+0.35$). The latter is striking, given that somewhere beyond 100,000 inhabitants there is actually a negative correlation with processing time (as shown in Figure 4).

When the graphs are broken down into populations of ≤ 100.000 and $1.000.000$ in Figure 7, the contrast becomes clear. Whilst the correlation rises to $+0.60$ for populations of ≤ 100.000, it drops

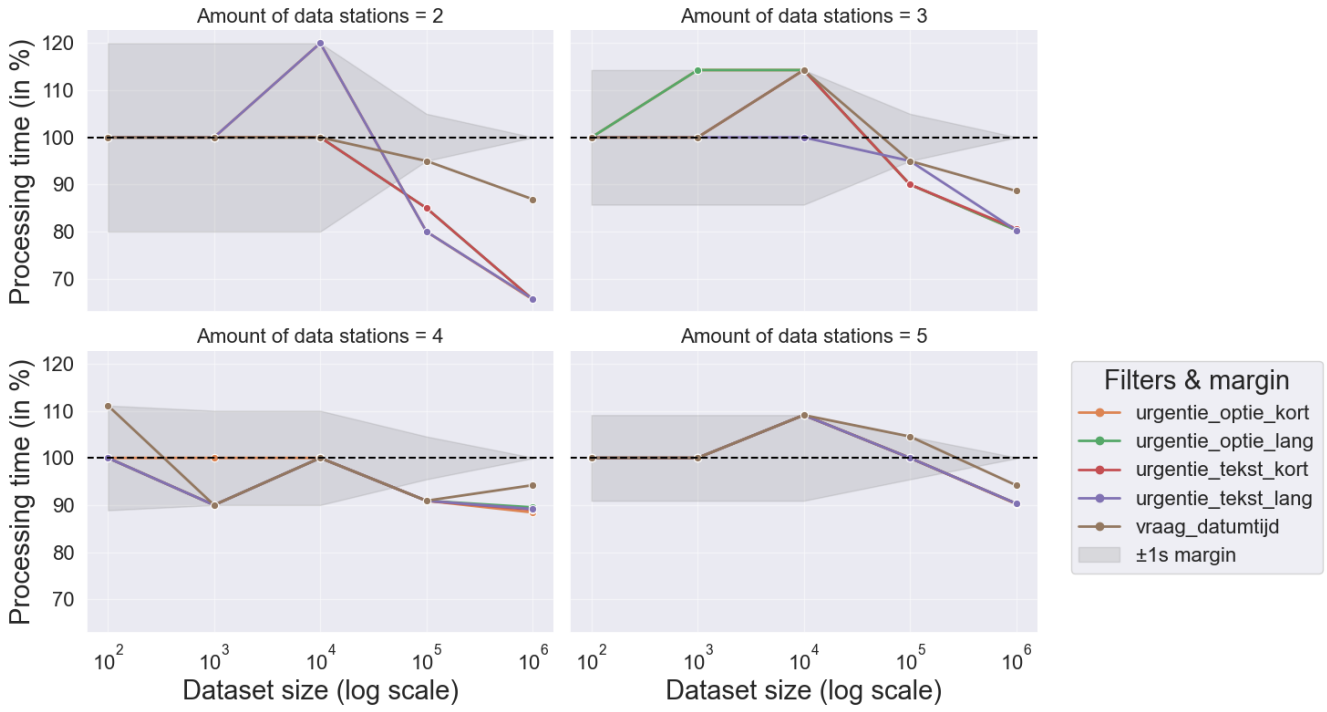

Figure 5: Relative processing speed of filters at different dataset sizes and amounts of data stations, using no filters as baseline (100%)

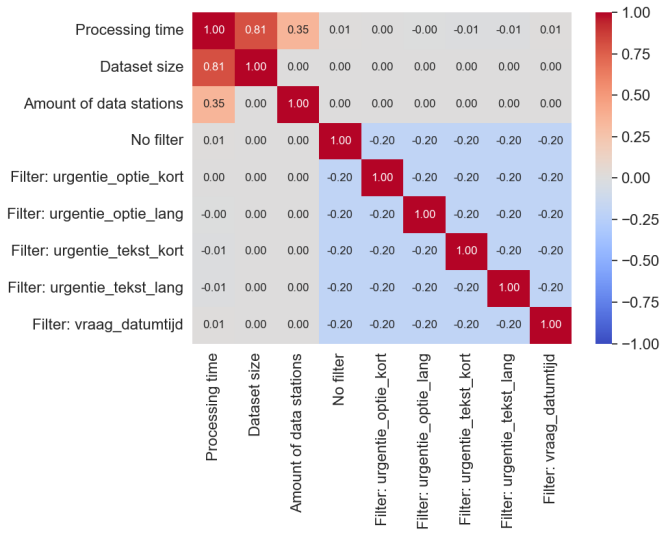

Figure 6: Spearman correlation heatmap for filters

sharply to −0.97 for the largest dataset. This confirms the pattern described earlier, namely that adding data stations to very large datasets actually results in a speed increase for these queries. The correlation between size and processing time weakens slightly, but remains present.

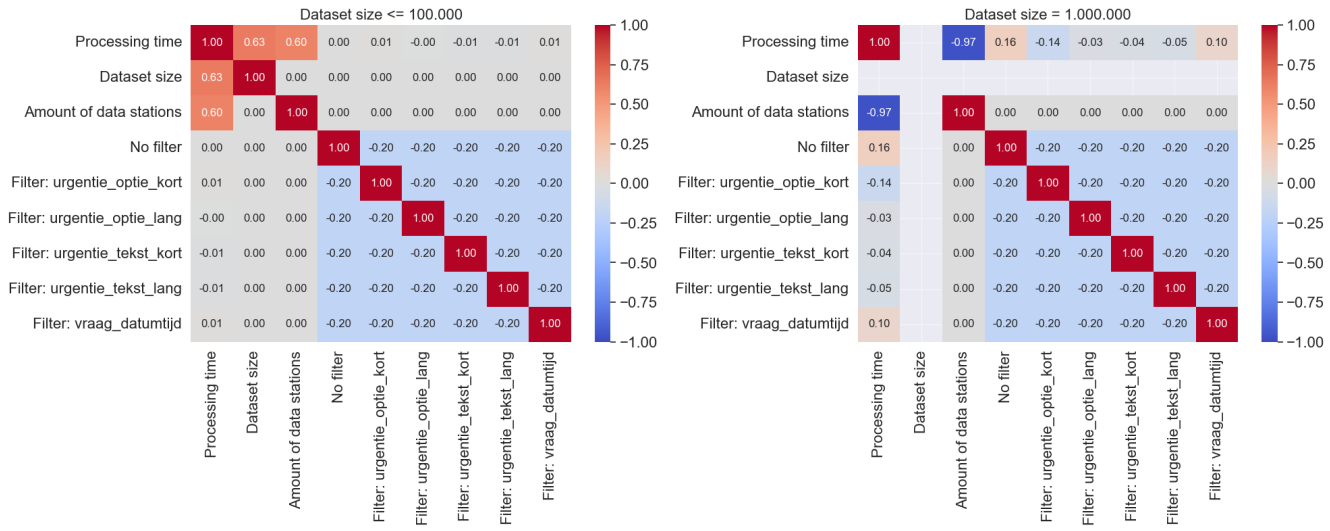

Figure 7: Spearman correlation heatmap for filters, split up in $\leq$ 100.000 citizens and 1.000.000 citizens

4.3.2 Group by

Figure 8 shows the processing time for grouping by year of birth (`geboortejaar`) plotted against the size of the dataset. The data points are coloured according to the number of participating data stations (variable `split`). As the Linksight implementation for grouping always requires a data station to act as an helper station, the results are limited to 2, 3 and 4 data stations.

With a population of 1.000.000, the processing time remains manageable: the process takes an average of 300 seconds (or 5 minutes). Distributing the data across multiple data stations is the most efficient approach, regardless of the size of the datasets.

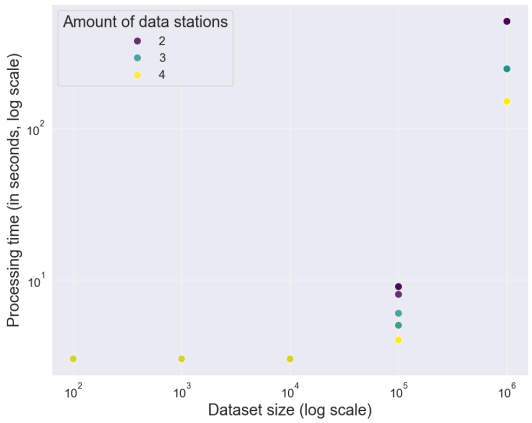

Figure 8: Processing time vs. dataset size, grouped by amount of data stations for group by

The Spearman correlation heatmap in Figure 9 shows that processing time is strongly correlated with the size of the dataset. The number of data stations, on the other hand, has a negative correlation. These results are consistent with those shown in Figure 8.

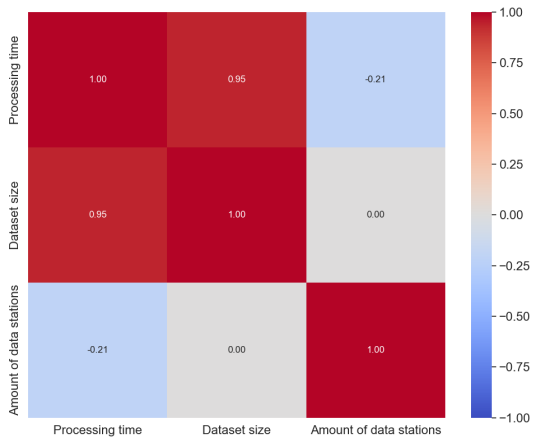

Figure 9: Spearman correlation heatmap for group by

4.3.3 Deduplication of a list

Figure 10 shows the processing time for deduplication plotted against the size of the dataset. The data points are coloured according to the number of participating data stations (variable `split`). In general, deduplicating lists takes just a few seconds, regardless of the size of the datasets. With 5 data stations, however, the processing time is longer.

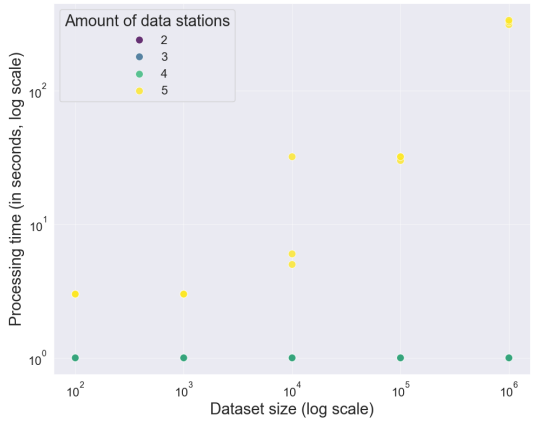

Figure 10: Processing time vs. dataset size, grouped by amount of data stations for deduplication

According to Figure 11, there seems to be a clear connection (+0.87) between the amount of data stations and the time it takes to complete the operation. The size of the dataset also seems to matter to a small degree (+0.24).

5 Evaluation

5.1 Interviews

The qualitative data collection provided foundational insights into the functional and non-functional requirements for an MPC-integrated data warehouse. The semi-structured interviews were conducted

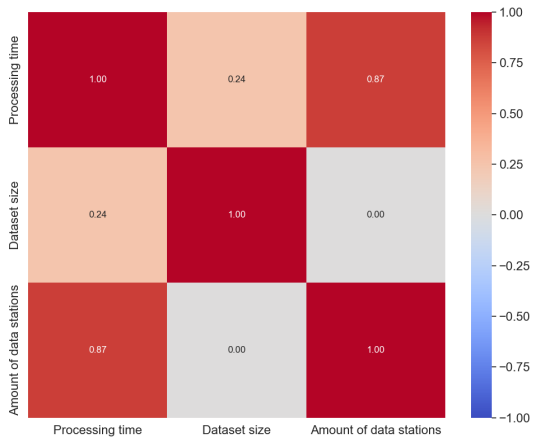

Figure 11: Spearman correlation heatmap for deduplication

with a highly relevant, albeit small, sample. While this sample size is limited, it effectively captured the perspectives of both the primary end-users (data analysts at RIGA) and the technical facilitators (Linksight). It is worth noting that the RIGA staff generally possessed limited prior experience with MPC technology. This likely influenced their initial architectural expectations, requiring the translation of traditional data warehousing concepts into MPC-compliant realities during the analysis phase.

The presented requirements may differ from person to person use case to use case. While these needs were mostly based on from the perspective of RIGA and its business activities using the Linksight system, they do contain notions necessary for any data warehouse based on Multi-Party Computation. Just like any other system, these requirements may change over time (especially with technological advancements or new and improved MPC protocols).

Finally, the interviews show a clear tension between the desires of data analysts and the inherent privacy constraints of MPC technology. Given the specificity of these restraints, there aren't any clear guidelines on what method should or should not be implemented. Given the sensitivity of the data handled, privacy laws like the GDPR/AVG are strict. As such, integrating MPC into a data warehouse is not solely a technical challenge, but also an organizational/legal one, since mistakes made in this department could ultimately cause the demise of the entire project.

5.2 System architecture

A key consideration in the proposed design concerns the strict bottom-up approach and the shift in the Extract, Transform, Load (ETL) process. By requiring data to be supplied in 3NF format and (pseudo)anonymised, this prevents sensitive information from ending up on the data stations unnecessarily or entering the network unintentionally. Furthermore, any incorrect uploads can be rejected directly at the data station using formats from the Reference Data Management system. Although this high degree of standardisation ensures the predictability and interoperability of the data, it simultaneously introduces an organisational challenge. This is because the complexity of data transformation shifts primarily to the supplying parties within the data collaboration. This requires a high degree of accuracy from the supplying data stewards, which in practice can constitute a functional barrier to participation.

Moreover, the integration of the Cache Manager, Job Scheduler and Workload Manager within the data warehouse engine should not be viewed merely as performance optimisations, but as architectural requirements designed to address the technological limitations of MPC. Performance tests revealed that complex grouping and filtering operations on datasets of considerable size result in a time complexity of the order of $O(2^n)$, with processing times that can run to thirty minutes. To keep the system practical for data analysts in real-world use, it is crucial to proactively compute heavy queries during off-peak hours (for example, via night-time processes) and to serve results asynchronously via a Time-To-Live (TTL) or dependency graph.

5.3 Testing of Linksight

The results for both filtering and grouping revealed an exponential increase in processing time as the datasets grew. Whilst datasets of up to 100,000 residents were processed relatively quickly, the processing time for datasets of 1,000,000 residents sometimes rose to as much as thirty minutes. This sharp increase suggests a time complexity of the order of $O(2^n)$ for such queries. This also explains the result from the Spearman heatmaps, where the correlation between dataset size and processing time increases proportionally with the extreme values of the dataset.

A striking pattern in the results is the interplay between dataset size and the number of data stations. With relatively small datasets, having more data stations is often counterproductive, but once the population reaches around 200,000, it is more efficient to spread the data more widely. Two mechanisms appear to underlie this:

- Greater row reduction: With a wider distribution across stations, the total number of rows that actually need to be processed in the query is lower. In a theoretical model with two data stations and 1.000.000 citizens, $\sum_{i=1}^{2} 1.000.000(\frac{1}{2})^n = 750.000$ rows are included in the calculation. When spread across five stations, this drops sharply to just $\sum_{i=1}^{5} 1.000.000(\frac{1}{5})^n \simeq 250.000$ rows.

- Parallel processing: In operations such as grouping, utilising more data stations allows tasks to be executed more effectively in parallel, thereby reducing the overall processing time.

Although the relative speed gains were similar for most filters, the filter on `vraag_datumtijd` lagged behind in performance. One possible explanation for this is the selectivity of the query. Whereas other queries reduce the set to, for example, one-twentieth, this filter may only reduce the dataset to one-fifth.

Finally, there is a noticeable delay when deduplicating lists across five data stations. This is likely due to the protocols chosen for MPC. Because Linksight uses one data station as an 'helper station' as a broker for certain (more efficient) calculations, the absence of such a station (which is the case in the configuration with five active data stations) results in a protocol switch. The system then falls back on an alternative process that is considerably slower, which explains the sudden increase in processing time.

5.4 Further research

In particular, testing the filters is an important aspect that could be explored further in a follow-up study. Although some conclusions can be drawn from the results, it is not certain whether

these trends would also emerge with a different query design. It would be valuable to assess the generalisability of these time notations.

The test setup may also have influenced the processing time. Instead of a fixed total amount of data in circulation, one could, for example, opt for a fixed number of residents per data station. This may offer different insights to those mentioned above. As an alternative to this, one could calculate, per query and number of data stations, how many rows are realistically involved in the query. If the datasets are generated and distributed intelligently, this would allow for the connected residents to be taken into account. The test suite could also encompass other features and analyses that will be utilized frequently by RIGA in production settings. An example of this would be filtering on booleans (i.e. excluding citizens who have opted out of data processing for research purposes) or the performance of different aggregation types (i.e. mean, total and standard deviation). Performance of process mining would be an important feature to research, when this becomes available on the platform.

The Linksight system is still under active development, which means that the speeds and scalability of the Multi-Party Computation protocols used may still change. As a result of these changes, the findings of this study may no longer be representative or relevant to a newer version of the system. It is important to periodically retest all key queries and protocols to ensure that the findings remain accurate.

This study was conducted under virtually ideal conditions, with all data stations running on the same server (the demo environment of Linksight). In practice, processing times may vary considerably due to additional factors such as network capacity, the physical distance between servers, and the computing power of external data stations. By tracking the processing times of requests to the production system, it is possible to investigate the extent to which these factors influence query speeds.

6 Conclusion

This research set out to investigate how Multi-Party Computation (MPC) can be deeply integrated into a data warehouse system to enable secure, privacy-preserving analyses. Through a combination of stakeholder interviews, architectural modelling and performance testing, this study demonstrates that such a system can be viable, if it accounts well for scalability issues of the technology.

To address the restrictions caused by MPC (like loss of row-level access), the proposed system introduces a strict bottom-up architecture where individual data stations function as de facto data marts. This design leans on a shift in the Extract, Transform, Load (ETL) process to ensure privacy and predictability; by requiring data to be supplied and standardized in 3NF format before entering a Linksight data station, the architecture successfully minimizes the risk of sensitive information unintentionally circulating within the data stations and ensures that the underlying data format is known to data analysts.

The inherent loss of performance by MPC is accounted for by the Cache Manager, Job Scheduler and Workload Manager in the proposed design. By proactively computing computationally intensive queries during off-peak hours and serving cached results, the system maintains practical utility for data analysts without compromising security. The cache is refreshed using a static Time To Live (TTL) or by analysing changes in the underlying datasets using a dependency graph. To ensure compliance with rules and regulations, an audit log is kept at each system that handles sensitive

data.

Performance evaluations of the Linksight system confirmed the heavy computational overhead inherent to MPC technology. Testing revealed a more than exponential increase in processing time for grouping and filtering operations, exhibiting a time complexity in the order of $O(2^n)$ and reaching processing times of up to thirty minutes for large datasets. Distributing data across a higher number of data stations generally reduced processing time through a combination of greater row reduction and parallel processing for some of the queries used. For some queries, usage of a helper station is either required or heavily recommended to circumvent the use of less efficient protocols.

Moving forward, further research should evaluate the system under real-world network constraints, as the physical distance and varying computing power of external data stations will inevitably impact query speeds. Additionally, as MPC protocols and the Linksight platform continue to undergo active development, continuous re-evaluation of query performance and system requirements will be essential to maintaining a future-proof, secure data warehousing infrastructure.

Given the novelty of this specific area of study, it is important to continue research and development. RIGA has the opportunity to become a trailblazer on this topic by establishing the GRIP platform as a standard-setting blueprint for secure, multi-party data collaborations across the wider healthcare sector.

References

Ben-David, A., Nisan, N., & Pinkas, B. (2008). Fairplaymp: a system for secure multi-party computation. In *Proceedings of the 15th acm conference on computer and communications security* (p. 257–266). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1455770.1455804 doi: 10.1145/1455770.1455804

Bennink, N., & Drukarch, H. (2023, September 16). *Handleiding Privacy by Design* (Handleiding). Ministerie van Justitie en Veiligheid. Retrieved from https://open.overheid.nl/documenten/8d338db4-cf0f-420b-8092-c8f7111e049c/file (Accessed: 2026-02-17)

Berisha, B., Mëziu, E., & Shabani, I. (2022). Big data analytics in cloud computing: an overview. *Journal of cloud computing*, *11*(1), 24.

Bogetoft, P., Christensen, D. L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., ... Toft, T. (2009). Secure multiparty computation goes live. In R. Dingledine & P. Golle (Eds.), *Financial cryptography and data security* (pp. 325–343). Berlin, Heidelberg: Springer Berlin Heidelberg.

Codd, E. F. (1993). Providing olap (on-line analytical processing) to user-analysts: An it mandate. *http://www.arborsoft.com/papers/coddTOC.html*.

Damgård, I., Pastro, V., Smart, N., & Zakarias, S. (2012). Multiparty computation from somewhat homomorphic encryption. In R. Safavi-Naini & R. Canetti (Eds.), *Advances in cryptology – crypto 2012* (pp. 643–662). Berlin, Heidelberg: Springer Berlin Heidelberg.

Evans, D., Kolesnikov, V., & Rosulek, M. (2018, December). A pragmatic introduction to secure multi-party computation. *Found. Trends Priv. Secur.*, *2*(2–3), 70–246. Retrieved from https://doi.org/10.1561/3300000019 doi: 10.1561/3300000019

Gath, S. (2024). *Principle of data warehousing*. Academic Guru Publishing House.

Gilad, A., & Yehuda, L. (2013). The bgw protocol for perfectly-secure multiparty computation. In *Secure multi-party computation*. IOS Press. Retrieved from http://dx.doi.org/10.3233/978-1-61499-169-4-120 doi: 10.3233/978-1-61499-169-4-120

Goldreich, O., Micali, S., & Wigderson, A. (1987, 01). How to play any mental game. , 218-229. doi: 10.1145/28395.28420

Gupta, C., Mehta, A., Wang, S., & Dayal, U. (2009). Fair, effective, efficient and differentiated scheduling in an enterprise data warehouse. In *Proceedings of the 12th international conference on extending database technology: Advances in database technology* (pp. 696–707).

Hassan, C. A. U., Hammad, M., Uddin, M., Iqbal, J., Sahi, J., Hussain, S., & Ullah, S. S. (2022). Optimizing the performance of data warehouse by query cache mechanism. *IEEE Access*, *10*, 13472–13480.

Hwang, H.-G., Ku, C.-Y., Yen, D. C., & Cheng, C.-C. (2004). Critical factors influencing the adoption of data warehouse technology: a study of the banking industry in taiwan. *Decision Support Systems*, *37*(1), 1–21.

Inmon, W. (1996). *Building the data warehouse*. Wiley. Retrieved from https://books.google.nl/books?id=9eJQAAAAMAAJ

Inmon, W., Linstedt, D., & Levins, M. (2019). Chapter 6.2 - introduction to data vault modeling. In W. Inmon, D. Linstedt, & M. Levins (Eds.), *Data architecture (second edition)* (Second Edition ed., p. 141-156). Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B978012816916200019X doi: https://doi.org/10.1016/B978-0-12-816916-2.00019-X

Inmon, W., Strauss, D., & Neushloss, G. (2008a). Chapter 2 - an introduction to dw 2.0. In W. Inmon, D. Strauss, & G. Neushloss (Eds.), *Dw 2.0* (p. 23-54). Burlington: Morgan Kaufmann. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780123743190000026 doi: https://doi.org/10.1016/B978-0-12-374319-0.00002-6

Inmon, W., Strauss, D., & Neushloss, G. (2008b). Chapter 3 - dw 2.0 components—about the different sectors. In W. Inmon, D. Strauss, & G. Neushloss (Eds.), *Dw 2.0* (p. 55-93). Burlington: Morgan Kaufmann. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780123743190000038 doi: https://doi.org/10.1016/B978-0-12-374319-0.00003-8

Inmon, W. H., Strauss, D., & Neushloss, G. (2010). *Dw 2.0: The architecture for the next generation of data warehousing*. Elsevier.

Kimball, R. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. John Wiley & Sons, Inc.

Linstedt, D., & Olschimke, M. (2016). Chapter 3 - the data vault 2.0 methodology. In D. Linstedt & M. Olschimke (Eds.), *Building a scalable data warehouse with data vault 2.0* (p. 33-88). Boston: Morgan Kaufmann. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128025109000039 doi: https://doi.org/10.1016/B978-0-12-802510-9.00003-9

Loshin, D. (2010). *Master data management*. Morgan Kaufmann.

Lyu, S., Craig, S., O'Reilly, G., & Taniar, D. (2025). The development and use of data warehousing in clinical settings: a scoping review. *Frontiers in Digital Health*, *Volume 7 - 2025*. Retrieved from https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2025.1599514 doi: 10.3389/fdgth.2025.1599514

Power, D. J. (2007). *A brief history of decision support systems.*

Seenivasan, D. (2022). Etl vs elt: Choosing the right approach for your data warehouse. *International Journal for Research Trends and Innovation (IJRTI)*, *7*(2), 110–122.

Sen, A., & Sinha, A. P. (2005, March). A comparison of data warehousing methodologies. *Commun. ACM*, *48*(3), 79–84. Retrieved from https://doi.org/10.1145/1047671.1047673 doi: 10.1145/1047671.1047673

Shamir, A. (1979, November). How to share a secret. *Commun. ACM*, *22*(11), 612–613. Retrieved from https://doi.org/10.1145/359168.359176 doi: 10.1145/359168.359176

Shamir, A., Rivest, R. L., & Adleman, L. M. (1981). Mental poker. In *The mathematical gardner* (pp. 37–43). Springer.

Singh, A. K. (2025, Jan). *Global data warehousing market size, share, and trends analysis report – industry overview and forecast to 2032.* Retrieved from https://www.databridgemarketresearch.com/reports/global-data-warehousing-market

Sprague Jr, R. H., & Watson, H. J. (1996). *Decision support for management.* Prentice-Hall, Inc.

Wadhwani, P. (2026). *Data warehousing market statistics - global 2025 forecasts.* Retrieved from https://www.gminsights.com/industry-analysis/data-warehousing-market

Yao, A. C. (1982). Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)* (pp. 160–164).

Yao, A. C.-C. (1986). How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (sfcs 1986)* (pp. 162–167).

Yao, Y., Zhang, K.-J., Song, T.-T., Zhang, L., & Wang, S.-N. (2023). The complete new solutions to the blind millionaires' problem in d-dimensional quantum system. *Physica A: Statistical Mechanics and its Applications*, *627*, 129138.

Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.-Z., Li, H., & an Tan, Y. (2019). Secure multi-party computation: Theory, practice and applications. *Information Sciences*, *476*, 357-372. Retrieved from https://www.sciencedirect.com/science/article/pii/S0020025518308338 doi: https://doi.org/10.1016/j.ins.2018.10.024

NB: Full anonimised transcripts of all interviews are available upon request to m.r.spruit@liacs.leidenuniv.nl