

The Evolution of Data Storage Architectures: Examining the Value of the Data Lakehouse

MASTER THESIS

Nathalie E. Janssen

Programme: MSc Business Information Technology

Track: Data Science & Business

Faculty: Electrical Engineering, Mathematics & Computer Science (EEMCS)

Student number: s1824902

Date: 29th of August 2022

GRADUATION COMMITTEE

Dr. F.A. Bukhsh

Organisation: University of Twente

Faculty: Electrical Engineering, Mathematics & Computer Science (EEMCS)

Department: Data management & Biometrics

Dr. R. Silva Souza Guizzardi

Organisation: University of Twente

Faculty: Behavioural, Management, and Social Sciences (BMS)

Department: Industrial Engineering & Business Information Systems (IEBIS)



**UNIVERSITY
OF TWENTE.**

Preface

Over the span of seven months, I have researched data storage architectures and written this thesis to complete the master's program Business Information Technology at the University of Twente.

This work therefore marks the end of my student life – upon which I look back with joy and pride. It all started with a bachelor's degree in International Business Administration. When I had the opportunity to go abroad for my minor, I started my orientation for further education. This is when my interest was drawn towards the field of technology – more specifically, data science. Soon after that, I knew that I wanted to pursue the master's degree in Business Information Technology.

After completing the pre-master, I chose the specialization of Data Science & Business. Even though I knew it was going to be a challenge, it was the best choice I could have made. The world of programming, data, and statistics has been very challenging, but it has taught me so much and I am happy to have found my passion. Moreover, I am very grateful for all the people that I have met along my journey, who always supported and motivated me. For that, I would like to thank and acknowledge the following people.

I first thank my main supervisor Faiza Bukhsh for her continuous guidance – both professionally and personally. By connecting on a weekly basis, I felt guided at every step on my path while, at the same time, a lot of autonomy was still given to challenge myself. I am also grateful for the constructive feedback from my second supervisor Renata Silva Souza Guizzardi. When I had questions or experienced setbacks, they were readily available and helped me to get back on track. It has been a very pleasant collaboration and I am thankful for all their efforts.

Besides all the support from the university, I notably appreciate my colleagues at Avanade for their welcoming hospitality and willingness to share their expertise. A special thanks go to Suvadeep Sinha and Iman Jabor who have supervised me during the internship.

Finally, I am passionately grateful for the love and encouragement I received from my boyfriend, family and friends. I feel blessed knowing that you were there for me during this journey – and that you will be there for me in the next chapter of my life.

I hope you enjoy your reading.

Nathalie Esmée Janssen

Rotterdam, 25th of August 2022

Abstract

In today's world, data has become an important resource a company can have. Given the digital shift society has been making and the continuous growth of data, companies desire to be data-driven. However, choosing a suitable storage architecture to efficiently store, process, and manage data from numerous sources remains challenging. Currently, there are three storage architecture generations of which the newest (known as data lakehouse) was introduced in 2020. Given its novelty, limited research has been done into the rationale behind its introduction, strengths, and weaknesses. In order to fill this gap, this study answers the following research question: "What is the added value of the data lakehouse architecture in your data management platform?". A systematic literature review and expert interviews were conducted to answer this question. As a result, this study presents two models 1) a data storage evolution model and 2) a fine-grained reference architecture of the data lakehouse. With the use of the produced models, findings from literature and expert interviews, this study demonstrates the added value of the data lakehouse compared to the preceding architectures. In essence, the value of the data lakehouse can be explained through 1) the combination of best practices from the data warehouse and data lake and 2) the introduction of a data management layer on the storage object. Finally, we acknowledge that this study has several limitations: 1) due to time constraints, a second round of validation interviews has not been carried out to validate the revised conceptual model and the designed reference architecture, 2) the sample size for the interviews is limited to five experts, 3) the chosen qualitative approach is prone to bias, and 4) given the novelty of the data lakehouse and the fact that it is still a work in progress, the information provided in this thesis might be incomplete due to new releases and upgrades in the near future. Concludingly, for future research, we recommend performing validation interviews to validate the conceptual model and reference architecture. Moreover, we recommend increasing the sample size to increase the validity of the results. Lastly, we recommend continuing to research storage architectures to keep the information up to date given the fast developments and new releases.

Table of Contents

List of Figures.....	6
List of Tables	7
1. Introduction.....	8
1.1. Company Information	8
1.2. Background Information	8
1.3. Scope.....	11
1.4. Problem Statement	12
1.5. Research Questions and Objectives	13
1.6. Thesis Outline	14
2. Exploration of the Data Management Platform.....	15
3. Methodology	18
3.1. Methodological Approach.....	18
3.1.1. Research Design.....	18
3.1.2. Rationale Behind Methodology Selection	19
3.1.3. Research Questions	19
3.2. Data Collection Method.....	20
3.2.1. Literature Review Approach.....	20
3.2.2. Qualitative Method: Interviews	23
3.3. Data Evaluation Method	25
3.3.1. Qualitative Research Approaches	25
3.3.2. Narrative Research Approach	27
3.4. Evaluation of Methodological Choices.....	28
4. Building the Data Storage Evolution Model.....	29
4.1. Evolution of Storage Architectures.....	29
4.2. Designing the Data Storage Evolution Model	30
4.2.1. Data Warehouses.....	31
4.2.2. Data Lakes	34
4.2.3. Data Lakehouses	38
4.3. Architectures of the Data Storage Solutions	42
4.3.1. Data Warehouse Architecture	44
4.3.2. Data Lake Architecture	46
4.3.3. Data Lakehouse Architecture.....	48
5. Evaluation.....	51
5.1. Interview Results on Data Storage Evolution Model.....	51
5.1.1. Insights Gathered on the Data warehouse Entity	52
5.1.2. Insights Gathered on the Data Lake Entity	55

5.1.3.	Insights Gathered on the Data Lakehouse Entity	58
5.2.	Redefined Data Storage Evolution Model	61
5.2.1.	Suggestions of Experts on Entity-Level.....	61
5.2.2.	Suggestions of Experts on Model-Level	73
5.2.3.	Final Version of the Data Storage Evolution Model.....	74
5.3.	Interview Results on Data Lakehouse Architecture.....	76
5.3.1.	Insights on Existing Architectures	77
5.3.2.	Additional Architectures Presented by the Experts.....	78
5.3.3.	Design of the Data Lakehouse Architecture	83
5.3.4.	Challenges and Current Shortcomings of the Data Lakehouse.....	84
5.3.5.	Future Perspectives of the Data Lakehouse	87
6.	Conclusion and Recommendations.....	89
6.1.	Research Questions	89
6.2.	Recommendations.....	92
6.3.	Threats to Validity	93
6.4.	Contribution to Science.....	94
6.5.	Contribution to Practice	94
6.6.	Limitations and Future Directions	94
7.	References.....	96
8.	Appendices.....	103
	Appendix A: Interview Questions.....	103
	Appendix B: Slides Used During Interviews.....	105
	Appendix C: Suggested Additions of Experts for Each Entity	107
	Appendix D: Suggestions of Experts on Improving the Model	110
	Appendix E: Expert's Evaluation of Existing Architectures	112
E.1.	Respondent 1	112
E.2.	Respondent 2	112
E.3.	Respondent 3	113
E.4.	Respondent 4	113
E.5.	Respondent 5	114
	Appendix F: Literature Review Protocol	115
F.1.	Literature Review Methodology	115
F.2.	Literature Review	117

List of Figures

Figure 1: Timeline with Milestones for the Emergence of Big Data and Cloud Computing.....	9
Figure 2: Data Lakehouse System Design by Armbrust et al., (2021).....	12
Figure 3: Overview of Thesis Outline.....	14
Figure 4: Generic Architecture for Data Management Platforms	15
Figure 5: Number of Existing DMPs per Domain	16
Figure 6: Engineering Cycle (Wieringa, 2014).....	18
Figure 7: Design Science Research Methodology Process Model (Peppers et al., 2007)	19
Figure 8: Relationship Between the Research Questions and the Engineering Cycle	20
Figure 9: Evolution Process of Storage Architectures	29
Figure 10: Data Storage Evolution Model	31
Figure 11: Two-tier Architecture with a Data Warehouse and Data Lake (Armbrust et al., 2021)	38
Figure 12: Evolution of Storage Architectures in Relation with Data Complexity	43
Figure 13: Data Warehouse Architectures (Al-Okaily et al., 2022; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)	45
Figure 14: Data Lake Architectures (Lavrentyeva & Sherstnev, 2022; B. Inmon et al., 2021; Ravat & Zhao, 2019)	47
Figure 15: Data Lakehouse Architectures (Armbrust et al., 2021; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)	49
Figure 16: Adjusted Data Warehouse Entity	55
Figure 17: Adjusted Data Lake Entity	58
Figure 18: Adjusted Data Lakehouse Entity	61
Figure 19: Adjusted Data Warehouse Entity 2.0	64
Figure 20: Adjusted Data Warehouse Entity 3.0	65
Figure 21: Adjusted Data Lake Entity 2.0	67
Figure 22: Adjusted Data Lake Entity 3.0	69
Figure 23: Adjusted Data Lakehouse Entity 2.0	71
Figure 24: Adjusted Data Lakehouse Entity 3.0	72
Figure 25: High-Level Model Showing the Idea Behind a Data Lakehouse	75
Figure 26: Revised Data Storage Evolution Model	76
Figure 27: Data Lakehouse Architectures (Armbrust et al., 2021; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)	77
Figure 28: Proposed Data Lakehouse Architecture by Respondent 1 (Microsoft Azure, 2021).....	79
Figure 29: Proposed Data Lakehouse Architecture by Respondent 1 (Databricks Blog, 2020).....	79
Figure 30: Proposed Data Lakehouse Architecture by Respondent 3 (Databricks, 2020).....	80
Figure 31: Proposed Data Lakehouse Architecture by Respondent 5 (Databricks Blog, 2021).....	80
Figure 32: Proposed Data Lakehouse Architecture by Respondent 3.....	81
Figure 33: Data Lakehouse Architecture from AWS (Kava and Gong, 2021).....	82
Figure 34: Proposed Data Lakehouse Architecture by Respondent 5.....	82
Figure 35: Reference Architecture for a Data Lakehouse.....	84
Figure 36: Results on the Question of Whether the Data Lakehouse Can Replace a DW or DL	87
Figure 37: Article Selection Process	116
Figure 38: Core Parts of a Data Management Platform	117
Figure 39: Process Flow with DropZone in Place	118
Figure 40: DMP Architecture Examples A (left) and B (right)	122
Figure 41: DMP Architecture Examples C (left) and D (right)	123
Figure 42: DMP Architecture Examples E (left) and F (right)	124
Figure 43: DMP Architecture Examples G (left) and H (right)	125
Figure 44: DMP Architecture Examples I (left) and J (right).....	126
Figure 45: Generic Architecture of a Data Management Platform	127

List of Tables

Table 1: Search Queries and Number of Results in Scopus for Sub-Question 1	21
Table 2: Search Queries and Number of Results in Google Scholar for Sub-Question 1	22
Table 3: Search Queries and Number of Results in Scopus for Sub-Question 2 and 3	22
Table 4: Search Queries and Number of Results in Google Scholar for Sub-Question 2 and 3	22
Table 5: Search Queries and Number of Results in Scopus for Sub-Question 4 and 5	22
Table 6: Search Queries and Number of Results in Google Scholar for Sub-Question 4 and 5	23
Table 7: Comparison of the Five Qualitative Approaches (Creswell, 2007)	28
Table 8: Overview of Assumptions Describing Advantages of Data Warehouses	35
Table 9: Overview of Assumptions Describing Disadvantages of Data Warehouses	35
Table 10: Overview of Assumptions Describing Advantages of Data Lakes	37
Table 11: Overview of Assumptions Describing Disadvantages of Data Lakes	39
Table 12: Overview of Assumptions Describing Advantages of Data Lakehouses	43
Table 13: Overview of Assumptions Describing Disadvantages of Data Lakehouses	44
Table 14: High-level Comparison Between the Three Storage Solutions	45
Table 15: Overview of Interview Respondents Profile	53
Table 16: Overview of Viewpoints of Each Respondent for Every Assumption	54
Table 17: Overview of Adjustments of Assumptions Related to Data Warehouses	57
Table 18: Overview of Adjustments of Assumptions Related to Data Lakes	60
Table 19: Overview of Adjustments of Assumptions Related to Data Lakehouses	63
Table 20: Suggested Advantages for Data Warehouses	64
Table 21: Suggested Disadvantages for Data Warehouses	66
Table 22: Suggested Advantages for Data Lakes	68
Table 23: Suggested Disadvantages for Data Lakes	70
Table 24: Suggested Advantages for Data Lakehouses	72
Table 25: Suggested Disadvantages for Data Lakehouses	74
Table 26: Overview of the Overarching Themes of Suggestions for Improving the Model	75
Table 27: Overview of Challenges and Shortcomings of Data Lakehouses	86
Table 28: Overview of the Evaluation of the Assumptions.....	91
Table 29: Big Data Storage Solutions in Microsoft Azure	109
Table 30: Existing Solutions Identified in Literature per Domain	117
Table 31: Overview of Challenges Identified with Data Management Platforms	122
Table 32: Suggested Additions of Experts for Each Entity	130
Table 33: Suggestions of Experts on Improving the Model	132
Table 34: Respondent 1 – Evaluation of Existing Architectures	133
Table 35: Respondent 2 – Evaluation of Existing Architectures	133
Table 36: Respondent 3 – Evaluation of Existing Architectures	134
Table 37: Respondent 4 – Evaluation of Existing Architectures	134
Table 38: Respondent 5 – Evaluation of Existing Architectures	135

1. Introduction

This chapter provides information on the collaborative company, background information to understand the scope and context of the problem statement. Finally, an overview of the research objectives and a thesis outline is provided.

1.1. Company Information

This research was conducted for Avanade, an IT consulting company founded in April 2000 by Accenture and Microsoft. Over the years, this joint venture has employed 56,000 professionals in 26 countries, with its headquarters in Seattle. In the Netherlands, around 450 professionals are working at Avanade, with two offices in Amsterdam and Heerlen. Using the Microsoft ecosystem, Avanade has served over 4000 clients worldwide by delivering innovative services and solutions using the Microsoft platform (Avanade Inc., 2022).

Within Avanade, there are 12 different Talent Communities (i.e., departments), and this research was conducted at the Analytics Talent Community. This Talent Community consists of 4 sub-fields, including advanced analytics, analytics experience, data engineering, and analytics architecture. This research was performed specifically for the advanced analytics sub-community, which aims to extract value and insight from data for their clients successfully.

1.2. Background Information

Throughout the years, information has become one of the most important resources a company can have. This includes information on their customer, processes, products, competitors, and anything related to the field in which they operate. Since our society has been making a digital shift, more and more data sources have become available that serve as “oil” to the company. As the years progressed, the rate at which data was generated accelerated, giving rise to the term ‘big data’.

Different explanations, from 3V's, to 4V's, and more V's have been provided to define big data. Doug Laney is the first to describe big data with the 3V's that stands for Volume, Variety, and Velocity. Volume is related to the size of the data set, variety is all about the different types of data formats, and velocity is the speed at which the data comes in and goes out (Philip Chen & Zhang, 2014). Later, IBM and Microsoft included another V which stands for veracity and touches upon the trustworthiness of the data. Also, McKinsey & Co added ‘Value’ an additional V and wanted to highlight the worth of the data, which can be referred to as the hidden insights in big data (Yaqoob et al., 2016). Hence, it can be observed that there are multiple variations on the V's to describe big data. Cartledge (2016) collected all the V's that were introduced in chronological order. His study presented a total of 19 V's that were found in the literature from 2001 until 2015. However, from a practical point of view, if the data does not fit on one machine, or it takes a very long time for the machine to return an answer when any operation is carried out on the data, you are dealing with big data.

Regardless of the number of existing V's, the essence of big data remains in the 3V's volume, variety, and velocity. With each of these characteristics, organizations face numerous challenges. Challenges that are related to the volume of the data are, for instance, 1) the way data should be stored, 2) how the stored data is still easily accessible when needed, and 3) how analyses can be performed in an efficient way (Philip Chen & Zhang, 2014). According to Lu et al. (2016), dealing with the variety of data in the current database ecosystems is the most challenging issue. The data can now be presented in a structured, semi-structured, or unstructured format. As a result, the need to deal with all these different types of data grew, so additional data management techniques started to emerge. This led to emerging techniques and technologies to deal with big data, and one of the most robust techniques is the use of cloud computing.

The emergence of both Big Data and Cloud Computing have some significant milestones, which are summarized in Figure 1. The events on top represent essential developments in the era of big data, and the events below represent important events in the emergence of cloud computing.

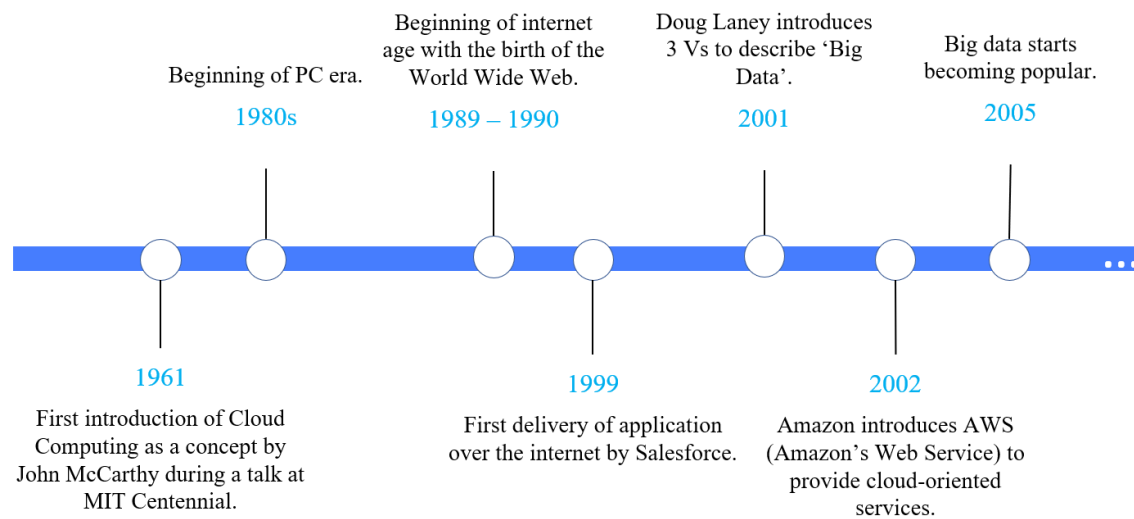


Figure 1: Timeline with Milestones for the Emergence of Big Data and Cloud Computing

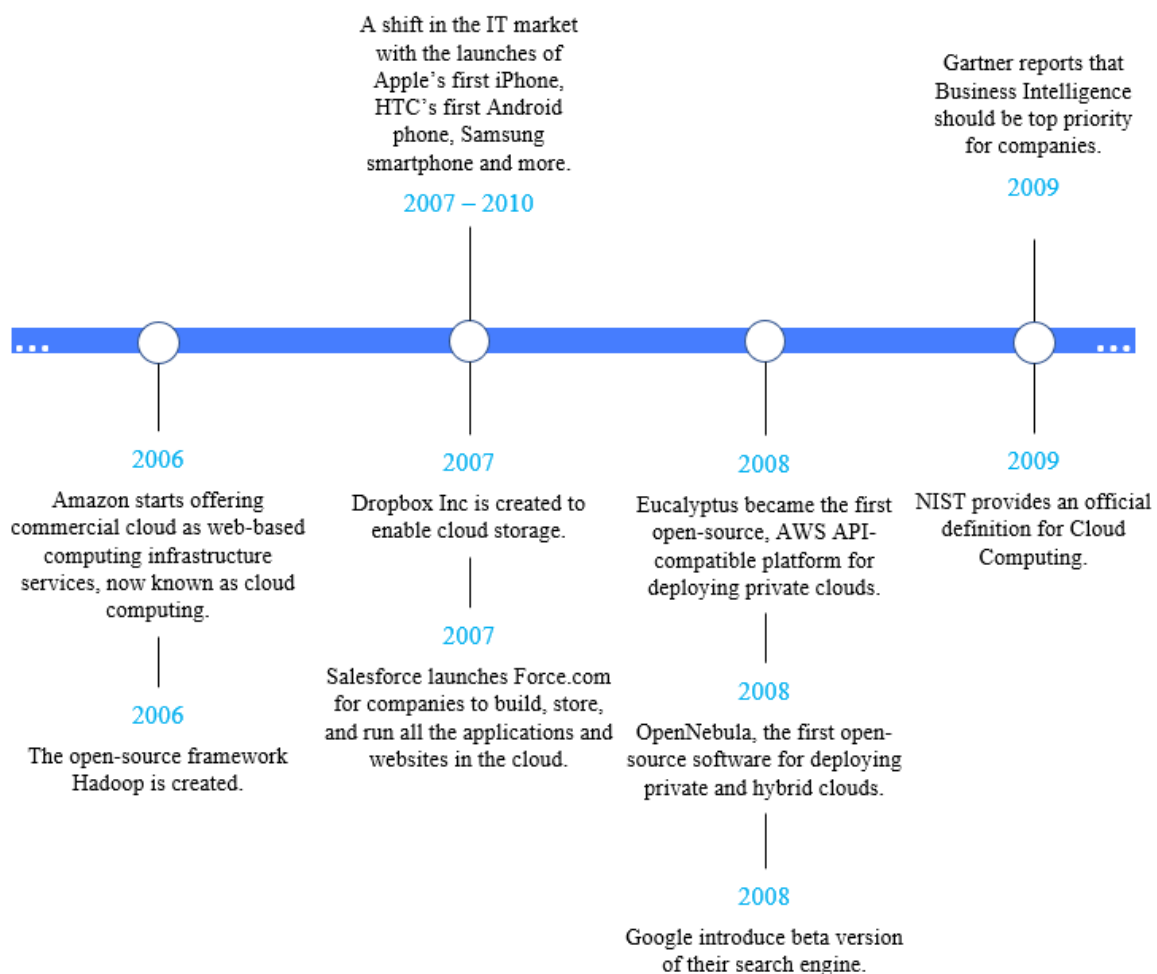


Figure 1: Timeline with Milestones for the Emergence of Big Data and Cloud Computing

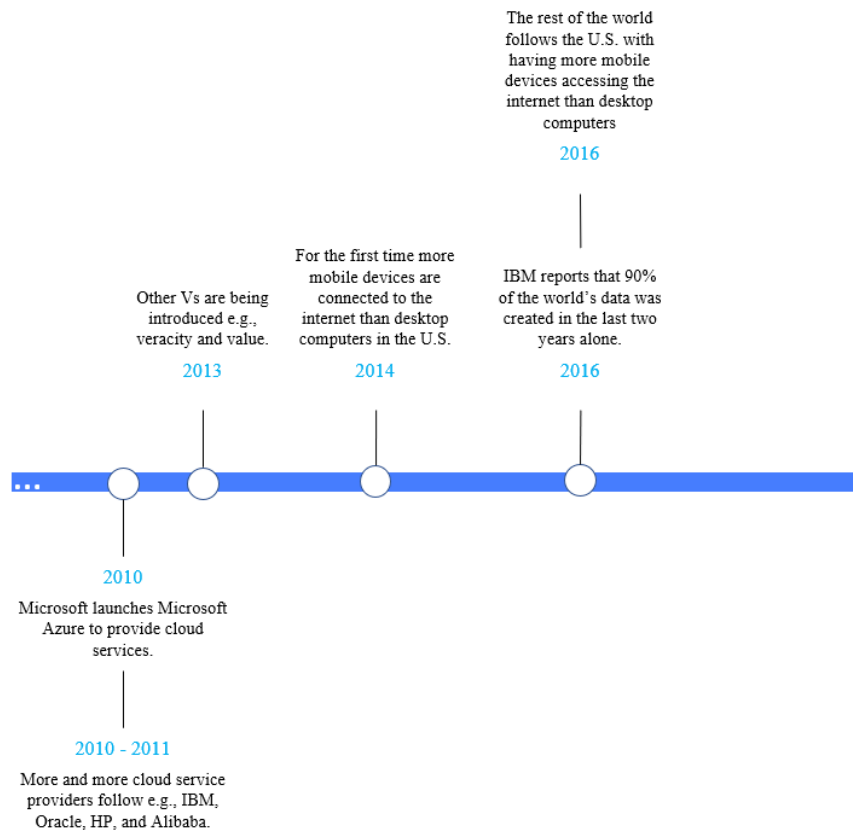


Figure 1: Timeline with Milestones for the Emergence of Big Data and Cloud Computing

Cloud computing has been one of the fast-emerging technologies revolutionizing IT infrastructures and their flexibility. Qian et al. (2009) examined the history of Cloud Computing. They found that the first time someone tried to describe the underlying concepts of cloud computing happened during an MIT Centennial talk in 1961 by John McCarthy. However, this was just a shallow reference to the idea of cloud computing. More than 30 years later, in 1999, the first milestone in cloud computing was set by Salesforce.com, which started to deliver enterprise applications over the internet, which gave rise to Cloud Computing. The subsequent development was in 2002 when Amazon began to provide cloud-oriented services through Amazon's Web Service (AWS). After that, more and more companies became cloud service providers. For instance, Microsoft launched Microsoft Azure in 2010, followed by IBM, Oracle, HP, and Alibaba. (Singh, 2021; W3Schools, 2022) In literature, the concept was officially introduced by Eric Schmidt in 2006 in literature on search engine strategies (Qian et al., 2009). After that, numerous definitions were formulated for this new technology. For instance, L. Wang & von Laszewski (2008) proposed the following: "computing cloud is a set of network enabled services, providing scalable, Quality of Service (QoS) guaranteed, normally personalized inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way". Qian et al. (2009) defined Cloud Computing as "a kind of computing technique where IT services are provided by massive low-cost computing units connected by IP networks." Moreover, they state that there are "5 major technical characteristics of cloud computing: 1) large scale computing resources, 2) high scalability, 3) shared resource pool, 4) dynamic resource scheduling, and 5) general purpose." Next to different studies that came up with definitions, a widely cited U.S. government entity, the National Institute of Standards & Technology (NIST), defined 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. This cloud model is composed of five essential characteristics, three service models, and four deployment models (Mell & Grance, 2011)." Admitting that the last definition is seen as the actual definition in literature, this study will take the definition provided by Microsoft which is as follows: "cloud computing is the delivery of computing services –

including servers, storage, databases, networking, software, analytics, and intelligence – over the internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scales (Microsoft Azure, 2022).” This is a lightweight definition of the available services and what is meant by ‘the cloud’, and they touch upon the business value of cloud computing. Hence, this is seen as a complete and easy-to-understand definition.

There are many possibilities that can be offered by cloud computing, one of which is a Data Management Platform (DMP) solution. A data management platform is a central hub where data from multiple sources are stored and managed. The data is formatted through a data pipeline to be used for any analytical purpose such as making predictions, detecting trends, gaining a deeper understanding of the customers, performing analyses and creating reports, or publishing critical insights on a dashboard. Due to the ‘pay-as-you-go’ pricing model, flexibility, high security, and endless choice of computing services, it is beneficial to host the data management platform in the cloud.

1.3. Scope

Allegedly, a data management platform tends to be a desired technology by any company dealing with data, given the opportunities it brings. However, evaluating the exact value of such a platform is not that simple due to the influence of numerous factors. Since it is quite an investment to implement a data management platform, the first literature review performed for Research Topics aims to examine the actual value of a data management platform. This review focussed on how a data management platform is constructed, for what purposes it is used in different domains, and existing solutions in the literature. One of the findings was identifying the data management platform's core parts and architectural layers. As a result, one core part was selected as the main subject for the thesis: the data storage solution. Since there are multiple possibilities in choosing and configuring a storage object, it is challenging to determine which storage solution is the best design for your data management platform.

Over the last few decades, it can be observed that three generations of storage architectures have evolved. The first is the Data Warehouse, which is now seen as a traditional storage solution where data from multiple sources are stored together in a unified data repository. This repository is seen as “the only data truth” and provides data for analytical and reporting purposes. Besides the benefits of this architecture, there are several disadvantages. One of which is that a data warehouse lacks flexibility. Data warehouses handle structured data very well, although it struggles with semi-structured and unstructured data. Moreover, the implementation, maintenance, and scaling costs are very high. Consequently, a new architecture was developed for storage solutions that deal with these disadvantages.

The second generation, one of the popular solutions nowadays, is the implementation of a data lake. A data lake can be defined as “a methodology enabled by a massive data repository based on low-cost technologies that improve the capture, refinement, archival, and exploration of raw data within an enterprise. A data lake contains the mess of raw unstructured or multi-structured data that for the most part has unrecognized value for the firm (Fang, 2015).” In other words, it is a centralized repository that enables ingesting, storing, and processing data in any format. Next to being able to support any format, data lakes are flexible, durable, and cost-effective. Nevertheless, this storage solution also has its downsides. For instance, it does not support data management functionalities, lacks the support for ACID transactions, has the risk of keeping corrupt data in your data lake since there is no quality control, and there is no possibility for versioning and time travel.

This gave rise to the third generation of data storage architectures: the Data Lakehouse. This architecture combines the best practices of a data warehouse and a data lake. In doing so, the data lakehouse will tackle the limitations of the previous generations of storage architectures. Even though this architecture is a promising solution, this technology is still very new to the market. Hence, it has not been widely adopted yet by many companies. Reasons for this are, for instance, a lack of trust in the potential of this architecture and the lack of expertise on how to implement it.

The architecture of a Data Lakehouse is recently introduced, and there is not much literature on this topic. Armbrust et al., (2021) proposed a Data Lakehouse system design, as shown in Figure 2.

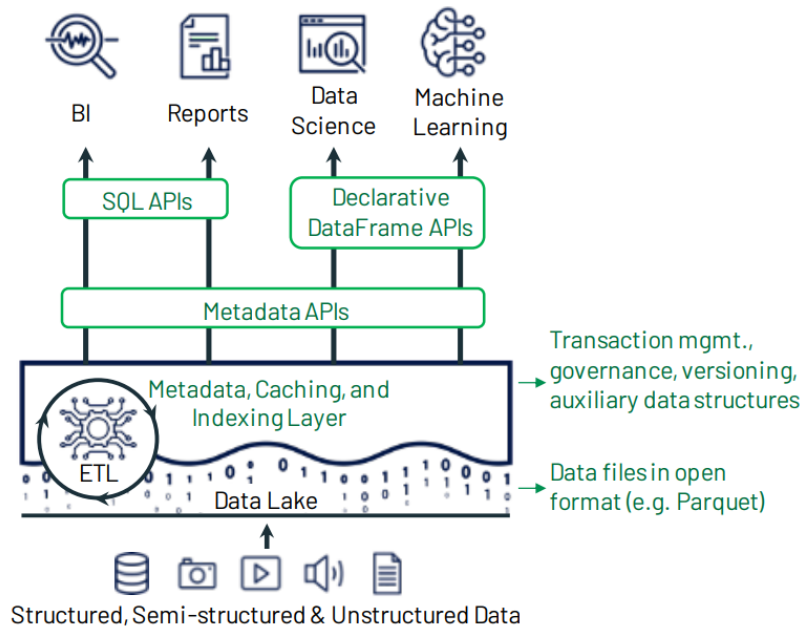


Figure 2: Data Lakehouse System Design by Armbrust et al., (2021)

The design shows that the data is stored in a data lake to support the storage of all types of data. On top of that, a transactional metadata layer is built. This layer is also referred to as a computing layer where the data warehouse capabilities are enforced (Lavrentyeva & Sherstnev, 2022). Here, management features such as metadata management, caching, indexing, schema enforcement, data layout optimizations, and ACID transactions (atomicity, consistency, reliability, and durability) are supported. Next, an API layer is in place to allow fast direct access to the data sets. Lastly, a serving layer is on top to support BI, reporting, data science and machine learning use cases. As a result, the system design addresses data warehouse constraints, including a lack of flexibility and support for different data formats and scaling costs. It also deals with the data lake challenges, including enforcing data security and robust governance policies, the lack of data consistency and ACID transactions, data quality, and concurrency problems.

1.4. Problem Statement

Provided the background information and context of the scope, the following arguments form the problem statement of this research. First of all, the evolution of storage architectures consists of three generations. However, in literature and practice, no model explains the rationale behind the evolution process. Without this knowledge, it is very challenging to understand the reason for the introduction of the newest architecture. Secondly, in the literature, no generic reference architecture has been presented. Whereas in practice, different high-level architectures are published by the data lakehouse vendor. However, according to the knowledge of practitioners at Avanade, there is no generic fine-grained architecture available that explains the data lakehouse concept. Thirdly, given the novelty of the newest storage architecture, a limited amount of research is available on this topic. In fact, the search term ‘data lakehouse’ on Scopus resulted in three papers published in 2021. Two academic articles and books were found using the same search term for the Google Scholar database.

All the problems mentioned above contribute to a knowledge gap in literature and practice regarding understanding the added value of the data lakehouse to the evolution of storage architectures. This research aims to close this knowledge gap by developing a data storage architecture evolution model and presenting a generic fine-grained reference architecture. This will be achieved by performing an in-depth literature review, producing an evolution model and a reference architecture, and performing

validation interviews. As a result, this research will contribute to understanding the added value of the data lakehouse to the evolution process and a data management platform.

1.5. Research Questions and Objectives

The objective of this research is two-fold. First of all, we want to create a comprehensive understanding of what a data management platform (DMP) entails, what the architecture is composed of, and its strengths and weaknesses. This will be done by examining platforms in different industries to grasp the purpose of a DMP in various domains. The following research question will guide this part of the research:

RQ1: “What is the added value of data management platform solutions?”

To be able to answer the main research question, the following sub-questions have been formulated:

SQ1.1. What are the core parts of a DMP?

SQ1.2. How is a generic architecture for a DMP constructed?

SQ1.3. Which solutions exist in current literature, and for what purpose are they used?

SQ1.4. What are the current solutions' comparative strengths and weaknesses?”

SQ1.5. What challenges are faced when building a DMP?

Secondly, this research will focus on the storage solution, one of the core parts of a DMP. As mentioned in Section 1.2., there is an emerging storage architecture: the data lakehouse. Given its novelty, several problems were identified and summarized in the problem statement (Section 1.3). Therefore, this research aims to close the identified knowledge gap and contribute to literature and practice by answering the following research question:

RQ2: “What is the added value of a data lakehouse architecture in your data management platform?”

In order to reach a well-grounded answer, the following sub-questions have been formulated:

SQ2.1. What are the different architectures for data storage solutions?

SQ2.2. What are the strengths of Data Warehouses, Data Lakes, and Data Lakehouses?

SQ2.3. What are the weaknesses of Data Warehouses, Data Lakes, and Data Lakehouses?

SQ2.4. How can the evolution of the storage architectures and the value of each architecture be explained in a conceptual model?

SQ2.5. How can the designed artefact be improved?

SQ2.6. What challenges are faced when utilizing the lakehouse architecture?

SQ2.7. What are the future perspectives concerning the data lakehouse?

SQ2.8. How should a fine-grained architecture for the data lakehouse be constructed to explain its way of working and value?

Concludingly, after answering the sub-questions, the main research objectives of this thesis are as follows:

Research objective 1: Examine the added value of the data lakehouse storage solution

Research objective 2: Develop a conceptual model showing the evolution and value of each storage architecture

Research objective 2: Present a fine-grained conceptual architecture for the data lakehouse

By answering these research questions and sub-questions, this study will contribute both to literature and practice through the following contributions: we will 1) examine the added value of a data

management platform, 2) provide a comparative study between data warehouses, data lakes, and data lakehouses, 3) present a validated model that explains the evolution of storage architectures, 4) present a fine-grained reference architecture that clearly explains the lakehouse concept. These contributions aim to close the current knowledge gap in literature and practice.

1.6. Thesis Outline

The remainder of the thesis is structured as follows. First, we will explain which methodologies have been used for this research. This is followed by extensive literature research, guided by the research- and sub-questions we have formulated. Next, the results are presented in Chapter 4 and will be discussed in Chapter 5. Finally, a chapter is dedicated to sharing our conclusions by answering the research questions, evaluating our contributions and limitations, and providing recommendations for future research and the company. An overview of the chapters and the related research question and sub-questions are presented in Figure 3.

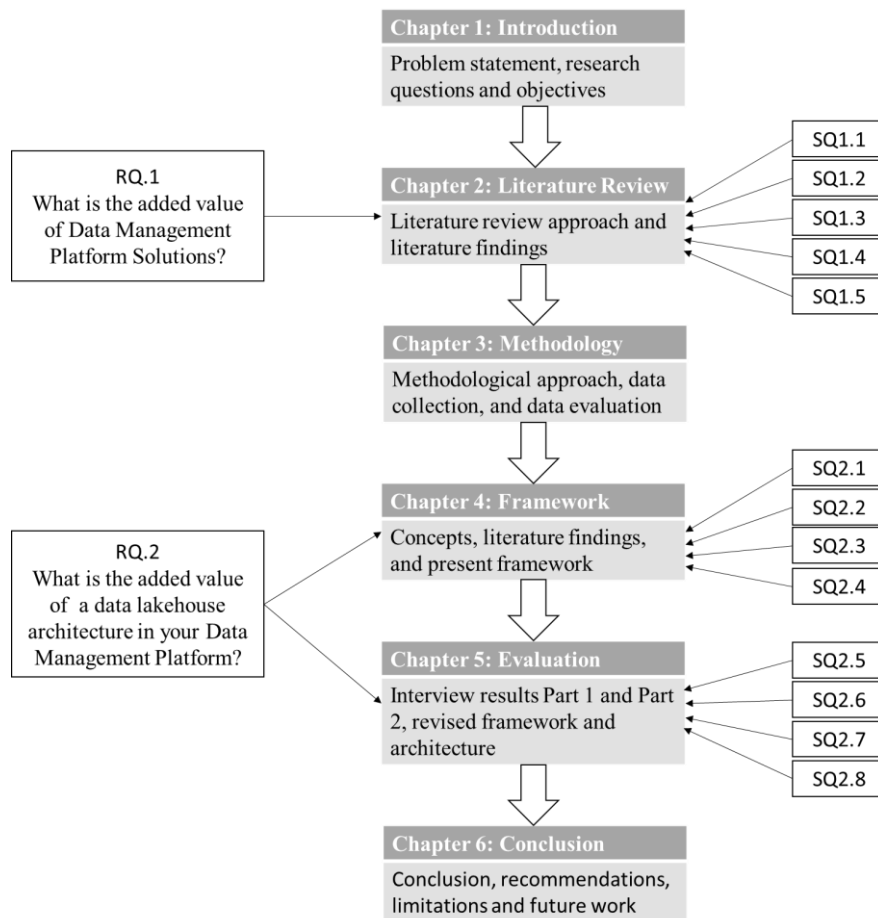


Figure 3: Overview of Thesis Outline

2. Exploration of the Data Management Platform

For this study, two distinct literature reviews were conducted. The first aimed to generate a solid base of information regarding data management platforms and serve as a base for the thesis. After that, the second literature review was entirely focused on data storage architectures. This chapter will provide a brief description of the insights gathered during the literature review with regard to data management platforms.

To perform a systematic literature review, the approach used by Bukhsh et al., (2020) and Wienen et al., (2017) has been used as inspiration for this literature review. An elaborate explanation of how this approach was constructed can be found in Appendix A.1. After selecting an appropriate set of academic papers, this set of papers was analysed to provide an answer to RQ1 and sub-questions 1.1. to 1.5. The extensive literature review is provided in Appendix A.2. The following sections will concisely give a concluding answer to each research question.

SQ1.1. “What are the core parts of a data management platform?”

The data management platform consists of 5 core parts: the data ingestion module, the data storage module, the data processing module, a data visualization/sharing module, and a data governance module. The literature review scope focused on the first three core modules since these are the three main functions and, therefore, always implemented in a data management platform. First, data from different sources are brought into the platform through pipelines. Then the data is centrally stored and can be used for analytical purposes through the tools implemented in the processing module. Hence, the main functions of a data management platform are to ingest and pull data, store data, and provide capabilities to process the data so that it becomes valuable for business stakeholders.

SQ1.2. “How is a generic architecture of a data management platform constructed?”

Ten different architectures were examined and compared to observe their similarities and differences to answer this question. There were two types of differences: 1) some layers had the same purpose, but they were given a different name, and 2) some architectures had included a layer that was excluded by others. However, we found that the average number of layers was 4 to capture all the necessary functionalities. Concludingly, we presented a general architecture that consisted of 4 horizontal layers and one vertical layer (Figure 4). The horizontal layers are a data storage & collection layer, a data processing layer, a data analysis layer, and a data visualization layer. Finally, the vertical layer is a security layer since adhering to security standards should be considered in every horizontal layer.

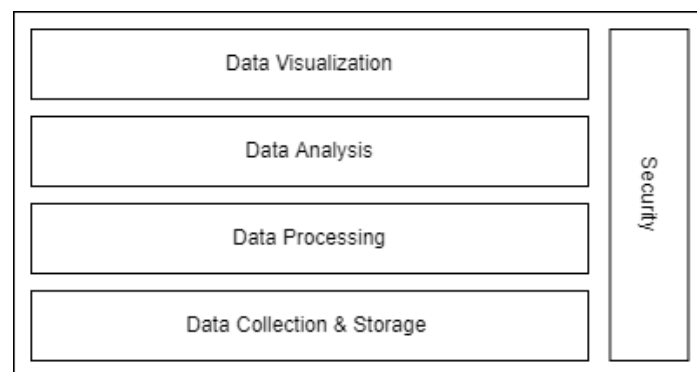


Figure 4: Generic Architecture for Data Management Platforms

SQ1.3. “Which solutions exist in current literature, and for what purpose are they used?”

The answer to this question is that there exist an endless number of data management platform solutions in the literature. This study limited itself to looking into a total of 35 existing platforms. Moreover, we tried to capture different domains to investigate for what purpose a specific data management platform

was built. Eventually, we found multiple solutions for the following domains: smart cities, health care, transportation, the internet of things, and marketing (Figure 5). However, we believe countless solutions in other domains could be investigated in future research.

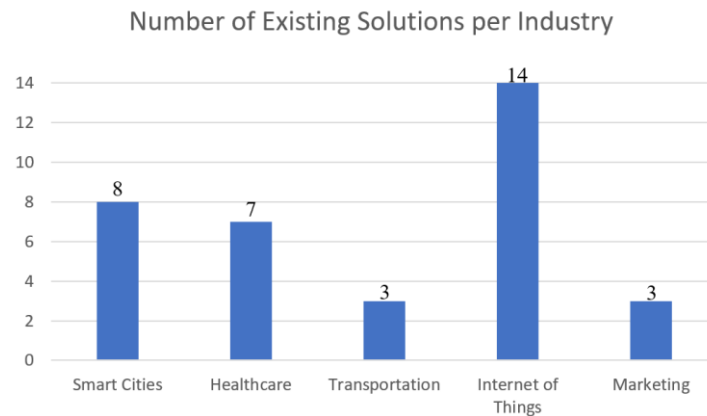


Figure 5: Number of Existing DMPs per Domain

SQ1.4. “What are the current solutions' comparative strengths and weaknesses?”

The strength that all the examined platforms had in common was their ability to handle the ingestion process of data from numerous sources. For two solutions, this included only homogeneous data. However, for all 33 other solutions, the platform supported ingesting heterogeneous data. Another strength of a data management platform is its capability to enrich data by combining data from different sources, which results in more meaningful data. Thirdly, a common strength of the data management platform is that it generates all the necessary information to answer specific business questions. Business problems such as becoming more efficient, discovering and predicting new trends, and creating analysis reports on the business' customers or performance are all answered with the support of a data management platform.

There were also multiple weaknesses identified in the analysed platforms across different domains. Some of them were use-case specific, however, a few weaknesses occurred in more than one platform. For instance, three studies indicated that their presented platform did not have the capabilities to provide in-depth analyses. This was either because the platform could not store heterogeneous data, or the platform could not target the behaviour of individuals but only of segmented groups. For some platforms, there were also some technical weaknesses. One argued that their platform was capable of ingesting and storing heterogeneous data. However, it was not competent enough to process and manage the different data types in a unified and scalable manner. Two solutions could not meet the low-latency requirement for ingesting, processing, and storing data efficiently and timely in the system. One argued that their level of analysis was negatively affected due to the longer processing times. Lastly, a common weakness was the transfer of different data types due to the fact that the data type changes regularly.

All in all, the identified strengths applied to almost all the examined data management platforms. Whereas the identified weaknesses were often specific to a particular data management platform.

SQ1.5. “What challenges are faced when building a data management platform?”

Multiple challenges were identified for the design of a DMP. A few challenges are faced with regard to the data. The chances are high that datasets are either incomplete or need to be complemented with other datasets to become valuable. Hence, the challenge is to design a flexible platform capable of dealing with imperfect data. The flexibility of the platform is also crucial for another challenge. Managing heterogeneous data requires a flexible approach to storing and linking data semantically in a

uniform way. This should therefore be taken into account when designing a DMP. Another challenge for platforms that want to support accessing real-time data is keeping their performance high. Since a platform can be overloaded with many activities, the design should be scalable and robust to deal with all the traffic. Lastly, the challenge of a DMP is to migrate different databases within a company. A DMP is implemented because it is desired to have all the data that is available to your company stored centrally in one place. However, sometimes different databases are not compatible with one another for transferring data. A DMP is responsible for being capable of dealing with these challenging migration issues.

RQ1. “What is the added value of a data management platform?”

After extensive literature research, we conclude that a data management platform takes on all the capabilities necessary for the entire data management life cycle. Its fine-grained architecture ensures that all the necessary capabilities for managing data are included in the platform. Moreover, it solves many problems concerning ingesting, storing, processing and managing heterogeneous data from numerous sources into one central repository. Hence, having a data management platform in place is highly valuable when dealing with numerous data sources because it enables the company to manage all those data sources from a central spot. The central management of data contributes to more efficient and effective business processes because searching for relevant data and obtaining valuable information from the data at your disposal is now more convenient. This increase in efficiency might lead to increased business revenue in monetary and knowledge terms. Concludingly, a data management platform is of added value to a company when it deals with numerous data sources and wants to manage and utilize the data effectively and efficiently.

3. Methodology

This chapter explains the methodology used to reach the research objectives and answer the research questions defined in Section 1.5. This chapter is structured according to 4 sub-sections. The first section describes the general methodological approach that was taken. This is followed by explaining which data collection method was conducted and how the collected data has been evaluated. Finally, an evaluation is given where possible limitations are considered, and arguments for particular methodological choices are given.

3.1. Methodological Approach

This research was focused on gathering insights from literature and practice on the evolution of the storage architectures, the reason for the latest one to be introduced, and the potential of the newest architecture. Given that the aim was to gain in-depth insight into specific concepts, we have conducted literature research and a qualitative method.

3.1.1. Research Design

For this research, two design science methods were considered for the development of an artefact. One design science method that was considered was developed by Wieringa (2014). He created the engineering cycle consisting of the following phases (see Figure 6). First, the problem investigation phase is where the problem/phenomena, stakeholders, and goals are determined. The second phase is the treatment design, where an artefact or more is developed to treat the problem that was identified in the first phase. After designing an artefact, treatment validation takes place. Here, the goal is to examine whether the designed artefact will solve the identified problem. When the results of the validation process are positive, the next phase can take place, which is the treatment implementation phase. In this phase, the problem is treated with the artefact. When this is done, the implementation evaluation takes place, where you evaluate the success of the implementation. This may lead to another iteration through the engineering cycle. (Wieringa, 2014)

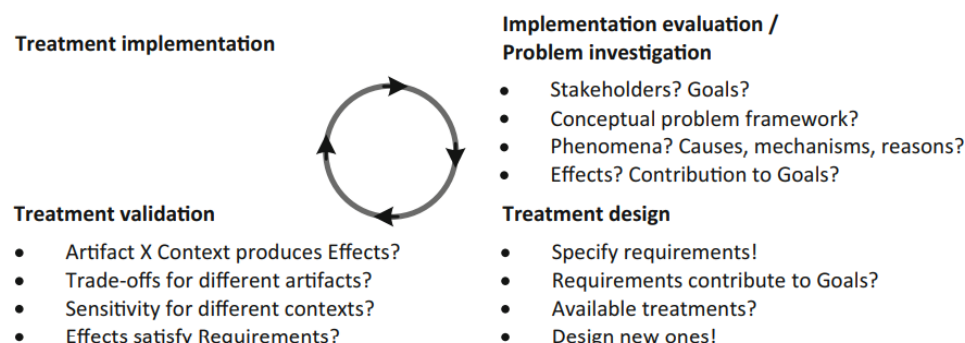


Figure 6: Engineering Cycle (Wieringa, 2014)

The second design science method that was considered was the design science research methodology (Figure 7) developed by Peffers et al., (2007). This study presented a process iteration model that consists of six phases, and the following observations were made when comparing these two design methods.

First of all, Peffers' approach consists of 2 additional phases compared to Wieringa's cycle: the defining objectives of the solution phase and the communication phase. Besides that, it can be observed that the order of the two design methods is not identical. If we were to put the phases in our own words, Wieringa's process is as follows: problem identification → design artefact → validation artefact → implementation artefact. Whereas Peffers' process is as follows: identify the problem → design artefact → implementation artefact → evaluation artefact. We observe that Wieringa (2014) starts with validating the artefact when it is designed, after which it will be implemented. In contrast, Peffers (2007) demonstrates the artefact by implementing it in some way when it is designed, after which the performance will be evaluated. Another observation we made is that Peffer's process model also implies

a cycle by having arrows pointed from evaluation and communication back to defining objectives. However, we find it strange that the arrow does not go back to problem identification, just as in the cycle of Wieringa. There is always a possibility that the problem statement was not sharp enough. This results in the development of an artefact that is not solving a problem or perhaps not the problem intended to be solved. The last observation is related to the box with ‘Possible Research Entry Points’ in Peffer’s model. At first, it was somewhat unclear what this layer was intended for, but it became clear after carefully reading the model’s description. The process model is structured in a nominal sequential order. However, this does not mean that this is the only way to follow this model. Peffers et al., (2007) explain that researchers may start at almost any step in the model and move forward from there. They would, for instance, start at phase 2 (define the objectives of a solution phase) because the motivation for their research is to develop an objective-oriented solution. Whereas a design- and development-centred approach would start with activity 3 (Peffers et al., 2007). Hence, this layer indicates at which phase a researcher should start, based on the type of solution or approach that fits best with their case.

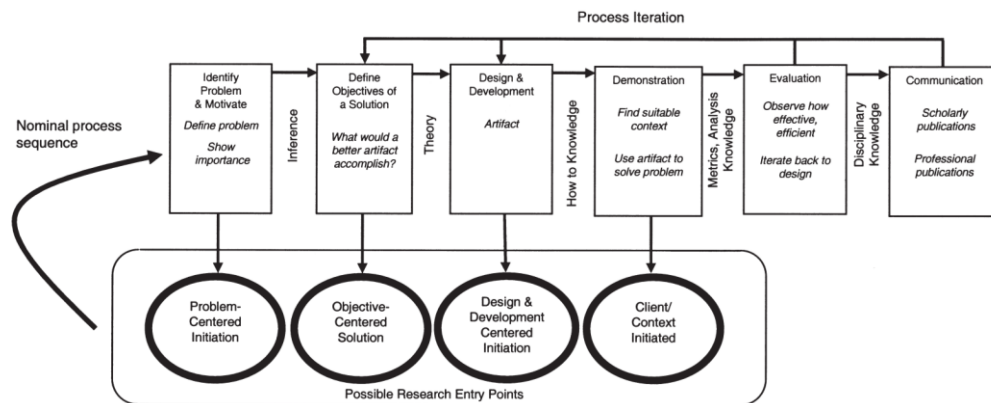


Figure 7: Design Science Research Methodology Process Model (Peffers et al., 2007)

3.1.2. Rationale Behind Methodology Selection

After carefully considering our observations, we have chosen to follow the engineering cycle developed by Wieringa (2017) to build our conceptual model. The main reason is that we believe his order of process is more logical and suitable for this research. Given that it is a conceptual model to bring knowledge to the audience, we find it more convenient to design the model and validate it before implementation instead of demonstrating the artefact immediately after designing it and evaluating its performance. We feel that Peffer’s model is missing a phase in which the design is validated. The engineering cycle is also preferred due to the possible questions in the model. This aids the researcher in structuring their way of working during each phase. Therefore, the methodological approach this research has taken is that first, the necessary knowledge through literature research is gathered to do the problem investigation. Secondly, an artefact is designed when the essential knowledge has been collected. Finally, experts are interviewed to validate the artefact in the treatment validation phase. Due to time constraints, this research does not implement the designed artefact. Hence, the first three phases of the engineering cycle are covered in this research.

3.1.3. Research Questions

In order to guide this study, a research question and a set of sub-questions were formulated in Section 1.5. Q2.1 creates an understanding of the evolution of storage architectures. This is essential for understanding the rationale behind the introduction of the newest (i.e., data lakehouse). After knowing the evolution of storage architectures, Q2.2 and Q2.3 focus on the strengths and weaknesses of each architecture. These three sub-questions will be answered through literature research and are part of the problem investigation phase of the engineering cycle.

Based on the findings of the preceding questions, an artefact will be produced, and Q2.4 examines how the artefact should be modelled. This sub-question is related to the treatment design phase of the engineering cycle.

With the guidance of Q2.5, the artefact will be validated and improved with the support of interviews. Finally, this research examines the potential worth of the newest architecture. Hence, the last three sub-questions are solely focused on the data lakehouse. Specifically, the challenges and future perspectives regarding a data lakehouse will be examined. Furthermore, the insights derived from the findings here will serve as input for developing a generic fine-grained reference architecture. The design of this architecture is a result of the findings obtained in the interviews. Still, it is also part of a new iteration of the engineering cycle and, therefore, part of the design treatment.

Finally, answering all eight sub-questions will provide an answer to research question 2 and reach the defined research objectives. Figure 8 provides an overview of which question is relevant for which phase of the engineering cycle.

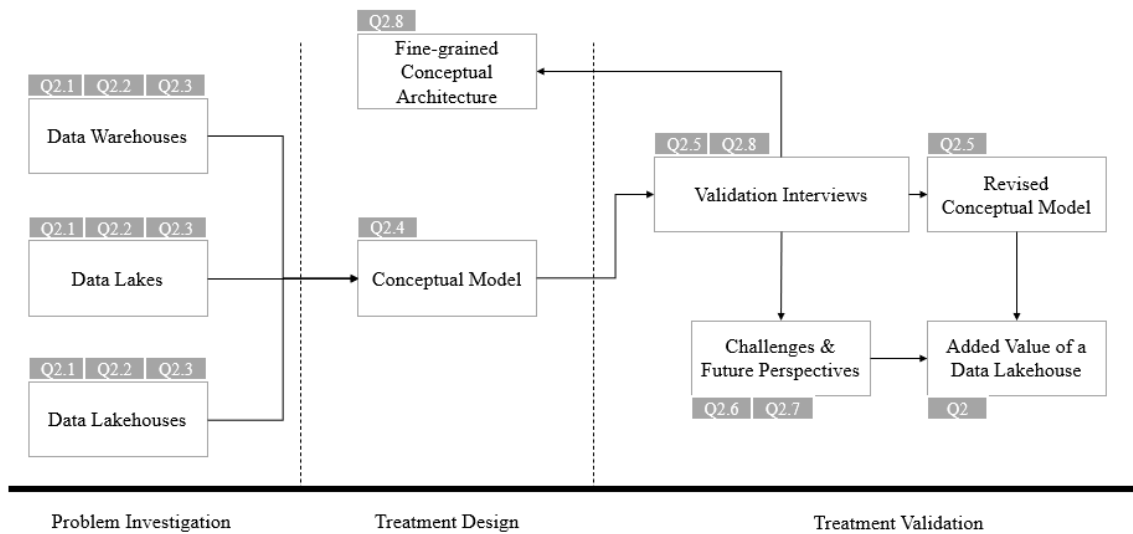


Figure 8: Relationship Between the Research Questions and the Engineering Cycle

3.2. Data Collection Method

The data collection method is two-fold: first of all, a narrow-focused literature review was performed to build an artefact. Second of all, interviews were conducted in order to validate the artefact and gain new insights based on the practical experience of the interviewees.

3.2.1. Literature Review Approach

3.2.1.1. Search Strategy

Another narrow-focused literature review was performed to answer the sub-questions and reach the research objectives. First, a database selection was made to gather relevant resources for this study. Then, a set of search queries was determined to guide our search and seek pertinent sources that would aid in answering the research questions. In order to decide on which materials were included in this study, a set of inclusion and exclusion criteria were used.

3.2.1.2. Database Selection

We have chosen to perform the literature research with the use of two databases: Scopus and Google Scholar. Scopus is known to be the most comprehensive and user-friendly database. Moreover, it relies on a set of source selection criteria and covers a variety of articles from journals and conference proceedings. Whereas Google Scholar is known to be very inclusive and covers every published

material. The downside of this can be that the quality and reliability are not that high. However, inclusivity is also a strength since more resources are included. Therefore, the Scopus database is chosen as our primary database. When it is assumed that search results from Google Scholar can complement the findings from Scopus, Google Scholar are utilized. With the use of inclusion and exclusion criteria, the most relevant sources can then be included in addition to the resources that were obtained from Scopus.

Next to using these two academic databases, the choice has been made to also use the Google search engine for collecting relevant materials. The reason is that one of the topics, the lakehouse architecture, is very new, and hence not much literature can be found. Therefore, the Google search engine has been utilized to broaden our view.

The search queries that were formulated to answer sub-question one are presented in Table 1. The queries presented here were used for the Scopus database, and the number of papers that resulted from these queries only represents the results from the Scopus database. In Table 2, the queries that were used in Google Scholar are presented with the number of articles that were returned.

Search Query	Search Query Results
("data warehouse") AND ("data lake") AND (lakehouse OR "data lakehouse")	2
("data warehouse architecture")	194
("data lake architecture")	36
("data lakehouse architecture" OR "lakehouse architecture")	2

Table 1: Search Queries and Number of Results in Scopus for Sub-Question 1

Search Query	Search Query Results
Data warehouse data lake data lakehouse	438
Data warehouse architecture	199.000
Data lake architecture	979.000
Data lakehouse architecture	1080
Lakehouse architecture	1530

Table 2: Search Queries and Number of Results in Google Scholar for Sub-Question 1

In order to be able to answer sub-questions 2 and 3, numerous queries were formulated. Table 3 presents the queries used for the Scopus database, and the number of papers that resulted from these search terms is shown in the second column. In Table 4, the queries used in Google Scholar are presented.

Search Query	Search Query Results
("data warehouse") AND (advantage OR benefit OR strength OR advantages OR benefits OR strengths) AND (weakness OR disadvantage OR weaknesses OR disadvantages)	91
("data lake") AND (advantage OR benefit OR strength OR advantages OR benefits OR strengths) AND (weakness OR disadvantage OR weaknesses OR disadvantages)	3
("data lake") AND (advantages OR benefits OR strengths OR disadvantages OR weaknesses)	88
("data lake") AND (advantages OR benefits OR strengths)	85
("data lake") AND (disadvantages OR weaknesses)	6
(lakehouse OR data lakehouse) AND (advantage OR benefit OR strength OR advantages OR benefits OR strengths) AND (weakness OR disadvantage OR weaknesses OR disadvantages)	0

(lakehouse OR data lakehouse) AND (advantages OR benefits OR strengths)	0
(lakehouse OR data lakehouse) AND (disadvantages OR weaknesses)	0
(lakehouse or “data lakehouse”)	3

Table 3: Search Queries and Results in Scopus for Sub-Questions 2 and 3

Search Query	Search Query Results
Data lakehouse advantages and disadvantages	450
Data lakehouse benefits	1880
Data lakehouse strengths and weaknesses	829
Data lakehouse	4140
Lakehouse	8480

Table 4: Search Queries and Results in Google Scholar for Sub-Questions 2 and 3

Lastly, Tables 5 and 6 present the query results used for sub-questions 4 and 5. Given the novelty of the lakehouse architecture, we did not expect many search results. This is reflected in the number of search query results in the Scopus database. However, as already mentioned, sub-question 4 and 5 are partly answered by literature but mainly by the experiences and thoughts of experts.

Search Query	Search Query Results
(lakehouse OR “data lakehouse”) AND (challenges OR future perspective)	1

Table 5: Search Queries and Results in Scopus for Sub-Questions 4 and 5

Search Query	Search Query Results
Data lakehouse challenges	2440
Data lakehouse future perspectives	2120

Table 6: Search Queries and Results in Google Scholar for Sub-Questions 4 and 5

3.2.1.3. Inclusion and Exclusion Criteria

In order to select the set of articles we would use for this study; the following inclusion and exclusion criteria were formulated:

- IC1: The paper is in English and available for download
- IC2: The paper addresses the search query directly
- IC3: The paper selected from Google Scholar should at least be cited by other studies
- IC4: The paper selected from Google Scholar should be within the first ten results while ordering the search results by relevance
- IC5: Web blog articles written in the past five years are seen as up-to-date and are eligible to be included
- IC6: The web blog article is written by credible authors whose purpose is to inform their audience
- EC1: The paper, article, or blog does not fit the scope of this research
- EC2: The paper, article, or blog does not have any evidence or reference for their claimed statements

These inclusion and exclusion criteria will support the selection of the most suitable, reliable, and appropriate scientific literature and blogs found through the Google search engine. Given that Google Scholar and the Google search engine show anything that is published when it is related to the topic, it is of high importance to critically look at the level of reliability of this source. This is done by looking at the credibility of the authors, whether it has been used in other studies, whether it entirely fits with our scope, and it shows some evidence of the claims they make by showing examples from practice or referencing other work.

3.2.1.4. Article Selection

Because this literature research had a specific focus, the selection of articles was mainly based on scanning through the paper with the inclusion and exclusion criteria in mind. The papers that were shown for each search query were opened if access was allowed, after which we searched for a specific term, e.g., advantage or benefit, in order to read through the paragraph where this was mentioned. This way, time spent on selecting an article was minimized since we did not read the entire paper. This was done for all the results generated from search queries on Scopus and the first ten results shown on Google Scholar.

3.2.2. Qualitative Method: Interviews

After building the artefact based on the findings of the performed literature research, interviews were conducted. This method is generally used to obtain in-depth information on the respondents' experiences and viewpoints. For this research, interviews were conducted to evaluate the model for validation, gain insights into the interviewee's practical experience, and gather their thoughts on the architecture of a lakehouse and how the architecture could be best visualized conceptually. The data collected through the literature research and interviews serve as the primary information resources for this research.

There are multiple reasons why interviews are advantageous as a method for data collection. According to Watson (1997), interviews have four significant advantages relative to surveys. First, with an interview, there is a direct interaction between the researcher and the respondent. Although this interaction does not always occur in real life, given that the world had to endure a pandemic, there is much to gain from an interview, even from online interviewing. For instance, the researcher is able to learn from non-verbal cues such as facial expressions, hand gestures, posture, the sound of the voice, and the emotions of the respondent. Secondly, the respondents are unconstrained in their way of responding. Unlike surveys, where you have limited options to select as an answer, an interview allows the respondent to formulate answers in their own words. Another advantage is that the researcher can further clarify their answer and probe to gain a greater understanding. Lastly, Watson (1997) argues that the interviewer is in a better position to motivate participation. Since interviews are always face-to-face, in real-life or virtual, the interviewer is in a better position to encourage the respondent to think critically about their answers. Moreover, the interviewer is even in the position to motivate reluctant respondents to participate. This last benefit is, however, not applicable to this research, given that all the respondents voluntarily participated in the interview.

3.2.2.1. Interview Design

In general, there are three distinct ways in which an interview can be designed (Turner III & Hagstrom-Schmidt, 2010).

1. Informal conversational interview
2. General interview guide approach
3. Standardized open-ended interview

The first type of interview can be worded as an unstructured interview where the interviewer and interviewee simply have an informal conversation and relies on "the spontaneous generation of questions in natural interaction" (Gall et al., 2003). Hence, there is no predetermined set of structured questions, and each interview can go in a different direction and have other focus areas. This offers a lot of freedom; however, the downside is that no interview will be the same, making it impossible to evaluate the results in a structured manner. Consequently, this type of interview design was not considered for this research.

The second type, the general interview guide approach, is more structured than the first; however, there is still room for some flexibility. This applies to the way the interviewer poses the interview questions. Consequently, the respondents may provide inconsistent answers to the same question based on how the interviewer posed them. On the contrary, this informal way of conducting an interview allows the opportunity to ask follow-up or probing questions based on the answers given to the predefined

interview questions. Thus, the interview was structured to some extent, given that a set of questions was determined before the interview. However, each interview was shaped according to each respondent. The strength of this type of interview is that “the same general areas of information are collected from each interviewee. This provides more focus than the informal conversational approach, but still allows a degree of freedom and adaptability in getting information from the interviewee” (McNamara, 2006).

The third type is, in general, the most-used design for an interview. The standardized open-ended interview is highly structured in terms of the wording of questions. This means that the same questions are asked in the same way to every respondent. Given that the questions are open-ended, the respondents still have all the freedom to answer each question in their own way. On top of that, the interviewer can still pose follow-up or probing questions. Therefore, this type of interview is close to being a structured interview. However, due to the freedom of asking follow-up and probing questions, this type of interview can be described as a semi-structured interview.

The second and third types of interviews were both considered for this research. The possibility of having more freedom and flexibility with the general interview guideline approach was very appealing, given that the interview aimed to gather as much information as possible. However, given the defined scope of this research, the research questions, and the conceptual model constructed based on literature findings, the interview should focus on very specific aspects. The general interview guideline approach is a little too flexible when considering this. Therefore, the standardized open-ended questions design has been used for this research.

3.2.2.2. Interview Structure

Before the interview was officially started, the interviewer would start with an introduction combined with the principles designed by McNamara (2006). Hence, the start of each meeting followed the following structure:

- Background information on the researcher
- Introduction to the thesis
- Explain the purpose of the interview
- Address terms of confidentiality
- Indicate how long the interview usually takes
- Tell them how to get in touch later if they want to
- Ask them if they have questions before the interview is started
- Ask for permission to record

The interview was structured in two parts, and the interview questions can be found in Appendix A. The first part focused on the advantages and disadvantages of each storage architecture. This was done with the help of the artefact built based on insights from the literature. The model includes three entities, each representing one storage architecture, and each entity was first shown on three separate slides (Appendix B). For each entity, the interviewees were asked to evaluate what was included in the model. Based on their practical experience, they would share whether they agreed or disagreed with certain aspects and their thoughts on improving the model. Afterwards, the entire model was shown. The interviewees were then asked to evaluate how the three entities were put into context and how the relationships were modelled.

The second part was entirely focused on the architecture of a lakehouse to get to the core of this concept. For this, three architectures that were found in the literature were shown to the interviewees (Appendix B). These were all evaluated by looking at how each architecture was constructed. Also, the interviewees were asked to focus on three aspects: security, data governance (specifically data lineage), and the API layer. Finally, the future perspective of the lakehouse was discussed by asking about the experiences and challenges the interviewees had faced while utilizing the lakehouse architecture and by asking whether the interviewee believed the lakehouse has the capability to replace the first and second-generation storage architectures.

To conclude the interview, two closing questions were asked to the respondent to close off the meeting. First, the respondents were asked whether they believed some vital topic or aspect was not covered by the interview questions. This was asked to ensure the completeness of the information that the interview had gathered with the interview questions. Secondly and lastly, the respondents were asked whether they had any questions for the interviewer, after which the meeting would end.

3.2.2.3. Sample Size Strategy

The most common question that is thought of when choosing a qualitative method is how large the sample size should be. Most studies argue that an essential factor is the concept of saturation when deciding on the sample size for qualitative research (Dworkin, 2012; Mason, 2010). The idea of reaching saturation can be defined as “when gathering fresh data no longer sparks new theoretical insights, nor reveals new properties of your core theoretical categories (Charmaz, 2006)”. Thus, it can be best described as reaching a point where no new or relevant data is gathered.

This is generally used as a guiding principle, but there are other factors that affect the sample size. Ritchie et al. (2003) propose seven factors that possibly affect sample size. These factors are 1) the heterogeneity of the population, 2) the number of selection criteria, 3) the extent to which nesting of criteria is needed, 4) groups of particular interest that require intensive study, 5) multiple samples within one study, 6) type of data collection method and 7) the budget and resources available (Ritchie et al., 2003)”. Each factor can argue why the sample size should be relatively large or small. Additionally, Charmaz (2006) mentions that the timeline the researcher faces and the level of experience of the researcher in terms of deciding when saturation is reached affect the sample size. Anyhow, the question of how large the sample size should be always comes down to the answer that it differs per study. “A vast number of articles, book chapters, and books suggest anywhere from 5 to 50 participants as adequate (Mason, 2010)”.

For this research, a sample size of 5 has been chosen due to several factors of the list provided by Ritchie et al. (2003) and Charmaz (2006). First of all, due to time constraints, it was not feasible to increase the size given that taking and processing the interviews is very time-consuming. Secondly, the population is considered to be relatively homogeneous, given that this research was done for a specific department within the company. This department comprises 85 individuals who can be categorized according to four disciplines of expertise within the department. Thirdly, of the entire population, four individuals are experienced technology architects. Due to misfitting schedules, two of them could not be included in the sample. Consequently, six other individuals were considered. Based on the availability of the individuals and their level of expertise, three experts in the field of data engineering were included in the sample. Lastly, after considering who to include in the sample, the researcher assumed that the point of saturation would likely be reached with the individuals that were included in the sample. Hence, given that conducting interviews is labour-intensive and reaching saturation is essential, the researcher has chosen not to include more individuals in the sample and leave it at 5.

3.3. Data Evaluation Method

In order to evaluate the data that was gathered during the interviews, the answers that the interviewees gave were first transcribed. This transcription was then used to create a table in which the interviewees are represented in the columns, and each row represents an interview question. The interviewer then formulated a brief answer that would capture the essence of what the interviewee had shared. In order to structure this evaluation process, several qualitative approaches were examined, and one was chosen to serve as the evaluation method for this thesis.

3.3.1. Qualitative Research Approaches

Five different qualitative approaches proposed by Creswell (2007) were considered to determine how to analyse the interview results. The five research methods he presented were 1) narrative research, 2) phenomenology research, 3) grounded theory research, 4) ethnography research, and 5) case study research. Table 7 shows a summary of the characteristics that were taken into account when choosing a research method.

Given the description for the first characteristic, the focus of each method, only two research methods remained that were to be considered for this thesis. The grounded theory research was eliminated since this thesis does not aim to develop a new theory. The ethnography research was eliminated since interpreting a culture-sharing group does not apply to this research. At the same time, the case study research was also eliminated, given that no case study will be evaluated.

Deciding between the narrative and phenomenology approaches was a bit more complicated given that specific characteristics of one method are applicable, but also a few characteristics of the other method are very suitable for this thesis. In the end, the narrative research approach seemed most appropriate given that this research is not studying “what all participants have in common as they experience a phenomenon (e.g., grief is universally experienced) (Creswell, 2007)” which is what phenomenology research does. Moreover, phenomenology research aims to “reduce individual experiences with a phenomenon to a description of the universal essence, a grasp of the very nature of the thing (Creswell, 2007)”. To achieve this purpose, persons who have experienced the phenomenon are interviewed, and the researcher then develops a composite description of the essence of the experience. This research is interested in the shared experience of utilising a specific technology and aims to describe the essence of the newest technology in storage solutions by interviewing individuals that share a certain level of experience and knowledge. Therefore, the phenomenology approach was considered. However, given the phenomenology study's definition and aim, this approach does not seem suitable anymore. Therefore, this research uses the narrative research approach to evaluate the interview results.

Characteristics	Narrative Research	Phenomenology Research	Grounded Theory Research	Ethnography Research	Case Study Research
Focus	Exploring the life of an individual	Understanding the essence of the experience	Developing a theory grounded in data from the field	Describing and interpreting a culture-sharing group	Developing an in-depth description and analysis of a case or multiple cases
Type of Problem Best Suited for Design	Needing to tell stories of individual experiences	Needing to describe the essence of a lived phenomenon	Grounding a theory in the views of participants	Describing and interpreting the shared patterns of the culture of a group	Providing an in-depth understanding of a case or cases
Unit of Analysis	Studying one or more individuals	Studying several individuals that have shared the experience	Studying a process, action, or interaction involving many individuals	Studying a group that shares the same culture	Studying an event, a program, an activity, more than one individual
Data Collection Forms	Using primarily interviews and documents	Using primarily interviews with individuals, although documents, observations, and art may also be considered	Using primarily interviews with 20-60 individuals	Using primarily observations and interviews, but perhaps collecting other sources during	Using multiple sources, such as interviews, observations, documents, and artefacts

				extended time in the field	
Data Analysis Strategies	Analyzing data for stories, “restory-ing” stories, developing themes, often using a chronology	Analyzing data for significant statements, meaning units, textural and structural description, and description of the “essence.”	Analyzing data through open coding, axial coding, and selective coding	Analyzing data through the description of the culture-sharing group; themes about the group	Analyzing data through the description of the case and themes of the case as well as cross-case themes
Written Report	Developing a narrative about the stories of an individual’s life	Describing the “essence” of the experience	Generating a theory illustrated in a figure	Describing how a culture-sharing group works	Developing a detailed analysis of one or more cases

Table 7: Comparison of the Five Qualitative Approaches by Creswell (2007)

3.3.2. Narrative Research Approach

The term narrative is defined by Creswell (2007) as “a spoken or written text giving an account of an event/action or series of events/actions, chronologically connected”. The way in which a narrative research approach can be taken is by “focusing on studying one or two individuals, gathering data through the collection of their stories, reporting individual experiences, and chronologically ordering the meaning of those experiences (Creswell, 2007)”. In other words, narrative research is the study of the experience of individuals about a particular topic. The aim is to capture their experiences and thoughts and then structure and order them to give meaning to what those individuals have shared.

Next to explaining what the narrative research approach entails, Creswell (2007) shares a general approach for conducting narrative research. It is not a “lock-step approach, but an informal collection of topics”. This collection of topics includes the following five actions:

- *Determine if the narrative research best fits the research problem or question.* This research is focused on capturing detailed stories or life experiences of a small number of individuals. The results gathered should be suitable for solving the research problem or question.
- *Select one or more individuals who have stories or life experiences to tell and spend considerable time with them, gathering their stories through multiple types of information.* This can be in any kind of form, from journals, letters and diaries, to analysing the behaviour and the stories they tell.
- *Collect information about the context of these stories.* With narrative analysis, the stories of individuals are placed into the context of their personal experiences and aspects such as their jobs, homes, culture, family and friends that influence their viewpoints.
- *Analyse the individuals’ stories and then “restory” them into a framework that makes sense.* The stories are rephrased in a logical order and put into a framework that consists of certain key elements to structure the answers. Often, with narrative research, the individuals do not share their stories in chronological sequence. Thus, during the process of restorying their answers, the researcher tries to put them into chronological sequence and provide causal links between ideas.
- *The researcher should collaborate with the participants by actively involving them in the research.* Since the stories of the participants are being told, the process of discussing the meaning of the participants’ stories and doing validation checks is essential.

The proposed actions by Creswell (2007) were followed in this research. First, five types of qualitative research approaches were evaluated to choose which approach best fits this study. Second of all, the researcher made a well-grounded decision on who to select as their participants for this study (Section

3.2.2.3.). After selecting the sample for this study, a set of interview questions was prepared to collect information from the participants. Considerable time has been spent with them to gather their viewpoints and experiences during the interview. During the interview, the shared stories were constantly checked and validated with the respondent to ensure that the researcher had fully understood their stories. Finally, after collecting information from the participants, the interviews were transcribed. The purpose of this was to have all the stories that were told on paper in the exact words the participants had used. This allowed the researcher to create a table in which each row represents an interview question, and the answer for each individual was then rephrased briefly to capture the essence of their experience.

3.4. Evaluation of Methodological Choices

One of the limitations we recognize is related to our selection of resources. Including Google Scholar and the Google search engine as a database is a risk. Due to the inclusivity and zero-boundary principle, sources may be incorrect, unreliable, and perhaps of poor quality. However, these risks are mitigated using the inclusion and exclusion criteria. For instance, only the first ten results were considered while having the sources filtered by relevance. In addition to that, the authors are checked for credibility by checking their background and seeing whether others referenced the study. Lastly, we evaluated whether the resource referred to any sources to substantiate their claims. These criteria will help to assess whether the source is eligible to be included critically.

One limitation of choosing a qualitative method is the fact that the data is not easily analysed. Moreover, it is labour-intensive and time-consuming to perform and analyse the interviews. Given that the data collected during those interviews have a lot of variety, the narrative analysis might also be inconsistent since the respondent never answers in the exact same way as another respondent. As a result, the narrative analysis that is performed on the collected data might have some inconsistencies. However, this is dealt with by having a structured set of interview questions and creating a table in which the results have been structured.

Another limitation of one of the methodological choices is related to the sample size. Even though we have argued why our sample consists of 5 participants, this sample size is still considered to be small. Moreover, the participants are all employees at Avanade. Hence, they may not represent the entire population of architects and data engineers. Nevertheless, given the time boundaries of this thesis research and the fact that this research was performed within Avanade, this sample is considered to be reliable and credible for this research.

One limitation of conducting interviews is that respondents can be biased. Every individual holds some bias to some extent, both implicit and explicit bias. Hence, it should be considered that the answers they provide can be biased to some extent. Even though some of the interview questions were factual, the participant was asked to substantiate their answers with their experiences. In some cases, sources were mentioned and shared on which the participant had based their response. Still, we recognize that most of the answers provided are entirely based on their personal experience. In order to limit the bias, measures like a structured literature review and following a structured interview protocol have been carried out. Thus, whenever participants would lean toward a specific cloud vendor, the researcher asked whether the participant was aware of this and whether they could also focus on other vendors. If this was not possible, the researcher would perform some additional research to avoid being biased toward a specific vendor.

Lastly, one of the limitations of conducting interviews is that the interviewer can interpret the answers given differently than the respondent intended. However, this limitation is tackled in two ways. First of all, the researcher carried out validation checks during the interview by repeating what the respondent had said briefly and asking whether this was correctly understood. In addition to that, the interviews were recorded, so the interview notes were not based on the recollection of the interviewer. Thus, the researcher could always refer back to the recording. In this way, we believe that the level of unclarity and perhaps even bias of the interviewer has been limited to a minimum.

4. Building the Data Storage Evolution Model

This chapter explains the design of the artefact that was built based on insights that were derived from literature. The artefact is a model that captures the strengths and limitations of the data warehouse, data lake, and data lakehouse. The chapter starts by briefly introducing the evolution of storage architectures, presenting the model and explaining how the model was constructed. Additionally, an explanation is provided on how an architecture of a data warehouse, data lake, and data lakehouse is built.

4.1. Evolution of Storage Architectures

The currently existing storage architectures are the data warehouse, data lake, and data lakehouse and the entire evolution process is visualised in Figure 9.

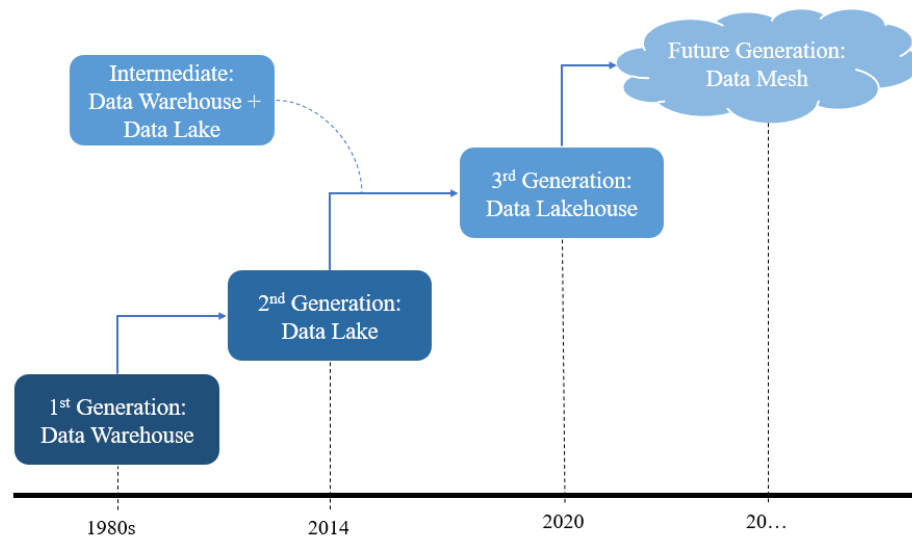


Figure 9: Evolution Process of Storage Architectures

The evolution started with the introduction of a data warehouse. This was first introduced in the 1980s because of an increasing need to store data centrally and have decisional support systems in place to gain a competitive advantage. With a data warehouse, companies would now be able to collect, store, and process data to obtain valuable information. However, with the increasing developments in the field of the Internet of Things and the rise of big data, the data warehouse started to lack specific capabilities that were now desired. As a result, in 2010, a new concept appeared: the data lake.

There was a need to have a flexible data storage architecture to support the storage of different forms of data and larger volumes of data. As a result, there was a need to have an “incubator” environment to store any type of data without knowing what type of information they could derive from it. Therefore, the data lake was introduced and gained a lot of popularity right after its introduction due to its flexible architecture and cost-efficiency. It was speculated that the data lake would, step by step, replace the data warehouse. However, this architecture also started to pose several data quality risks, security, and access control risks. Hence, the introduction of the data lake did not mean that it was better than a data warehouse and did not serve as a replacement for the data warehouse. It rather became a complement to the data warehouse. Thus, now it was possible to choose a storage architecture based on the requirements of each use case. (Madera & Laurent, 2016)

Now there are two storage architectures to choose from; however, why choose between the two when one wants to profit from the strengths of both architectures? Consequently, engineers started to create hybrid solutions and build pipelines between a data warehouse and data lake to create a hybrid solution. This two-tier architecture can now be seen as the intermediate solution between the second and third generations of storage architectures. Regrettably, this two-tier architecture posed some significant

issues, like the high total cost of ownership and keeping the data reliable and consistent. Consequently, the third generation evolved: the data lakehouse.

In 2020 the data lakehouse was introduced, which is “a new, open data management architecture that combines the flexibility, cost-efficiency, and scale of data lakes with the data management and ACID transactions of data warehouses, enabling business intelligence and machine learning on all data (Databricks, 2020)”. This architecture differs from the two-tier architecture given that you do not have a separate data lake and data warehouse but a data lake with an advanced layer on top that enables specific data warehousing capabilities (Loxton, 2022).

4.2. Designing the Data Storage Evolution Model

This study presents a Data Storage Evolution Model. This model aims to capture the essence and value of each generation of storage architecture. This model aims to help researchers and data consultants to have a summarized overview of the capabilities of each architecture and to understand the evolution process. The reason why this model is developed in this thesis is because of the following reasons:

- In literature, there is no conceptual model that compares the existing storage architectures
- In literature, there is no conceptual model that explains the value of each architecture in one model
- In literature, a limited number of studies are available concerning the data lakehouse due to its novelty
- In practice, people find it hard to grasp the true value of the data lakehouse
- In practice, no tool/model includes the data lakehouse as an option when choosing which storage architecture would fit best with a particular use-case

In order to explain this design problem, a template proposed by Wieringa (2014) has been used. Since the engineering cycle is already chosen as our method, the template for formulating the design problem has also been used. This template is as follows:

Improve *< a problem context >*
By *< (re)designing an artefact >*
That satisfies *< some requirements >*
In order to *< help stakeholders achieve some goals >*

For this thesis, the design problem template is as follows:

Compare *the existing storage architectures*
By *creating an artefact that summarizes the strengths and limitations of each storage architecture*
That *enhances and increases the awareness and knowledge of researchers and practitioners*
In order to *clarify the evolution process and highlight how the newest architecture overcomes the limitations of the existing architectures.*

The goal of the designed conceptual model is as follows: provide an overview that captures the value of each storage architecture and brings the different storage architectures together by modelling relationships between them that will explain the evolutionary developments in the field of storage architectures. The design of this model is valuable for both researchers and practitioners.

The conceptual model consists of three separate entities representing the three types of storage architectures: the data warehouse, data lake, and data lakehouse (Figure 10). They are represented in orange boxes. On top of the orange box, a list of advantages can be found that are typically related to

that particular storage architecture. Below the orange box, a list of typical disadvantages is presented. Since there is an evolution process between the three architectures, careful thought was given to how a relationship between the three could be modelled. As a result, the relationship can be explained as follows: specific advantages of Architecture A, tackle the disadvantage of Architecture B. Since the model would become quite messy with all the arrows going from one box directly to another, an intermediate box, the green hexagons, was used. These hexagons mention the advantages that can tackle another architecture's disadvantages.

Since the data warehouse was the first generation, no arrows point from a data warehouse to another architecture. The data lake is the second generation, so only arrows are pointed toward the data warehouse architecture. Whereas the data lakehouse has arrows pointing at both the data warehouse and data lake. Since it is the third generation, it can solve the disadvantages of the two precedent architectures. Therefore particular capabilities have been incorporated to tackle the weaknesses of the data warehouse and the data lake.

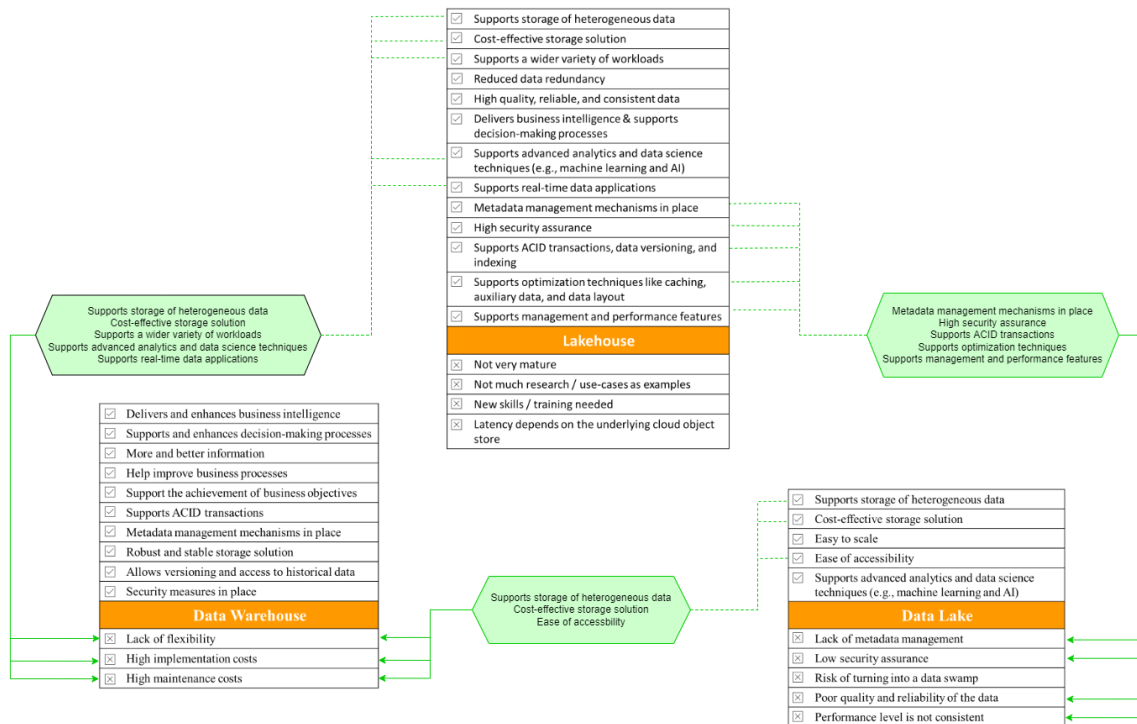


Figure 10: Data Storage Evolution Model

The presented model is built based on gathered insights from the literature. Since experts validated this model, the strengths and weaknesses that are now included are called ‘assumptions’. The following sections explain why each strength or weakness was added to the model.

4.2.1. Data Warehouses

The data warehouse was first introduced in the 1980s by IBM researchers Barry Devlin and Paul Murphy. The purpose of the data warehouse is to enhance and support the decision-making process by allowing for data collection, consolidation, analytics, and research (Keith D. Foote, 2018). Hence, the reason for the introduction of a data warehouse was to have an architectural model that assists the entire flow of data from operational systems to decision-making support systems. Bill Inmon, who is recognized as the father of the data warehouse, defined a data warehouse as a “subject-oriented, integrated, time-variant and non-volatile collection of data in support of management’s decision-making process” (Inmon, 2002). This definition can be further explained by focusing on the four key terms in the definition. First of all, *subject-oriented* refers to the fact that “all relevant data concerning a particular subject is gathered and stored in a single database” (Sheta & Nour Eldeen, 2013). Secondly, *integrated* means that data coming from different sources must be merged into a data warehouse

according to a specific format in a consistent way. Hence, the data is said to be integrated when there are no problems such as naming conflicts and inconsistencies. The third term, *time-variant*, refers to the capability of a data warehouse to tag data with time so that data can be viewed at all points in history. Finally, *non-volatile* stresses that a data warehouse is always static or stable to ensure highly consistent and reliable data. This enables a data warehouse to be heavily optimized for query processing (Sheta & Nour Eldeen, 2013). Another person who is recognized to have laid down a foundation for the data warehousing concept is Ralph Kimball. His definition of a data warehouse is "a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for decision making (Kimball & Ross, 2002)".

4.2.1.1. Advantages

The data warehouse holds numerous benefits. All benefits will be explained in the following paragraphs, and an overview is given in Table 8. One is that, given that it only stores structured data, the ability to mine and analyse the data is very convenient (Kutay, 2021). The data is cleansed, labelled, and possibly migrated with other data into structured tables. With simple SQL queries, one can obtain data and utilize it directly for analyses. As a result, business intelligence is enhanced, and the overall decision-making processes are improved due to the support of analyses and reports generated from the data stored in a data warehouse. With the help of these analyses and reports, problems and opportunities are identified sooner, leading to higher business intelligence and thus better decisions (H. J. Watson et al., 2002). Hence, the first assumed benefit of a data warehouse is that it enhances business intelligence. And the second is that it improves decision-making processes.

Looking at the fact that data warehouses enhance business intelligence and improve decision-making processes, it can be argued that another benefit of data warehouses is that it provides improved quality and increased quantity of information (Roelofs et al., 2013; H. J. Watson et al., 2002). This can be explained by the fact that data is cleansed and transformed into a format suitable for analyses and business intelligence before storing it in a data warehouse (Al-Okaily et al., 2022). Therefore, our third assumption is that data warehouses provide more and better information.

The fourth benefit of data warehouses is applicable whenever it is used to redesign and improve business processes (Watson et al., 2002; Roelofs et al., 2013). The benefit of this is a little hard to measure. However, it is guaranteed that the purpose of improving business processes is to achieve strategic business objectives. This can impact a sole department, but it can also affect the entire organization and create a significant shift in its strategy. For this to be successful, the correct information is needed to initiate the change and guide it in the right direction. Thus, our fourth assumption states that data warehouses help improve business processes. And the fifth assumption is that data warehouses support the achievement of business objectives.

One of the technical benefits of a data warehouse is the ability to perform ACID transactions (Chen et al., 2002). ACID is an acronym for atomicity, consistency, isolation, and durability. The implementation of ACID transactions contributes to the guarantee of having reliable data. Atomicity ensures that each CRUD operation (create, read, update, or delete) is done correctly. Thus, the entire operation is not executed when a process fails mid-stream. This will prevent data corruption and data loss. Consistency also ensures that the data will not become corrupt or that errors occur in your data. This is done by only making changes to the data in a predefined, predictable, consistent way. In this way, the integrity of the data is maintained. Thirdly, isolation ensures that concurrent transactions do not interfere with each other. More specifically, when multiple people are reading and writing to the same table, the transactions are isolated and performed as if they occurred one by one, even though they are happening simultaneously. Lastly, durability is about ensuring that any change that is successfully made to the data is saved. All in all, ACID transactions contribute to keeping the data consistent, integer, and reliable (Databricks, 2022). Therefore, the sixth assumption is that data warehouses support ACID transactions which is an advantage of the data warehouse.

Another technical benefit is that the data warehouse has metadata management controls in place. The metadata is essential for users to access a data warehouse efficiently. In order to do so, the user must know what data is available and where this is located. This is enabled by having metadata stored in a repository providing any information needed on the stored data. This is a crucial success factor for the data warehouse, given that it helps to better understand, manage, and use the data. There can be three different types of metadata: business metadata, operational data, and technical data (Jarke et al., 2002). Business metadata provides information on a high level regarding the ownership of datasets and legal information about the datasets. Operational data is more on a data-level in terms of information about the status of the data, whether it is archived or active, and data lineage. Lastly, the technical data is about the specifics of individual data tables, such as file types, size, and critical attributes. All this metadata serves the two purposes of minimizing the efforts for developing and administering a data warehouse and making the process of extracting information as convenient as possible (Vaduva & Vetterli, 2001). Therefore, the seventh assumption is that data warehouses have metadata management mechanisms, which is another advantage of the data warehouse.

Another advantage of the data warehouse is that they are seen as robust and stable storage solutions. As mentioned before, the data warehouse was first introduced in the 1980s and ever since a lot of research has been done into this storage architecture. Much research has been done on the data warehouse, and many updates were developed in the field of data warehouses. In the field of business intelligence, many improvements have been made in the design of the data warehouse to simplify the integration of data from multiple sources, access to the database, data enrichment, and innovative visualization through automated procedures (Bourbonnais & Morency, 2018). Hence, the data warehouse has had a lot of time to be incrementally improved with the use of insights from research and the actual updates that were developed. Therefore, the eighth assumption states that the data warehouses are a robust and stable storage solution.

The seventh assumption explained that a data warehouse's benefits are that metadata management controls are in place. This metadata is stored in a separate repository that provides different information about the data stored in the data warehouse. In addition to knowing what data is stored and where it is located, it is also possible to have information about data transformations and version control there (Jarke et al., 2002). By storing this, the user can track what changes have been made to the data and the corresponding metadata. This enables versioning of the data and obtaining access to historical data, given that the user can obtain specific versions of the data. This is perceived to be an advantage because it allows you to revert to the latest version of the data in cases of failure. Moreover, historical data is helpful for reports and predictive analyses because the more data you have, the more accurate these analyses are (Al-Okaily et al., 2022). Hence, we assume that data warehouses allow versioning and access to historical data, which is another strong point of the data warehouse.

Finally, one more advantage that is added to the model is that a data warehouse has security measures in place. Since a lot of data with a span of multiple years and sensitive data is stored, securing all this data is crucial for the sustainability and reliability of the data warehouse (Gosain & Arora, 2015). Over the years, security controls have become more enhanced and sophisticated. Security measures like masking or encrypting data, specifying access control on table level and even column-/row-level, implementing different viewing permissions based on roles, metadata describing security mechanisms, audit controls, allowing third-party integrations in terms of security services, and build-in security measures by the vendor are all existing security measures (Gosain & Arora, 2015; Rosenthal & Sciore, 2000; Vishnu et al., 2014). Given that the level of sophistication within the security level can be very granular, we believe another advantage of the data warehouse is that it has security measures in place. This is advantageous given the vast amounts of data and sensitive data stored in a data warehouse.

Assumptions
A1: Data warehouses enhance business intelligence
A2: Data warehouses improve decision-making processes
A3: Data warehouses provide more and better information
A4: Data warehouses help improve business process
A5: Data warehouses support the achievement of business objectives
A6: Data warehouses support ACID transactions
A7: Data warehouses have metadata management mechanisms in place
A8: Data warehouses are robust and stable storage solutions
A9: Data warehouses allow versioning and access to historical data
A10: Data warehouses have security measures in place

Table 8: Overview of Assumptions Describing Advantages of Data Warehouses

4.2.1.2. Disadvantages

Some studies also acknowledge the weaknesses a data warehouse holds. The following paragraphs summarize the weaknesses found and are summarized in Table 9. First of all, the fact that it only supports structured data is considered to be a weakness given the rise of semi-structured and unstructured data. Nowadays, there are so many new sources of data that generate data in different formats. Regrettably, the traditional data warehouse architecture does not support storing semi-structured and unstructured data. As a result, companies may miss out on many valuable sources that generate semi-structured or unstructured data. Therefore, our assumption states that one of the disadvantages of a data warehouse is that it lacks flexibility.

Next to that another disadvantage of the data warehouses is the high cost of total ownership. Both the implementation costs and maintenance costs are typically high (Kutay, 2021). Traditionally, data warehouses were implemented on-premises. This requires an organization to purchase all the hardware and software, which is very costly. The reason why companies would choose this is because of having more control over how and where the data is stored. When extracting data, you are not relying on high-speed internet and connectivity, and when you are in control, it also feels as if the data is more secure. However, this requires high up-front implementation costs for hardware and software, on-site IT staff, location/space for machines, and operational charges like electricity are incurred when implementing a data warehouse on-site. In addition to that, one must know that there will be regular maintenance costs that keep the data warehouse up to date. For this, both the hardware and network must be upgraded, which happens to be periodical. Besides, the database will not remain static. In other words, new data will be added, the design may change, the data warehouse will grow in size, and it will grow in the number of users, queries, and reports. All in all, the continuing maintenance costs are likely to exceed the initial implementation cost (Adelman, 2021). Hence, we assume that data warehouses incur high implementation and maintenance costs which is a significant disadvantage.

Assumptions
A11: Data warehouses lack flexibility
A12: Data warehouses incur high implementation costs
A13: Data warehouses incur high maintenance costs

Table 9: Overview of Assumptions Describing Disadvantages of Data Warehouses

4.2.2. Data Lakes

For years, the traditional data warehouses have been considered a unique storage solution to support decision-making processes. The data that is brought into a data warehouse is first cleansed and processed according to pre-defined formats and aggregated with other data in most cases. Afterwards, they are stored in the data warehouse in structured tables, and end-users can use these tables for their specific use-case. However, with the rise of the Internet of Things and Big Data, many challenges arose with this way of storing data. Not only did the volume of data increase, but the variety of data was also another significant problem that the data warehouses could not deal with efficiently. Since the data format was not always known prior to collecting the data, data pipelines for cleaning and processing

the data were not suitable anymore. As a result, with the big data wave in 2012, the second storage architecture evolution step appeared, and the data lake was introduced.

James Dixon first coined the term ‘data lake’ in 2010 (Giebler et al., 2019). James Dixon visualized a data warehouse as a bottle of water representing cleansed and structured data. Whereas the data lake is more like a “large body of water in a more natural state, typically a lake”. That is how the data lake term was introduced (Dixon, 2010). The essence of his definition was that data would be stored in its raw state. Hence, this meant that the data's original format could be maintained, and data in heterogeneous formats could be stored and accessed for various analytical use cases. Ignoring almost everything regarding storing data seems to challenge traditional ways of storing data in a data warehouse. Nonetheless, in the age of big data, new ideas and techniques for storing and processing diverse, changing, and evolving data are crucial (Khine & Wang, 2018).

This technology was increasingly adopted throughout the years, and different definitions were given to the data lake concept. Fang (2015) defines a data lake as “a methodology enabled by a massive data repository based on low-cost technologies that improve the capture, refinement, archival, and exploration of raw data within an enterprise. A data lake contains the mess of raw unstructured or multi-structured data that for the most part has unrecognized value for the firm.” To go even further into the core idea of a data lake, the following definition was also considered: A data lake uses a flat architecture to store raw data, cleansed data, and transformed data. “Each data entity in the lake is associated with a unique identifier and a set of extended metadata, and consumers can use purpose-built schemas to query relevant data, which will result in a smaller set of data that can be analysed to help answer a consumer’s question (Alrehamy & Walker, 2015).” This definition enhances the understanding of the architecture of the data lake and is therefore taken into account in this thesis.

4.2.2.1. Advantages

The following paragraphs elaborate on the strengths of a data lake, and an overview is given in Table 10. One of the apparent main advantages of a data lake is that it supports storing heterogeneous data. This is something that a data warehouse cannot do; hence, this was one of the reasons a data lake was developed. With the rise of big data and the popularization of data represented in different formats, storing heterogeneous data is crucial. The idea of storing heterogeneous data in a data lake is typically closely tethered to the Apache Hadoop ecosystem. However, other similar frameworks can also be utilized, as long the data can be stored in its raw format (Mehmood et al., 2019). The framework incorporates a flat architecture that holds all the data in a central repository. Concludingly, we assume that ‘data lakes support storage of heterogeneous data’ as one of the advantages of a data lake.

Secondly, with the big data wave, the volume of the data also started to increase immensely. Due to this growth, considering the costs of storing data have become more important than before (Fang, 2015). Consequently, one of the purposes of a data lake is to be as cost-effective as possible since traditional data warehouses are typically costly. Because of how a data lake is designed and implemented, it is a cost-effective storage solution. First of all, a data lake is typically built in the cloud, which eliminates the implementation costs that are incurred with traditional warehouses. Moreover, a data lake is designed to be stored on low-cost commodity hardware, like an object storage (Kutay, 2021). Object storage is short for object-based storage, which is a data storage architecture that stores vast amounts of unstructured data. This is usually optimized for a lower cost per GB stored. And lastly, fewer CRUD operations are needed since all the data is dumped in the data lake. This explains how a data lake can offer lower prices compared to a data warehouse. Therefore, we assume that an advantage of the data lake is that it is a cost-effective storage solution.

As explained in the previous paragraph, the sharp growth in data volume required a storage solution to be able to offer the storage of vast amounts of data at a reasonable price. Not only do the costs start to grow with a growing data volume, but also the need for scaling up your storage capacity started to grow.

Hence, a data lake's architecture is designed to be scaled for continuous growth to deal with the volume in which big data comes (Mehmood et al., 2019). Just as for being cost-effective, to allow a storage solution to be flexible, the fact that it is designed in a cloud environment contributes to achieving this. Given that the cloud can use distributed data centres to store the data, there are endless possibilities to scale up or down the company's storage. Therefore, the assumption that is added to the model is that data lakes are easy to scale.

Data lakes facilitate easy access to the data. First of all, given the ability of data lakes to store a huge variety of data, data lakes are also able to store real-time data. Since it is stored in its raw form, the real-time data will be more, better, and also near real-time accessible with the data lake (Madera & Laurent, 2016). Another reason why accessibility can be perceived as more convenient is because all the data is stored in a centralized repository. Hence, no search has to be done across different 'data marts' that are implemented in a data warehouse. Another possibility is that multiple users can access the data repository for monitoring, exploration, and analysis simultaneously (Mehmood et al., 2019). Lastly, the need for easily accessing data was also driven by the fact that end-users wanted to obtain all relevant data for data analyses (Madera & Laurent, 2016). Consequently, all the end-users of the data lake can access the stored data and easily scan it to gather relevant data. A data warehouse has more of a seek-to approach (Fang, 2015). Therefore, the advantage that data lakes facilitate easy access to the data has also been added as an assumption.

Finally, data lakes support advanced analytics and data science techniques. Given the fact that data lakes support the storage of data in their raw format, the methods of processing the data have become more advanced. Now, it is easier to apply various machine and deep learning algorithms to perform predictive analytics, obtain meaningful insights and find patterns (Kutay, 2021). Moreover, as mentioned before, a data lake supports the storage and access of real-time data, therefore, performing real-time analytics is also supported, which is a more advanced type of analysis (Madera & Laurent, 2016). Next to handling real-time streams, a data lake is also capable of handling batch processing at scale (Fang, 2015; Mehmood et al., 2019). The team of people that is typically able to perform these types of analyses are the data engineers and the data scientists. Therefore, because of the support of different kinds of workloads in a data lake, all the stakeholders in a business environment can utilize the data for their types of analyses. Therefore, we assume that an advantage of the data lake is that it supports advanced analytics and data science techniques.

Assumptions
A14: Data lakes support the storage of heterogeneous data
A15: Data lakes are cost-effective storage solutions
A16: Data lakes are easy to scale
A17: Data lakes facilitate easy access to the data
A18: Data lakes support advanced analytics and data science techniques

Table 10: Overview of Assumptions Describing Advantages of Data Lakes

4.2.2.2. Disadvantages

Besides the advantages, there are also a few disadvantages related to data lakes. An overview is provided in Table 11, and an explanation for each is given in the following paragraphs. According to several studies, it is quite a challenge, hardly possible, to address general requirements for metadata management. Because of the integration of various data sources, particular metadata management requirements are necessary. However, this appears to be quite a challenge, especially when making metadata available for the raw data (Mehmood et al., 2019). There have been several attempts to develop the concept metadata catalogue to guarantee the respect of data governance and properly manage metadata (Madera & Laurent, 2016). Unfortunately, this seems to be very challenging due to the fact that a data lake accommodates all data formats. As a result, the data lake is not able to provide specific

mechanisms for handling the management of metadata. Therefore, one of our assumptions states that data lakes lack metadata management.

Implementing the appropriate security controls also appears to be challenging. This is also caused by the fact that different data formats are stored in a data lake (Kutay, 2021). Thus, it is challenging to implement proper data security for each data format without knowing beforehand what data formats will be ingested and stored in your data lake. Sometimes the data is structured so vaguely, which makes it hardly possible to have security measures in place that have considered this specific format in advance. One study explained that the only way of securing a data lake is by providing access only to whitelisted IPs (Mehmood et al., 2019). In that sense, there is some security control in place regarding access control, although this study recognized that this is still insufficient to protect the data against all kinds of attacks. Thus, it is fair to say that data lakes are far from adhering to privacy, and regulatory requirements, given the limited possibilities in implementing security controls. As a result, we assume that one of the disadvantages of a data lake is that it provides low-security assurance.

Basically, when looking at the previously mentioned disadvantages, data lakes risk turning into data swamps (i.e. messy data). Because there are no proper metadata management and security controls implemented, the data can quickly become a mess and prone to security issues which reduces the integrity and reliability of the data (Khine & Wang, 2018). As a result, the data lake can turn into a data swamp. Another reason why it can turn into a data swamp is because of the lack of governance principles (Madera & Laurent, 2016). This is part of the metadata management strategy but is worth mentioning for this assumption in particular. Once there are no controls or principles in place that regulate security, data governance, and (meta)data management, the chances are high that the data storage becomes a mess. Therefore, the third assumption that describes a disadvantage of the data lake is that it has a risk of turning into a data swamp.

Another disadvantage of the data lake is that it can have poor quality and unreliable data stored in the data lake. This can be explained by the previously mentioned disadvantages of the data lake. When the stored data is not adequately managed and governed, there is a high risk of data becoming corrupt or data lakes being infiltrated by unauthorized users, which puts the quality and reliability of the data at risk. Therefore, numerous studies concluded that the data is likely of low quality and reliability due to a lack of proper metadata management and security controls. Hence, we have concluded this as a disadvantage, and the assumption states that data lakes have poor quality and reliability of data.

The very last disadvantage that has been included is that the performance level of data lakes is inconsistent. Due to the storage of different data formats, different workloads are supported. From standard reports and generating business intelligence to advanced modern workloads that include machine learning and data science techniques. Although flexibility is a strong point of the data lake, the performance, however, is inconsistent when comparing the performance level of all the different workloads that are supported (Kutay, 2021). A data warehouse makes it very convenient to perform business intelligence analyses and generate reports because the data is nicely structured. A data lake should enable the user to perform the same analyses, however, when a data lake is not adequately managed, it can become very disorganized. This makes it hard to connect business intelligence tools to the data lake since it cannot work with nicely structured data. Also, the lack of ACID transactions and inconsistent data structures can result in sub-optimal query performance, hindering the performance for reporting and business intelligence use cases. On the contrary, the performance for machine learning and data science use-cases is actually very good as long as the end-user is capable of performing these use-cases correctly. Due to these differences in performance levels, we assume that the performance level of data lakes is inconsistent, and we perceive this to be a disadvantage.

Assumptions
A19: Data lakes lack metadata management
A20: Data lakes provide low-security assurance
A21: Data lakes have a risk of turning into a data swamp
A22: Data lakes have poor quality and reliability of the data
A23: Data lakes have inconsistent performance levels

Table 11: Overview of Assumptions Describing Disadvantages of Data Lakes

4.2.3. Data Lakehouses

After the first generation of data warehouses in the late 1980s and the second generation around the 2010s, a new generation was introduced in 2020. Even though a data lake tackled some of the downsides of the data warehouse, the data lake itself also had some inconveniences. The robustness and reliability of a data warehouse were lacking in a data lake mainly due to the fact that it does not support ACID transactions, data quality is not enforced, and it lacks governance and security controls. Even though the data lake tackled some downsides of the traditional warehouses by being a flexible and cost-effective storage solution that was easy to scale. Since both generations have their strengths and limitations, the third generation is supposed to be a combination of the best features of a data warehouse and a data lake. This gave rise to the data lakehouse in 2020.

Data lakehouse can be defined as “a data management system based on low-cost and directly-accessible storage that also provides traditional analytical DBMS management and performance features such as ACID transactions, data versioning, auditing, indexing, caching, and query optimization. Lakehouses thus combine the key benefits of data lakes and data warehouses: low-cost storage in an open format accessible by a variety of systems from the former, and powerful management and optimization features from the latter (Armbrust et al., 2021).” This is the only definition that was found in the literature, and a few other existing studies that examine the lakehouse platform all cited the definition provided by Armbrust et al. (2021). Before the lakehouse was introduced, many companies attempted to implement a two-tier architecture to gain the benefits from both a data warehouse and a data lake (Figure 11).

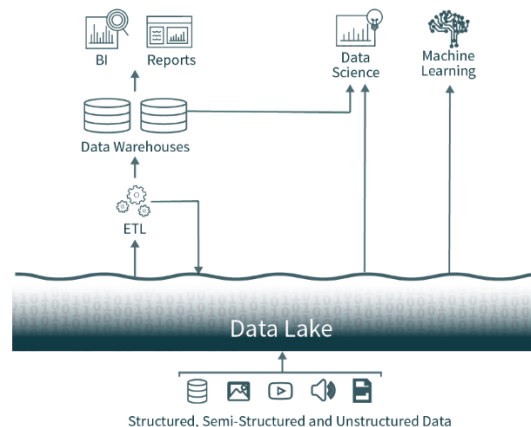


Figure 11: Two-tier Architecture with a Data Warehouse and Data Lake (Armbrust et al., 2021)

However, the two-tier architecture suffers from four problems. First of all, keeping the data reliable and consistent in a two-tier architecture is difficult and costly. The continuous ETL data transfers between the two systems are doomed to fail due to the differing nature of a data warehouse and a data lake. The data quality deteriorates quickly due to failure risks and bugs. Secondly, it is a challenge to deal with data staleness. Stale data does not necessarily mean that the quality of the data is insufficient, but rather that the data is not fluctuating with time and updates. In other words, when data is stale, it means the data is out of date and might not have been used or accessed for a significantly long time. In 2020, a global survey was conducted in which 496 qualified participants were included, and they found that 86% of analysts deal with stale data, of which 41% use data that is more than two months old, and 62%

wait on engineering resources numerous times per month (Fivetran, 2020). Thirdly, there is limited support for advanced analytics. Businesses desire to answer predictive questions with the use of warehousing data. Unfortunately, none of the leading machine learning systems works well on top of warehouses. One of the recommendations to deal with this is to add a third ETL step where data from the warehouse is loaded back into the data lake. However, this would mean that 3 ETL processes further increase complexity and data staleness. The other recommendation is to have the machine learning systems run on data in open formats. However, the downside of this is the lack of rich management features that data warehouses are able to provide, like ACID transactions, data versioning, and indexing. Lastly, a significant downside of the two-tier architecture is the total cost of ownership. Given that you have a data warehouse and a data lake, you have to pay double the storage cost, you spend significant engineering costs for keeping the data between the two storages in sync, and the usual charges for continuous ETL are to be added to the total cost of ownership. Hence, the downside is that this two-tier architecture is very costly. (Armbrust et al., 2021; Boyd, 2021) With the introduction of the lakehouse architecture, it is believed that its design is compelling and has the potential to deal with these challenges.

As mentioned before, the lakehouse was introduced in 2020. Hence it should be noted that the concept is still a work in progress. Despite its novelty, the lakehouse has proven to be successful in various forms in the industry. Databricks, the initial founder and first vendor to introduce and produce a lakehouse, is still improving this concept by adding support for new features. Besides, the lakehouse architecture is open-sourced, thus, any other vendor can see how the lakehouse is constructed and create a version of their own with perhaps slightly different features or combine the lakehouse with their architectures and APIs.

4.2.3.1. Advantages

Before explaining the advantages that have been included in the model, it is of importance to highlight that the data lakehouse is supposed to be a combination of the strengths of a data warehouse and a data lake. Consequently, several advantages that are listed here are already mentioned in the list of advantages for the data warehouse or the data lake. Since the advantages were already explained in the corresponding sub-section, they will not be repeated here. Still, a complete overview of all the advantages of a data lakehouse is presented in Table 12.

Assumptions A29, A32, and A34 are strengths of a data warehouse that are also applicable to the data lakehouse. Hence, to explain why these are perceived as advantages, we refer to section 4.2.1.1. As for the strengths of the data lake, assumptions A24, A25, and A30 apply to both the data lake and the data lakehouse. An explanation of these advantages can be found in section 4.2.2.1.

The first assumption that is unique to the data lakehouse in our model is A26 stating that data lakehouses support a wider variety of workloads. This is a lakehouse-specific advantage because it provides direct access to the most widely used business intelligence tools to enable advanced analytics. Additionally, it allows the implementation of all different kinds of APIs and machine learning libraries to perform the more modern workloads that are performed by data engineers and data scientists (Kutay, 2021). This is a clear indication of how the strengths of both the data warehouse and data lake are coming together in the data lakehouse. Given that data warehouses are optimized for business intelligence analyses, and data lakes are better capable of facilitating modern workloads. Since the data lakehouse is capable of facilitating both properly, we assume that an advantage of the data lakehouse is that it supports a wider variety of workloads.

Another advantage of the data lakehouse is that it supposedly reduces the level of data redundancy. Redundancy can be achieved either through the storage of deduplicated data or the fact that only relevant data is stored (Kutay, 2021). As for the first, the data lakehouse reduces data duplications by providing a single, centralized, all-purpose repository. It is not just a data lake where anyone can dump any data. It is also not a hybrid version in which data is stored in a data lake and a separate data warehouse that is connected to the data lake. In these scenarios, this likely results in many data

duplicates. Whereas the data lakehouse has a central all-purpose platform to cater to all the business data demands, and duplicates can be managed and removed. Concerning the storage of relevant data, a data lake literally has no filter, and any data can be stored here. This also means irrelevant data that will never be used for analysis are stored. On the contrary, the data warehouse has a massive filter where only data can be stored when it adheres to a fixed schema. Thus, an evident consequence of this is that it misses out on valuable data that has valuable information but is not coherent with the schema that was pre-defined and can therefore not be used. Whereas the data lakehouse is in the middle where data from all different sources can be stored, however, there is a lightweight filter in terms of deduplicating the data. This will result in storing only relevant data and freeing up space by removing the duplicates that are not necessary to be stored. As a result, we assume that a data lakehouse reduces the level of data redundancy.

Another assumption is that the data lakehouse contains high-quality, reliable, and consistent data. This claim is based on the fact that a metadata layer is implemented on top of the data lake. This layer typically supports management features like having ACID transactions in place, proper data governance controls, and security measures. All these features contribute to maintaining high reliability and consistent data. The metadata layer is also where data quality enforcement features are typically implemented. For instance, in Delta Lake, which is developed by Databricks, schema enforcement can be implemented in the metadata layer. With schema enforcements, it is possible to ensure that the data that is uploaded to a table matches the schema of the table. Additionally, users are able to define specific constraints for data that is being ingested through a data lake pipeline to improve its quality (Armbrust et al., 2021).

A data lakehouse's fourth advantage is that it typically supports real-time streaming use-cases (Lorica et al., 2020). This is achieved through an integration of streaming systems such as Kafka and Spark (Orescanin & Hlupic, 2021). Specifically, efficient streaming I/O (input/output) is supported, letting streaming jobs write small objects into the table at low latency (Armbrust et al., 2020). Before the lakehouse, this was mostly done with a separate streaming storage system that could not be natively integrated to the object storage. However, the data lakehouse developed by Databricks provides the appropriate features for all use-cases, including streaming use-cases. It should be noted, however, that Delta Lake is limited by the latency of the underlying cloud object store for these streaming workloads. Since it is a huge benefit of the data lakehouse that it can support streaming use-cases, this has been included as an assumption in our model.

Studies say the security, in terms of access control and privacy, in a data lakehouse is both mature and still maturing (Orescanin & Hlupic, 2021). The reason for this is that due to the implementation of ACID transactions and a metadata layer to enforce data integrity, it is possible to implement robust data security mechanisms just as it is implemented in a data warehouse. However, given that all the data is stored in a data lake object store, it is impossible to have security at a fine-grained level (Kutay, 2021). The allowance for storing heterogeneous data makes it complicated to implement proper security measures for each data file format. Despite that, there are numerous practices supported by the data lakehouse to enforce security. For instance, audit logging is a data security best practice that tracks and documents the activity of events like the time at which the event occurred, the responsible user/device/service, and the impacted entity (Armbrust et al., 2020). Reviewing these audit logs allows user activities to be easily tracked, which helps detect breaches and ensure compliance with regulatory requirements. Another example is the availability of access controls to regulate who has access to what dataset. All in all, the lakehouse is capable of implementing security measures as a traditional database would do, only not everywhere on the most fine-grained level due to the heterogeneous data formats. Still, we assume that a data lakehouse provides a high-security assurance and is therefore included as an advantage in the model.

The following assumption applies to the specific lakehouse concept that is developed by Databricks, which is called 'Delta Lake'. "Delta Lake uses a transaction log that is compacted into Apache Parquet format to provide ACID properties, time travel, and significantly faster metadata operations for large tabular datasets (Armbrust et al., 2020)". This Delta Lake is basically an approach for implementing

the Data Lakehouse concept, and it uses a metadata layer over a data lake storage object (Orescanin & Hlupic, 2021). When implementing the Delta Lake provided by Databricks, several techniques to implement optimization techniques for SQL performance are available. These techniques do not depend on a chosen data format hence, they can be applied to existing or future unknown formats (Armbrust et al., 2020, 2021). The first technique is caching, which allows files to be cached from the cloud object store on faster and local storage devices such as SSDs and RAM on the processing nodes. This is beneficial given that caching increases the data retrieval performance since the underlying storage layer does not need to be accessed. Secondly, auxiliary data structures are supported, which is the same as saying a ‘helper data structure’. To solve a specific problem, extra space can be used for a temporary period in which data is copied and transformed into a particular structure to solve the problem. Lastly, the data lakehouse system can make optimization decisions about the data layout. This includes decisions like automatically optimizing the size of objects in a table and clustering data records to access and read those records more easily and faster. These three techniques lead to optimized SQL performance in terms of velocity and accessibility. Hence, we assume that an advantage of the data lakehouse is that it supports these three optimization techniques.

Lastly, data lakehouses support management and performance features. This has already been covered by several advantages that are included in the model. However, given that this is such a strong capability of the lakehouse, it has been included as a distinct advantage. This is an outstanding advantage because the data lakehouse is capable of implementing the typical management features that are implemented in a data warehouse and enforcing these features on the data lake, which is the underlying object store. The management features are primarily represented in the metadata management layer that is built on top of the data lake. Whereas the performance features are mostly related to the possible optimisation techniques and the capability of supporting all kinds of workloads while guaranteeing decent performance for all use-cases. Therefore, we assume that an advantage of the data lakehouse is that it supports both management and performance features.

Assumptions
A24: Data lakehouses support the storage of heterogeneous data
A25: Data lakehouses are a cost-effective storage solution
A26: Data lakehouses support a wider variety of workloads
A27: Data lakehouses reduce the level of data redundancy
A28: Data lakehouses contain high-quality, reliable, and consistent data
A29: Data lakehouses deliver business intelligence & support the decision-making process
A30: Data lakehouses support advanced analytics and data science techniques
A31: Data lakehouses support real-time data applications
A32: Data lakehouses have metadata management mechanisms in place
A33: Data lakehouses provide high-security assurance
A34: Data lakehouses support ACID transactions, data versioning, and indexing
A35: Data lakehouses support optimization techniques like caching, auxiliary data, and data layout
A36: Data lakehouses support management and performance features

Table 12: Overview of Assumptions Describing Advantages of Data Lakehouses

4.2.3.2. Disadvantages

It is always hard to identify the disadvantages or weaknesses of new technology. If there were a lot of disadvantages, it would not have been logical for it to be spread across the market and research. Despite the novelty of the data lakehouse, four disadvantages were identified and formulated as assumptions. They are presented in Table 13 and will be explained in the following paragraphs. The first one is that data lakehouses are not very mature. As explained in the introduction of this sub-section, the data lakehouse was introduced to the market in 2020. This is only two years ago, while a data warehouse has been in existence for over 40 years now and a data lake for more than ten years. The two previous data storage solutions have been under a loop for quite a while, and numerous advancements and improvements were developed. As for the data lakehouse, it is unclear whether it will be able to live up to its promises since it is still very new (Kutay, 2021). Therefore, we assume that a disadvantage of the data lakehouse is that it is not yet very mature compared to the other storage solutions.

The second disadvantage is closely related to the previous one. Namely, there is not much research or use-cases as examples available regarding the implementation of a data lakehouse. This is perceived as a disadvantage given that this storage solution has not been exploited yet to the fullest, and probably not many updates have been developed when there are no practical use-cases available. Hence, it is impossible to refer to many use-cases, and there is little room for improvement. This will change in the future. However, as of now, we assume that a disadvantage of the data lakehouse is that there is not much research or use-cases as examples available.

Thirdly, given that the data lakehouse is a new technology, it requires training for new skills. This can be seen as a disadvantage mainly because people can be reluctant to change or are not motivated to learn new things. It may even be that this technology is too complex for people to understand and take a long time to master the necessary skills. It is not entirely proven yet, though in general, with new technologies, new skills are required. Therefore, this disadvantage has been included in the model.

Finally, the last disadvantage is not people-related but technical-related. We assume that a disadvantage of the data lakehouse is that the latency depends on the underlying cloud object store. An object-store is short for object-based storage, which is a type of architecture that handles large amounts of unstructured data. Since a data lake is typically used for the lakehouse architecture and a data lake is generally stored in the cloud, the latency depends on the chosen cloud object store. With a traditional data warehouse, this problem does not occur. Even though this has not formed such a disastrous disadvantage of the data lakehouse, one should be aware that latency might come in and can be high or low depending on the chosen object-store.

Assumptions
A37: Data lakehouses are not very mature
A38: There is not much research or use-cases as examples available regarding data lakehouses
A39: Data lakehouses require training for new skills
A40: The latency of a data lakehouse depends on the underlying cloud object-store

Table 13: Overview of Assumptions Describing Disadvantages of Data Lakehouses

4.3. Architectures of the Data Storage Solutions

While the big data wave emerged over the years, the data storage solutions also started to develop and adjust themselves accordingly. Nowadays, data warehouses are seen as traditional data storage solutions. For a long time, this has been considered to be the unique solution that was able to deliver accurate and reliable information to an organization. With the big data wave around 2010, the second information architecture evolution rose. First of all, there was a need for a storage solution that could deal with the challenges of big data. Secondly, data governance became a crucial part of the new information architecture since privacy and compliancy became more important. As a result, the data lake was developed, which can be seen as an “incubator” environment in which data of any type can be stored to generate insights from (Madera & Laurent, 2016). This storage architecture challenged the whole rationale of a data warehouse and can be seen as the exact opposite. However, it turned out that new problems occurred with the data lake technology and the governance and security aspects were still not fully solved. Besides, the emergence of the big data wave has not yet stopped, society is still shifting toward a digitalised society, and the continuous development of new technologies is still going on. Consequently, this gave rise to a third information architecture: the data lakehouse. This can be defined as “a data management system based on low-cost and directly-accessible storage that also provides traditional analytical DBMS management and performance features such as ACID transactions, data versioning, auditing, indexing, caching, and query optimization (Armbrust et al., 2021).”

Figure 12 displays the developments in the storage architectures in relation to the big data wave. As previously mentioned, the big data wave has not stopped and is expected to continue. When looking at the trend showing the development of data storage architectures, the line started with a very steep line. The reason is that the data warehouse and data lake were disruptive technologies. Afterwards, the line

tends to slow down, indicating the newer technologies not to be as disruptive as the previous ones. This claim is based on the fact that a data lakehouse incorporates best practices from the data warehouses and data lakes. As for the data meshes that are expected to be the next generation, the rationale will be built on the ideas of the data lakehouse. As a result, the trendline showing the evolution of storage architectures is flattening.

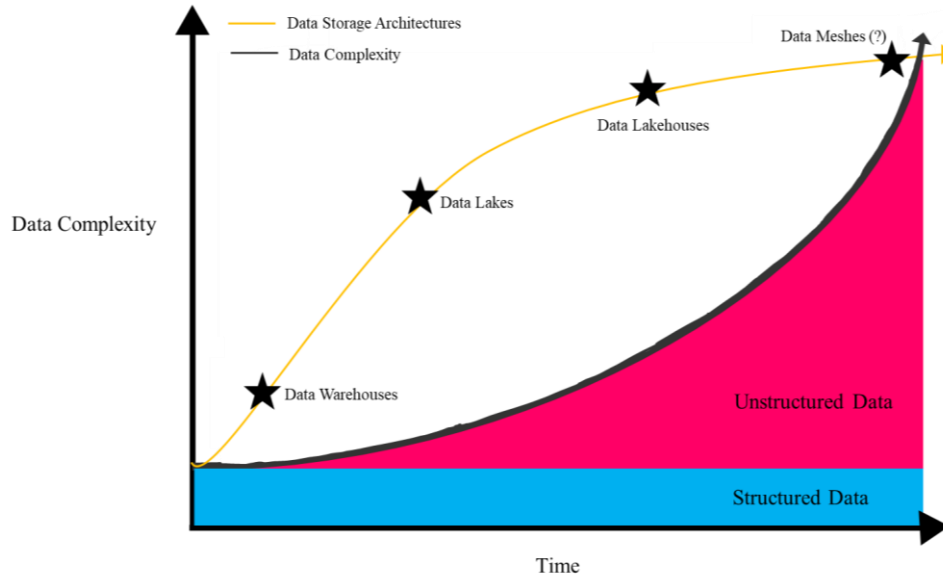


Figure 12: Evolution of Storage Architectures in Relation with Data Complexity

In the following sub-sections, the architecture of each storage architecture will be presented and explained. Understanding how the architecture is constructed is required to understand why certain benefits are associated with a specific storage architecture. Moreover, in the literature, only one architecture was presented for the data lakehouse. However, the architecture we found was not as fine-grained as the architectures that explain the data warehouse or data lake. Therefore, this research aims to understand the evolution of architectures and how they have been constructed to present a fine-grained and sophisticated architecture for the data lakehouse. Table 3 presents a high-level comparison between the three storage architectures (Kutay, 2021; Lavrentyeva & Sherstnev, 2022; Segal & Team Zuar, 2021; Orescanin & Hlupic, 2021). This comparison already contributes to building foundational knowledge on how the three storage solutions differ according to several aspects.

	Data Warehouse	Data Lake	Lakehouse
Data Types	Structured data and processed data	Structured, semi-structured and unstructured raw data	Structured, semi-structured, and unstructured, both processed and raw data
Data Format	Closed, proprietary format	Open format	Open format
Purpose	Optimal for data analytics and business intelligence use-cases	Suitable for machine learning and artificial intelligence workloads	Suitable for all use-cases (data analytics, business intelligence, machine learning and artificial intelligence workloads)
Cost	Storage is costly and time-consuming	Storage is cost-effective, fast, and flexible	Storage is cost-effective, fast, and flexible
Users	Business professionals	Business analysts, data scientists, data engineers, and data architects	Everyone in the business environment

Scalability	Scaling might be difficult because of tightly coupled storage and compute	Scaling is easy and cost-effective because of the separation of storage and compute	Scaling is easy and cost-effective
Agility	Less agile, fixed configuration	Highly agile, adjustable configuration	Highly agile, adjustable configuration
Analytics	Reporting, BI, dashboards	Advanced analytics	Suitable for all types of analytics workflows, both advanced analytics and BI
Ease of use	The fixed schema makes data easy to locate, access, and query	Time and effort are required to organize and prepare data for use. Extensive coding is involved	Simple, interfaces are provided that are similar to traditional data warehouses together with in-built AI support
Processing	Schema-on-write	Schema-on-read	Schema-on-write and Schema-on-read
ACID compliance	Records data in an ACID-compliant manner to ensure the highest level of integrity	Non-ACID compliance: updates and deletes are complex operations	ACID-compliant to ensure consistency as multiple parties concurrently read or write data

Table 2: High-level Comparison Between the Three Storage Solutions

4.3.1. Data Warehouse Architecture

In Section 4.2.1. two definitions of the data warehouse were already evaluated, which were formulated by people who laid the foundation of the data warehouse. There we have thoroughly evaluated what the definition provided by Bill Inmon (2002) entails. He provided a quite technical definition, although when looking at a data warehouse from a more enterprise-point-of-view, one could say that a data warehouse is capable of storing and managing data from various sources to gain a single, detailed view of part- or all of the business resources (Gardner, 1998).

Understanding the definition of a data warehouse is essential to grasp how the architecture is constructed. As already mentioned in the introduction of Section 4.3., the concept of data warehouses dates back to the 1980s when two IBM researchers developed the business data warehouse. From then on, the data warehouse has been examined to the fullest, and much research has been conducted that has led to making the data warehouse a robust and stable storage architecture. There were, however, numerous visualizations of how the data warehouse is constructed. To obtain a general understanding of the main components of a data warehouse, we have chosen to look into three different visualizations. The first is taken from an article that was published in a Business Journal (Al-Okaily et al., 2022). The second architecture is taken from B. Inmon et al., (2021). Lastly, we evaluated an architecture published on a blog dedicated to technologists by Lavrentyeva & Sherstnev, (2022).

When looking at the three architectures presented in Figure 13, the first observation that was made is that the arrow directions are not consistent with each other. The top-left architecture displays the arrows from bottom to top, visa versa, and left to right. The second picture has only two arrows that are pointed from top to bottom, and the third picture shows the arrows going from left to right. Another observation that can immediately be made is that the second and third architectures clearly have a distinguishment between the three layers. Those layers are even named in the third layer with the bottom, middle, and top tiers. Since the design of the three architectures is not very consistent, we will evaluate each architecture individually.

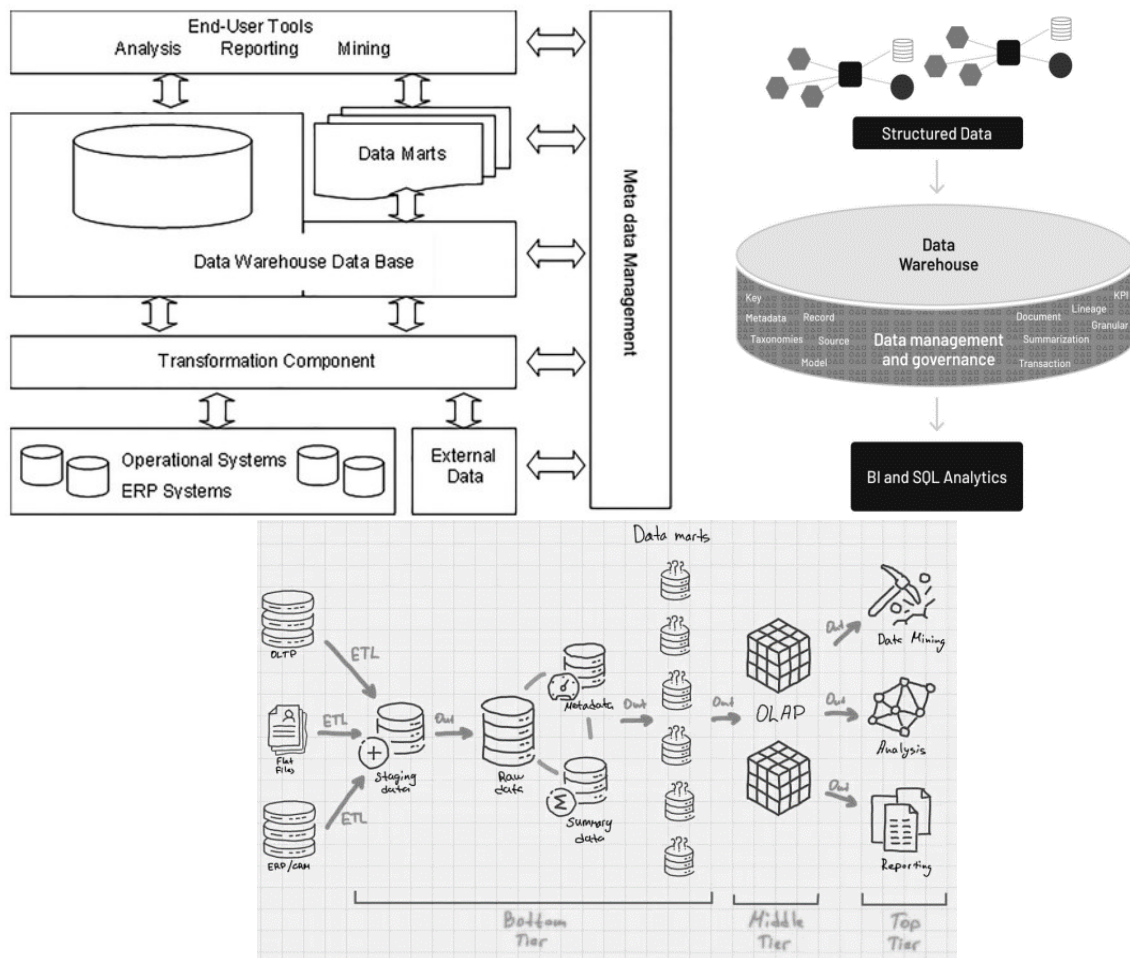


Figure 13: Data Warehouse Architectures (Al-Okaily et al., 2022; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)

The first architecture should be read from bottom to top, where it starts with a layer of data sources. Given that a data warehouse was also used to gain insights from internal data, one box is dedicated to operational systems and ERP systems that generate data within the business. In addition to that, data from external sources can also be ingested into the data warehouse. Before the data is stored in the data warehouse, there is a transformation component where the data lands. This component is dedicated to cleaning and transforming the data to a specific structure since a data warehouse applies a schema to all stored data. That is how they manage to store only structured data. When the data is ready, it will be stored in the data warehouse database. From here, there are two possibilities, either the end-user directly obtains the data that is stored in the data warehouse, or it is moved and stored in a data mart. A data mart is a small and simple form of a data warehouse that stores data related to a specific department or subject. A data warehouse can consist of multiple data marts, given that this is a well-structured way of storing the data. The risk here is that data becomes isolated because data marts are stand-alone entities. Finally, the layer on top is dedicated to the possible end-users that desire to utilize the data that is stored in the data warehouse. Then there is one vertical layer, the metadata management layer, and this is a layer that is a specific trait of the data warehouse. Due to the metadata management layer, the data warehouses can efficiently incorporate data governance practices.

The second picture projects a very minimalistic architecture of the data warehouse. It only consists of three layers: 1) data source, 2) data warehouse, and 3) use-cases. Evidently, the first layer represents the data sources that are suitable for a data warehouse. This happens to be only structured data, given that it needs to adhere to some structural scheme to be stored in the data warehouse. Then the second layer is the most interesting part of the architecture. Here, it can be observed that the data warehouse contains a 'layer' called the data management and governance layer. Here, the features and

characteristics of these layers are shown, but it is almost hidden. The architecture lacks some details and process flows. Although, a few terms that are shown in the data management and governance layer are explained in the source. For instance, metadata is defined as a guide to what data is stored and where it is located; a data model is an abstraction of the data found in the data warehouse; data lineage is a feature that can display the entire flow of transformations that has been done on the data; ETL stands for extract – transform – load which is a data integration process that combines data from multiple sources into a single and consistent data store. The last layer implies that the use-cases that would utilize a data warehouse are by analysts that perform BI and SQL analytics.

The third architecture is a very elaborative architecture explaining different processes that take place in and around a data warehouse. First of all, it should be noted that the architecture is split up into three parts, the bottom tier, middle tier, and top tier. The bottom tier is the layer where data from different sources is loaded and pushed into the data warehouse via ETL processes. In the picture, the data store that is called ‘raw data’ can be seen as the data warehouse where all the data is being stored. Then, multiple smaller data stores are part of the data warehouse. For instance, the metadata store where only data describing the data that is in the warehouse is stored. Another example is the operational data store that acts as a repository where snapshots of the organization’s most current data are stored. Next to that, there are multiple data marts in the data warehouse where data related to a specific subject or business department is categorized and stored accordingly. These are subsets of the data warehouse and serve specific business stakeholders. The second layer, the middle tier, is a service layer where online analytical processing (OLAP) is carried out. OLAP is a computing method reorganising data into a multidimensional format, enabling users to easily and selectively extract and query data. This can then be analysed from different points of view. Finally, the top tier is the layer where all the tools are connected to utilize the data for particular use-cases like data mining, analysis, or reporting. Even though this architecture shows some extra processing steps, it clearly explains how the data warehouse is constructed and works.

In summary, the first architecture consists of 5 layers or components: 1) data source, 2) transformation component, 3) data storage including both a data warehouse database and data marts, 4) end-user tools/use-cases, and 5) metadata management layer. The second architecture is a simplistic representation of the data warehouse, starting with a data source layer, then the data warehouse layer, and finally the use-case layer. Finally, the third architecture is very sophisticated because it is built out of three tiers, and it describes the processes that take place while the data flows through the data warehouse.

4.3.2. Data Lake Architecture

For this research, a definition of a data lake was already introduced in Section 1.1. However, given the scope of the thesis, the following explicit explanation of a data lake will also be taken into account due to its architectural view on a data lake: A data lake uses a flat architecture to store data in its raw format and also support the storage of cleansed and transformed data. “Each data entity in the lake is associated with a unique identifier and a set of extended metadata, and consumers can use purpose-built schemas to query relevant data, which will result in a smaller set of data that can be analysed to help answer a consumer’s question (Alrehamy & Walker, 2015).” Now, the data lake “is trying to challenge the reliable, traditional data warehouses for storing heterogeneous complex data (Khine & Wang, 2018).” All the data that an organization wants to ingest, will be stored in a data lake in their original format. As a result, “complex pre-processing and transformation of loading data into data warehouses are eliminated, and the upfront data ingestion costs are reduced (Khine & Wang, 2018).” Once the data is stored, the data is made available for anyone from the organization who is authorized to perform analyses on the data. This definition contributes to enhancing the understanding of what and how the data lake architecture is constructed. Again, we evaluated three architectures that were found during our literature search, shown in Figure 14.

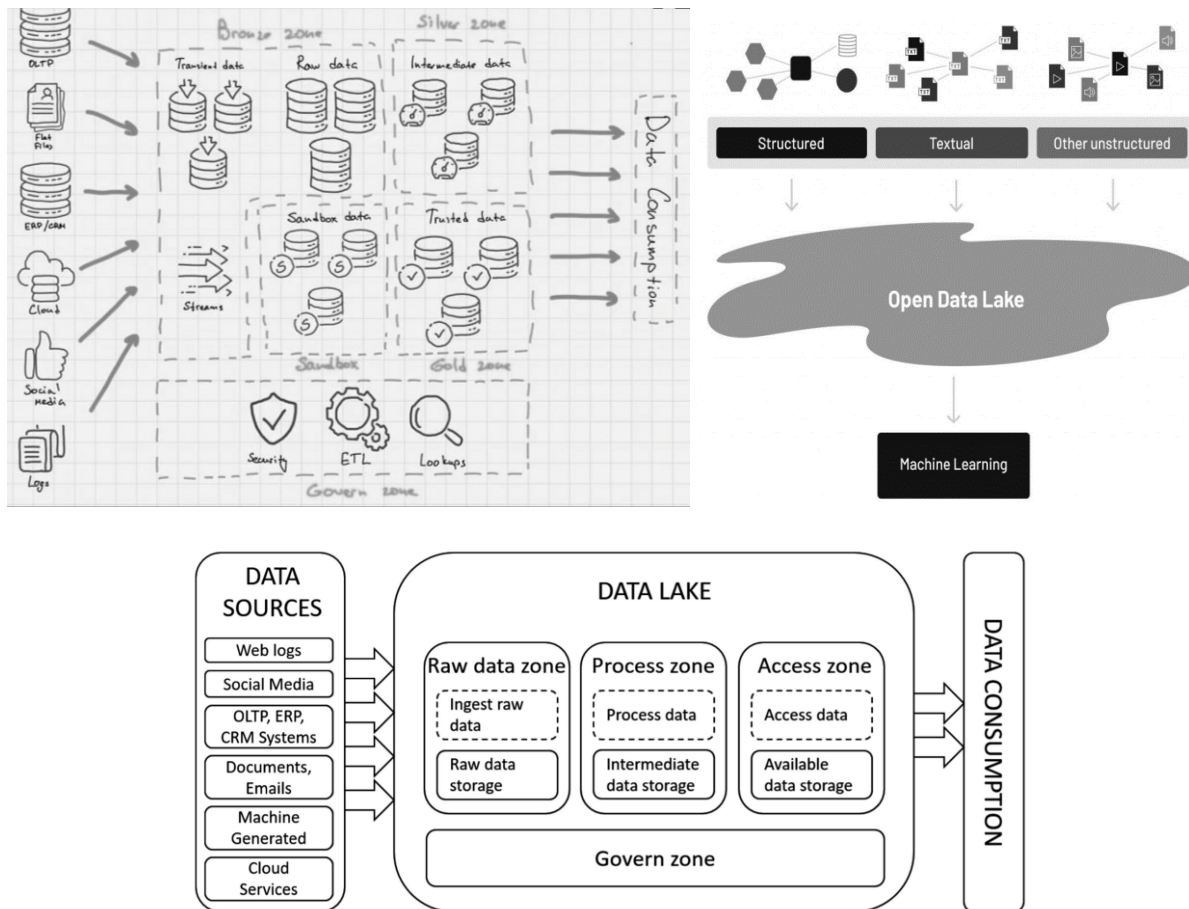


Figure 14: Data Lake Architectures (Lavrentyeva & Sherstnev, 2022; B. Inmon et al., 2021; Ravat & Zhao, 2019)

It can be immediately observed that the top-right picture is a very minimal representation of the data lake compared to the top-left and bottom pictures. Despite the more in-depth architectures of pictures 1 and 3, it can be observed that the architecture of the data lake consists of 3 main layers: data sources, data lake, and data consumption. Next to that, the flow of arrow directions goes from left to right or from top to bottom in the second picture. However, this does not seem too complex, and all architectures are easy to follow.

When looking at the first layer, all three architectures emphasize the fact that a data lake is capable of supporting every data file type that exists. The simplest way of including all possible data formats is by saying that structured, semi-structured, and unstructured data are all allowed in the data lake. Although, it is helpful to create an idea of what examples can be given for either structured, semi-structured, or unstructured data.

The second layer is the data lake itself which, as we learned from the definition by Alrehamy & Walker (2015), has a flat architecture to store the data in its raw format. This is demonstrated by the second architecture, where simply a lake is pictured without having complex processes upfront to store the data according to a specific schema. However, the first and third architecture go more into depth concerning what is actually happening in the lake. This makes this architecture a functional architecture instead of a purely technical-focused architecture (Ravat & Zhao, 2019). The reason for this is that a functional architecture can be implemented by different technical solutions. This makes their architecture unbiased and very suitable for analysis in this research. They presented three distinct data zones and a governing zone. When comparing this structure to the first architecture, it can be observed that the first architecture has an extra data zone and has named the zones differently: bronze zone, silver zone, gold zone, and sandbox zone.

The first zone in both architectures, the ‘bronze zone’ and the ‘raw data zone’, is meant for all the data that is ingested into the data lake. The data is ingested in their raw state without being processed first and can be done in batches, real-time, or hybrid. Utilizing data in their raw format is especially desired by data engineers and data scientists who perform advanced analyses such as predictive analytics or streaming workloads. The second zone is again meant for the same purpose in both architectures. The ‘process zone’ and the ‘silver zone’ is utilized for storing data that has been transformed or enriched according to particular business needs. This is, for instance, done by using transformations like joins, aggregations, selection, etc., and both batch and real-time processing are supported. This zone can be used for specific data analytics that requires the data to be structured to some extent. Typically, the data engineers and data scientists are the ones who access the data in this zone. However, business analysts could also utilize this data for some purposes. The third zone, ‘access zone’ and ‘gold zone’, is where well-structured data is stored. This data is easily accessible and is especially useful for BI tools and reporting. However, machine learning algorithms can evidently also utilize the data that is stored here. Hence, all the stakeholders in the business environment can utilize this data for different purposes. Then, the extra zone that is modelled in the first architecture, the ‘sandbox zone’, is “where data can be experimented with for assumption validation and tests. It is implemented either as a completely separate database for Hadoop or other NoSQL technologies or as part of the gold zone (Lavrentyeva & Sherstnev, 2022)”. As the developers of this architecture imply themselves, this sandbox zone could also be integrated into the gold zone. It depends on the client whether this extra zone is desired or not. In conclusion to this evaluation, the data lake typically consists of 3 zones that store data in different states. So the first state is the raw format, the second state can be described as a processed and enriched state, and the final state is the very well-structured data.

Next to the data zones, there is also a govern zone in the data lake. This zone is responsible for “ensuring data security, data quality, data life-cycle, data access and metadata management (Ravat & Zhao, 2019)”. This should be applied to all zones to avoid the data lake becoming a data swamp where corrupted data is stored and a lot of redundant data that will never be used. It is therefore essential to have an efficient data management strategy and implement at least some essential controls to enforce data validation and quality. Unfortunately, this is a true challenge for many data lake projects, regardless of the maturity of a data team. The fact remains that the architecture has designed a govern zone and is, therefore, one of the essential components of the data lake.

Lastly, the final layer of the architecture is the data consumption layer. In the second picture, there is only an arrow pointed to machine learning use-cases. This is too limited because the data lake can support all use-cases given that it stores unstructured, semi-structured, and structured data. However, it is, in fact, true that the data lake is popular for supporting machine learning and data science use-cases, given that this was lacking in a data warehouse. Therefore, often when the end-user desires to perform BI analysis and reports on structured data, a data warehouse is recommended because a data warehouse is simply optimized for these use-cases. And when the customer desires to do more advanced analytics, a data lake is recommended. However, we still think that only putting ‘machine learning’ as a use-case is too narrow-minded.

To summarize our findings, after evaluating three architectures from three distinct sources, we have concluded that a data lake architecture typically consists of 3 layers: the data source layer, the data lake layer, and the data consumption layer. The first layer represents all the different data formats a data lake can ingest. The second layer consists of different data zones where data is stored in different states. And finally, the third layer is dedicated to all the different use-cases for which a data lake is suitable.

4.3.3. Data Lakehouse Architecture

In literature, one paper was found where the architecture of the data lakehouse was presented (Armbrust et al., 2021) and is shown in the top-left. The top-right architecture was published by B. Inmon et al. (2021), and the architecture at the bottom was published by Lavrentyeva & Sherstnev (2022). In this order, the architectures are presented from left to right to the bottom in Figure 15.

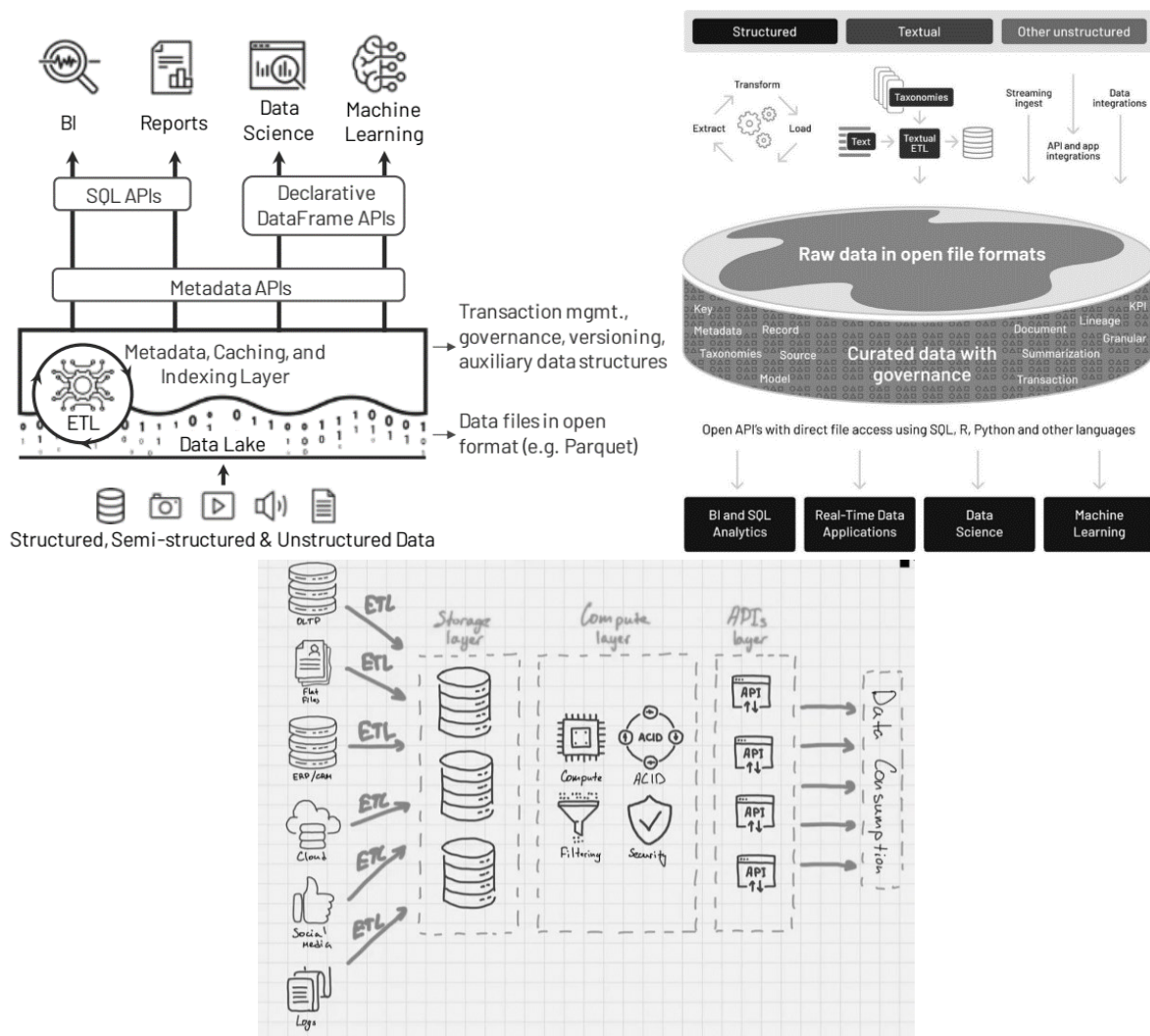


Figure 15: Data Lakehouse Architectures (Armbrust et al., 2021; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)

The first observation made when evaluating these three architectures is the direction in which the arrows are pointing. The first architecture goes from bottom to top, the second architecture flows from top to bottom, and the last architecture flows from left to right. Even though the essence remains the same, it is important to evaluate the reasoning behind it, given that this research will construct an architecture based on the literature findings and the knowledge of the experts that were interviewed. When diving into the existing architectures, we observe that the lakehouse architecture typically consists of 5 constructs: 1) data sources, 2) data storage, 3) compute layer, 4) APIs and 5) use-cases for data consumption. The first construct is dedicated to the different types of data sources. The data lakehouse can ingest, store, and process all different kinds of data. The pictures show that this can be structured, semi-structured, and unstructured.

Then the following construct in the architecture, when following the arrows, a data storage object is presented in the architecture. The first two representations of the lakehouse architecture specifically indicate that the data is stored in a data lake. This is done to emphasize that the different data types can be stored in an open-format storage object, also known as the data lake. The third visualization only visualizes that there is a storage layer in which the data is being ingested. However, from this layer, it is not clear what kind of storage object is being utilized. It also makes it unclear whether the data formats can remain the same or whether they are transformed into some scheme when they enter the storage layer. Although nothing implicates that anything is changed to the data, the way it is visualized in the third architecture can create ambiguities. Whereas the first two illustrations used a data lake to show on the one hand that all different data types can be stored, and second of all to prove how the data lakehouse is utilizing aspects of the data lake.

After the storage object, all three pictures name the following construct differently. The first illustration mentions that there is a metadata, caching, and indexing layer. This layer also enables ETL processes, and the layer is built on top of the storage object. The second illustration calls the layer on top of the storage object the curated data with the governance layer. Whereas the last illustration simply calls it the compute layer. Despite the different terms and attributes shown in the picture, the goal of this layer is generally the same. The architectures intend to show that the lakehouse is capable of all different types of processing and computing techniques. Additionally, governance and security practices are also taken into account and implemented here. Each picture names different examples like transaction management, governance, versioning, lineage, metadata, filtering, security, etc. However, the sources from which the architectures are taken explain that there are a few more functionalities to these layers than is portrayed in the picture. The reason why some are in there and others are not is partly due to the fact that this is the layer in which data warehouse features are present. That is why certain features were chosen to be included to highlight this aspect. In addition to that, there is also some uniqueness to the data lakehouse. It is not just a combination of the data lake and data warehouse. This is also what is attempted to be captured in the architecture. However, this is not very obvious. This is probably the most challenging layer to design since careful thought must be given to what features are included and how the layer will be named. In the previous section, we have dived into the advantages and disadvantages of a data lakehouse. Hence, when a data lakehouse architecture is designed for this research, we will carefully think of what will be included in this layer and how it will be called.

Then, second to last, all the architectures show that there is an API layer between the processing layer and the data consumer. From the first picture, we can deduce that different types of APIs like SQL APIs, Metadata APIs, and Declarative DataFrame APIs can be used to utilize data from the data lakehouse for different use-cases. The second picture highlights the fact that different computing languages like SQL, R, Python can be used to access the data with APIs. The last picture does not explicitly show that different types of APIs can be used to access and utilize the data. Still, it has a layer dedicated to APIs, conveying the message that multiple APIs can be connected. All in all, the API layer seems to be a fundamental layer in the architecture of a data lakehouse.

Finally, all the architectures imply that the data is consumed by end-users. Although, the first two architectures provide specific use-cases as examples. The source in which the third architecture is published mentions in the description that data can be used for different workloads from reporting to BI, data science, or machine learning. Hence, the same use-cases are mentioned as the ones that are shown in the first two architectures. This is also a unique selling point of the data lakehouse, namely that it can support the delivery of BI and reports and the more modern workloads like advanced analytics with data science and machine learning.

To summarize, all three architectures contain the same components, although it has been presented slightly differently for each architecture. Nonetheless, it became very clear that the architecture of a data lakehouse consists of 5 constructs: 1) data sources, 2) data storage, 3) compute layer, 4) APIs and 5) use-cases for data consumption. Moreover, we have observed that the data lakehouse combines the data warehouse and the data lake. This is shown by the fact that the storage layer of the data lakehouse is a data lake. And the metadata, caching, and indexing layer (Figure 15 – top-left), curated data with governance layer (Figure 15, top-right), and the computing layer (Figure 15 - bottom) represent features of the data warehouse.

5. Evaluation

This chapter describes the insights that were obtained from the expert interviews. First, in Section 5.1. background information is provided on each respondent that participated in the interview. Afterwards, a concise summary is provided that describes the main insights obtained from the interview. Next, Section 5.2 provides a redefined design of the artefact in which insights from practice are incorporated. Finally, Section 5.3 provides a conceptual architecture that describes the core components of a lakehouse architecture.

5.1. Interview Results on Data Storage Evolution Model

For this study, five experts were interviewed to gather insights from practice. Table 15 gives an overview of how long each interview lasted, the respondents' current role within Avanade and their years of experience in their current role and in the IT field.

Respondent	Length	Role	Years of Experience in Current Position	Total Years of Experience in the IT Field
1	01:20:06	Senior Consultant, Data Engineering	2 years and 9 months	10 years and 3 months
2	02:06:17	Senior Technologist, Analytics Architect	1 year and 6 months	15 years
3	01:32:55	Consultant, Data Engineering	1 year and 4 months	6 years and 1 month
4	01:54:18	Manager, Data Engineering	1 year and 6 months	16 years and 3 months
5	00:47:20	Group Manager, Analytics Architect	6 years and 11 months	14 years and 4 months

Table 15: Overview of Interview Respondents' Profile

The first part of the interview was focused on the conceptual model that was designed based on the knowledge gained from the literature. Each entity and its attributes and the entire model were discussed.

Table 16 presents an overview where each row represents an advantage or disadvantage of a storage architecture, and the columns represent the respondents. The advantages and disadvantages are represented in numbered assumptions, and the corresponding (dis)advantages can be found in Tables 8 to 13. For each respondent, an indication is given on whether they accepted the assumption, adjusted it, rejected it, or indicated that it depends on certain factors. In the following sections, the reason why some assumptions were rejected is given, and explanations are given when a respondent suggested adjusting the formulation or when it depends on a specific factor or a set of factors.

		Respondent 1	Respondent 2	Respondent 3	Respondent 4	Respondent 5
DW++	A1	Depends	Accepted	Accepted	Adjusted	Accepted
	A2	Depends	Accepted	Accepted	Accepted	Accepted
	A3	Rejected	Rejected	Accepted	Rejected	Accepted
	A4	Depends	Accepted	Accepted	Accepted	Accepted
	A5	Depends	Accepted	Accepted	Accepted	Accepted
	A6	Accepted	Depends	Accepted	Depends	Accepted
	A7	Accepted	Accepted	Accepted	Depends	Accepted
	A8	Accepted	Accepted	Accepted	Accepted	Accepted
	A9	Accepted	Depends	Accepted	Depends	Accepted
	A10	Accepted	Accepted	Accepted	Accepted	Accepted
DW--	A11	Accepted	Adjusted	Adjusted	Accepted	Accepted
	A12	Accepted	Accepted	Accepted	Accepted	Accepted
	A13	Accepted	Accepted	Accepted	Accepted	Accepted

DL ++	A14	Accepted	Accepted	Accepted	Accepted	Accepted
	A15	Depends	Depends	Accepted	Accepted	Accepted
	A16	Accepted	Accepted	Accepted	Accepted	Accepted
	A17	Accepted	Depends	Accepted	Rejected	Depends
	A18	Accepted	Accepted	Accepted	Accepted	Accepted
DL --	A19	Accepted	Accepted	Accepted	Accepted	Adjusted
	A20	Depends	Adjusted	Rejected	Adjusted	Adjusted
	A21	Accepted	Accepted	Accepted	Accepted	Accepted
	A22	Rejected	Depends	Rejected	Accepted	Accepted
	A23	Accepted	Depends	Adjusted	Accepted	Accepted
LH ++	A24	Accepted	Accepted	Accepted	Accepted	Accepted
	A25	Accepted	Accepted	Accepted	Accepted	Accepted
	A26	Accepted	Accepted	Accepted	Accepted	Accepted
	A27	Adjusted	Accepted	Accepted	Rejected	Accepted
	A28	Accepted	Accepted	Accepted	Accepted	Accepted
	A29	Depends	Accepted	Accepted	Accepted	Accepted
	A30	Accepted	Accepted	Accepted	Accepted	Accepted
	A31	Accepted	Accepted	Accepted	Accepted	Accepted
	A32	Accepted	Accepted	Accepted	Depends	Depends
	A33	Accepted	Depends	Accepted	Adjusted	Rejected
	A34	Accepted	Accepted	Accepted	Accepted	Accepted
	A35	Accepted	Accepted	Accepted	Depends	Accepted
	A36	Depends	Accepted	Accepted	Accepted	Accepted
LH --	A37	Accepted	Adjusted	Adjusted	Adjusted	Accepted
	A38	Accepted	Accepted	Accepted	Accepted	Accepted
	A39	Rejected	Accepted	Rejected	Rejected	Accepted
	A40	Accepted	Accepted	Accepted	Accepted	Accepted

Table 16: Overview of Viewpoints of Each Respondent for Every Assumption

The following sections will summarize the main findings for each storage architecture individually. The most interesting discoveries are related to the rejection or adjustment of an assumption. However, some in-depth knowledge of why certain assumptions were accepted or the advantage or disadvantage depends on a particular factor will also be explained. Given that the sample size consists of an odd number, we have chosen only to reject an assumption when three or more respondents have chosen to reject it. As for the adjustment, depending on the suggestion, one respondent is enough to adjust an assumption because we believe that it will increase the clarity of the model. As for the assumption where a respondent answered with “depends”, an explanation will be given why and on what it depends. This does not necessarily mean that the assumption will be adjusted or left out, especially not when the majority has accepted this assumption. However, their arguments will be carefully considered.

5.1.1. Insights Gathered on the Data warehouse Entity

Regarding the number of rejections, only one assumption was rejected by three respondents. This assumption states that an advantage of the data warehouse is that it provides more and better information. All three respondents had in common that they did not understand why this would be an advantage of a data warehouse. After clarifying why it is included in the model, they still rejected this advantage. One of the arguments was that more information does not necessarily mean it is better. Another respondent replied that more information does not say anything about the quality of the information; hence you cannot argue it is ‘better’. Lastly, one argued that it depends on the perspective you take. More historical data is required for some use cases, and a data warehouse can provide access to historical data. Nonetheless, just having a data warehouse does not lead to more and better information. Moreover, the data quality is good because of all the upfront work that must be done before the data is ingested and stored in the data warehouse. When this process is poorly done, the quality of

the data will not be sufficient. Hence, this advantage is not seen as an advantage of the data warehouse itself. Due to the fact that it has been rejected by three experts, and given their arguments for this, the assumption is rejected, and the advantage is removed from the model.

Next to one rejection, respondents answered ‘depends on...’ or ‘adjusted’ for eight assumptions. First of all, for A1, there was one adjustment suggestion and one respondent answered that this advantage depends on a specific factor. This assumption states that the data warehouse delivers and enhances business intelligence. Respondent 1 replied that this entirely depends on the user and is therefore not per se an advantage of the data warehouse itself. In essence, it is not the data warehouse that creates and delivers business intelligence out of nowhere. The user is responsible for building the right models and using the right queries to obtain business intelligence from the data stored in the data warehouse. The data stored in a data warehouse is popular for creating reports and business analyses, thus obtaining business intelligence from the data. But the result is still dependent on the user. The answer given by Respondent 3 is actually complementary to what Respondent 1 already had said. The suggestion that Respondent 3 gave was to rephrase this advantage as “a data warehouse enhances *the ability* to gather more business intelligence”. Although Respondent 3 was the only one suggesting adjusting the phrasing, this adjustment has been accepted since it only leads to more clarity and does not contradict the respondents who accepted the assumption.

Furthermore, according to Respondent 1, assumptions A2, A4, and A5 were dependent on another variable. A2 states that the data warehouse supports and enhances decision-making processes, A4 says that the data warehouse helps improve business processes, and lastly, A5 claims that data warehouses support the achievement of business objectives. Respondent 1 indicated that these are all dependent advantages because they are all related to the users. Everything relies on the insights that are obtained from the data that is stored in the data warehouse. However, the capabilities of the users are the determining factor for enhancing the decision-making processes, improving business processes, and achieving business objectives. Despite the argument that it depends on the user's capabilities, the respondent still agreed with the fact that it should be included in the model. Since the other respondents have all accepted these three assumptions, the advantages are unchanged and still included in the model.

Respondents 2 and 4 indicated that assumption A6 also depends on a specific factor. This assumption states that the data warehouse supports ACID transactions. Respondent 2 indicated that it is indeed allowed by the technology. However, it is actually not followed because performance has to be sacrificed to have ACID transactions. According to the experience of Respondent 2, this function is often disabled to run logs faster. Additionally, ACID transactions are extremely memory intensive. Thus, because of these two reasons, this function is often disabled. However, it is essential to point out that this is only the case when working with big data. Thus, when working with small data, the ACID transactions are a big advantage of a data warehouse. However, the data warehouses are often not ACID compliant when working with big data due to slower performance. Respondent 4 also mentioned that this advantage depends on how the underlying store is configured, but ACID transactions are generally supported. Hence, the concluding answer was that this advantage depends on the configuration, but the advantage should still be included in the model since it is generally supported. This is an interesting viewpoint that none of the other respondents shared. Respondent 1 even indicated that the main point of utilizing a data warehouse, in general, is because of having ACID transactions. Concludingly, the viewpoints shared by Respondents 2 and 4 should be considered when choosing a data warehouse. Still, since this is one of the outstanding capabilities of a data warehouse, this advantage is definitely included in the model.

Next to that, Respondent 4 answered with ‘depends’ for assumption A7, which states that the data warehouse has metadata management mechanisms in place. According to Respondent 4, the meta-data management mechanisms could actually become a downside of the data warehouse due to how the users perceived this. Having these mechanisms in terms of data governance ownership and the ability to monitor data quality is very beneficial. However, when this is taken too seriously by the users that interact with the data warehouse, it can quickly turn into a downside. This is because you typically have different departments that have ownership over their data. This data is brought into the data warehouse,

however, sometimes, the data from different departments are combined for a specific use case. The issue that now arises is who is the owner of this data. In the older days, there were not really proper tooling and conversations about who owns the data and who is the right person to monitor and keep up the data quality. Regardless of the fact that this was a common problem, Respondent 4 recognises that it is advantageous to have metadata management mechanisms in place to regulate the data governance and has outlined who the data owners are and who is responsible for the quality of the data. This is because when issues occur with the data, someone is responsible for solving them. This is important to consider, however, Respondent 4 still accepted that this is an advantage of the data warehouse. Besides, Respondent 3 replied that the metadata management is very well organized according to his experience. On top of that, Respondent 1 focused more on the metadata part and not so well on the governing part, and he shared that metadata management in a data warehouse is very important because it is beneficial to have information on what kind of data is actually stored. Concludingly, this advantage remains in the model. However, careful thought is given to mentioning data governance as a separate attribute of the data warehouse.

Then, for assumption A9, two respondents said this advantage depends on the configuration of a data warehouse. The assumption is that an advantage of the data warehouse is that it allows versioning and access to historical data. According to Respondent 2, this advantage is again dependent on the configuration of the data warehouse. This is caused by the fact that having versioning all over the place will lead to performance issues. In general, there is some versioning on the highest level, but this is not out-of-the-box. You have to build them yourself. This is usually not done since the performance is then sacrificed. This is in line with the response of Respondent 4, which also included the fact that it depends on how the data warehouse is implemented. If you do not design your data warehouse appropriately, versions of the data will not be saved, and the previous version will be lost when the data is being updated. Hence, it is not a specific trait however, when it is implemented, it can be seen as an advantage. All in all, this is a feature that is supported by the data warehouse. It only depends on the implementation whether it is included or not. Still, it remains in the model since it is an advantage when implemented, and it is not really there in the data lake or data lakehouse.

Finally, two respondents suggest adjusting assumption A11 for more clarity. As of now, the assumption states that a disadvantage of the data warehouse is that it lacks flexibility. While three respondents fully agreed with this, two respondents indicated that this disadvantage could be better formulated. Both respondents explained that the lack of flexibility could be divided into two different aspects. First of all, a data warehouse is perceived to be lacking flexibility since it only supports structured data. Second of all, the lack of flexibility can also be expressed in terms of not being easy to upscale or out scale. In fact, scalability is not even attainable due to the fact that storage is linked to the processing part of the data warehouse. This makes it very complex to scale up. Therefore, the suggestion is to have two disadvantages one focusing on the inflexibility with regards to different data types and one focusing on the scalability aspect. Respondent 3 even added that the data format in the storage is proprietary. In other words, it is not open-sourced, which can also be seen as a disadvantage of the data warehouse. The other three respondents accepted the assumption because they are aware of the reasons why a data warehouse is inflexible. However, we agree that this disadvantage is better clarified when splitting it up into distinct disadvantages explaining in what regard the data warehouse is inflexible. Therefore, A11 will be removed from the model. Instead, the following assumptions are added: 1) data warehouses lack flexibility since they only support structured data, 2) data warehouses lack flexibility because it is difficult to upscale and out scale, and 3) data warehouses lack flexibility because the data format is proprietary.

Concludingly, a total of 4 out of 13 assumptions that were formulated about the data warehouse were accepted by all respondents. Hence, they remained in the model. Based on the experts' arguments, assumption A3 was removed from the model, assumption A1 was reformulated, and finally, assumption A11 was removed and replaced by three new assumptions. A descriptive overview of the exact changes can be found in Table 17. In addition to the descriptive table, an adjusted entity is shown in Figure 16.

Assumption	Action	Result
A1: Data warehouses enhance business intelligence	Rephrased	Data warehouses enhance the ability to obtain business intelligence
A3: Data warehouses provide more and better information	Rejected	Removed from model
A11: Data warehouses lack flexibility	Split up	Data warehouses lack flexibility because they only support structured data
		Data warehouses are difficult to upscale and out scale
		Data warehouses lack flexibility because data formats are proprietary

Table 17: Overview of Adjustments of Assumptions Related to Data Warehouses

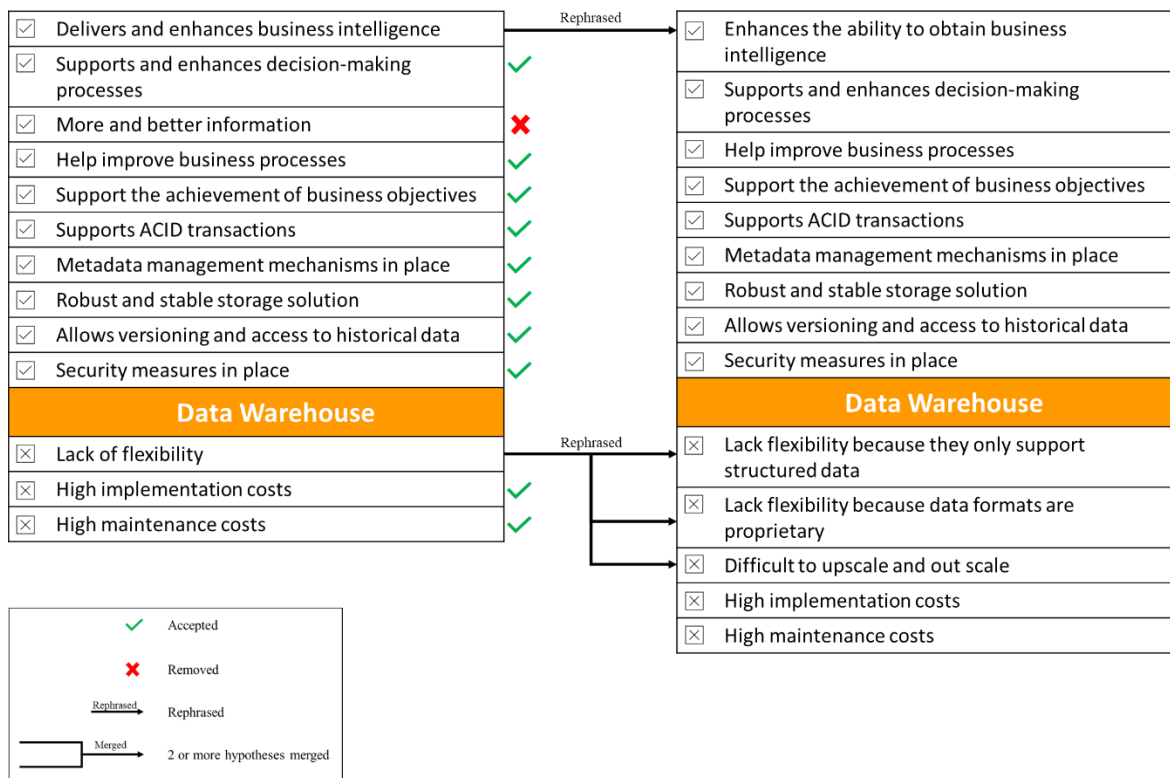


Figure 16: Adjusted Data Warehouse Entity

5.1.2. Insights Gathered on the Data Lake Entity

When looking at the number of rejections, three assumptions have been rejected by at least one respondent. The first is for assumption A17 which states that an advantage of the data lake is that it provides easy accessibility to the data. This advantage is rejected by Respondent 4 due to the fact that accessibility is actually limited since it is difficult for people to find the data they need. Respondents 2 and 5 provided a similar answer and mentioned that this advantage depends on the people who interact with the data lake. The business users will find it hard to find anything in the data lake, whereas the data engineers will know how to work around the lake and for them, it is indeed an advantage that the data is stored the way it is. Given that only one respondent wanted to reject this assumption, two respondents indicated that it depends on the people, and two respondents agreed with this advantage, we have chosen to reformulate this assumption to be more precise. The advantage will now be that data lakes provide easy access to raw data.

Second of all, assumption A20 was rejected by Respondent 3, it was adjusted by Respondents 2, 4, and 5, and the disadvantage depends on a variable according to Respondent 1. Hence, this assumption caused quite some discussion during the interview. The assumption states that the low-security assurance is a disadvantage of a data lake. This was rejected by Respondent 3 because there are quite some options to implement security measures. First of all, role assignments can be enforced on the storage accounts in a data lake. Suppose there is a storage account with five containers representing five different owners. In that case, each owner is allowed to do anything to their container but not to the container owned by someone else. Another access mechanism that can be implemented is Access Control Lists (ACLs). When the owner of a data container does not want everyone to have access to the data they have stored in their container, it is possible to specify which individuals have access to the data and who do not. So, according to Respondent 3, the security mechanisms are available however, the implementation part is a bit tricky. He also recognizes that the available security mechanisms are not yet sufficient to say that the data lake is secured very well. Still, he does not see it as a substantial disadvantage and rejects this assumption. Whereas Respondent 3 wanted to reject this assumption, Respondents 2, 4, and 5 advised adjusting the formulation of the disadvantage. They all explained that you could have security rules in a data lake on object levels but not record levels. This means that an individual either has access to the table or file or does not. When looking at a data warehouse, you can have record-level security, which makes the security more fine-grained. When a user is not allowed to see a specific column or row, it is possible to specify this in the access rules. Lastly, Respondent 1 did not reject nor accept the assumption fully. Instead, the respondent indicated that it depends on how you implement your data lake. If there is only one data container where all the data is stored, then security assurance is low because everyone will have access to all the data. However, when the data is grouped into multiple containers, it is possible to deny access to specific individuals. In essence, all respondents shared the same facts on which security mechanisms are available for a data lake. They just did not all agree on the wording of the assumption. Consequently, the suggestion given by the respondents who wanted to adjust the assumption has been taken on. Therefore, the assumption will be rephrased from ‘low-security assurance’ to ‘data lakes do not provide fine-grained security’.

The third and last assumption that at least one respondent rejected is assumption A22. This assumption states that a data lake has a poor level of quality and reliability of the data. This was rejected by Respondents 1 and 3, and it depends on a specific factor according to Respondent 2. All three respondents stated that the quality and reliability are poor when the data lake has turned into a data swamp. Since this is already included in the model, they argued that this assumption about poor quality and reliability could be rejected. Respondent 2 argued that it depends on how the users utilize the data lake. For a data warehouse, it is even possible to have data of poor quality or reliability. However, due to transaction support, this can be limited. Unfortunately, data lakes do not support transaction support, making it very challenging to manage the lake and keep up the quality and reliability. Therefore, Respondent 2 would argue that it is indeed possible that data can be of poor quality and unreliable. However, it depends on how well the users are managing it. Because of the arguments provided by these respondents, we have chosen to merge assumptions A21 and A22. This leads to one assumption stating that ‘data lakes have a risk of turning into a data swamp which means poor quality and unreliable data’.

Then there were a few assumptions for which an adjustment was recommended, or some explanation was given on why this assumption is not naturally valid. For assumption A15, two respondents indicated that this is not an obvious advantage of the data lake. The assumption states that a data lake is a cost-effective storage solution. Both respondents mentioned that a data lake is a cost-effective storage solution, but it depends on your object storage. It is possible to make even a data lake very expensive if advanced implementations are chosen, and expensive storage accounts are chosen. For instance, Blob Storage in Azure is cheap and works well. Though it is not instant, so it would not be suitable, e.g., for fast-trading or computer games, because for that, super-fast storage is required. Hence, there are layers in the level of latency and availability. Also, a data lake is flexible and can be easily scaled up, but that also has a price tag. Hence, this assumption is generally accepted since it is cheaper than a data warehouse on average. However, one should remember that a data lake can be very expensive, depending on the client's desires.

Next to that, Respondent 5 suggested an adjustment for A19, which claims that a disadvantage of the data lake is that it lacks metadata management. The suggestion was to change metadata management to data management to make it broader and capture more aspects that are lacking. Metadata management focuses purely on managing what data is in the data lake and how it should be used. Whereas data management will also include the fact that a data lake, for instance, hardly takes into account GDPR rules and regulations, data governance is a mess, and data lineage is also unavailable. This has been taken into consideration. However, the assumption will remain as it is for a few reasons. First of all, all other four respondents accepted the assumption. Second of all, we believe it would be better to have separate assumptions instead of making this one more general. Lastly, all respondents were asked to think of any assumptions that could be added to the model, which will be discussed in Section 5.2.1, and two respondents mentioned data governance and GDPR rules there. Thus, the argument from Respondent 5 for A19 is taken into account. However, it will be dealt with in the following sub-section.

Lastly, for assumption A23, Respondent 2 indicates that it depends, and Respondent 3 provided an adjustment to the statement. It now says that the performance level of the data lake is inconsistent, which is a disadvantage. This assumption has led to varying answers due to different interpretations. According to Respondent 2, it depends on what the user is actually doing and what processing application is connected to the data lake. Respondent 2 stresses the fact that a data lake is technically only storage. Hence, a compute/processing application or tool is always connected to the data lake. The performance can vary depending on which tool this is and the workloads performed. Thus, the performance can be inconsistent. On the contrary, Respondent 3 argues that the performance level can be inconsistent due to latency. This can be explained by the fact that the provision of resources in Azure is region-specific. For a company that is, for instance, operating in the Netherlands, the data will be stored in data centres located in either northern Europe or western Europe. This is fine because the performance levels across Europe are good. However, performance is affected when a company has departments situated all over the world. If the data is stored in a region in Europe and some department in Australia requests to obtain data, then the performance is affected because the latency comes in. Hence, the conclusion of Respondent 3 is that the latency hampers the performance of a data lake, and he suggests adjusting the assumption to this. Contradicting to what Respondent 3 said about different data types not affecting performance, Respondent 1 states that performance levels are actually affected by the fact that there are different file types. The reason is that different file types use different libraries to work with the data. The performance of reading and doing transformations is quite different when comparing various libraries. To make it even more complex, Respondent 4 explained that the data might be stored so that it is not performant for your specific use case. Imagine three different use-cases that utilize the same set of data. It is most likely that the performance of each use case will be different. All these insights made this assumption quite complex because all respondents agreed with the fact that performance levels are inconsistent when working with a data lake. However, the reasons for why and how these performance levels are affected differed for every respondent. The fact that each respondent provided a different cause only shows that so many factors can influence the performance level. There is not a sole reason why performance levels can be inconsistent. Since all of them accepted the assumption, we decided to leave it as it is. Suppose we were to specify this assumption into four separate assumptions stating that these are the four factors why performance levels are hampered. In that case, we imply that there is no other reason why it can be affected. Moreover, it is likely that if five other experts were asked the same question, they would come up with five more reasons for how the performance level can be affected. Therefore, assumption A23 will remain as it is, giving room to give your interpretation based on your own experiences.

Concludingly, 4 out of 10 assumptions about the data lake was accepted by all respondents and are therefore still included in the model. Given the insights provided by the experts, we have chosen to rephrase assumptions A17 and A20, reject assumption A22, and finally merge assumptions A21 and A22. A descriptive overview of these changes is shown in Table 18, and the adjusted entity of the data lake is shown in Figure 17.

Assumption	Action	Result
A17: Data lakes facilitate easy access to the data	Rephrased	Data lakes provide easy access to raw data
A20: Data lakes provide low-security assurance	Rephrased	Data lakes do not provide fine-grained security
A21: Data lakes have a risk of turning into a data swamp A22: Data lakes have poor quality and reliability of data	Merged	Data lakes can easily become data swamps which result in poor quality and unreliable data

Table 18: Overview of Adjustments of Assumptions Related to Data Lakes

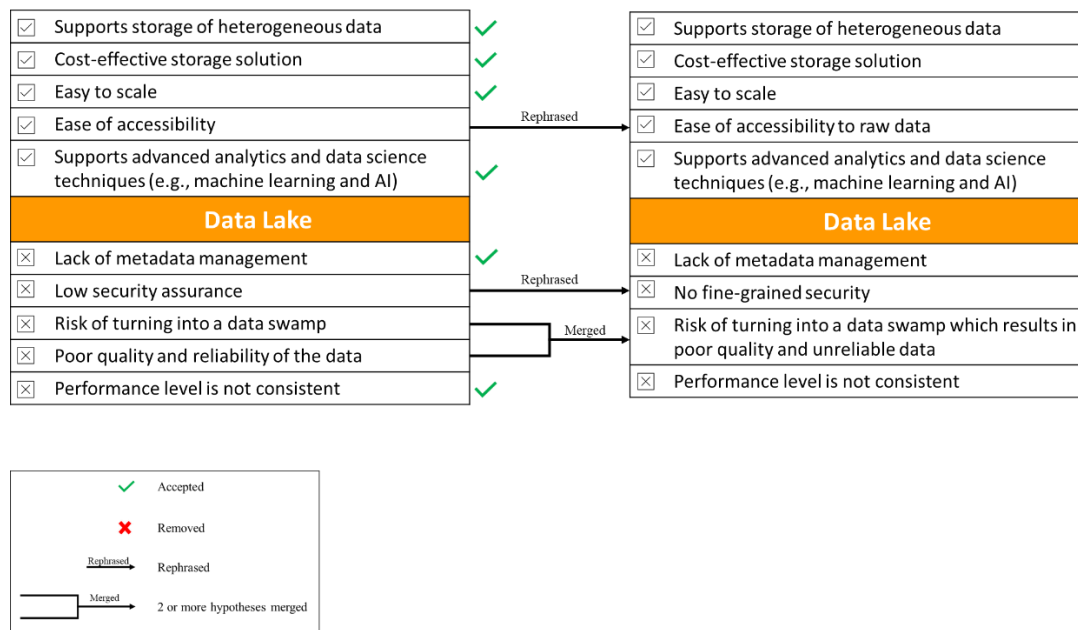


Figure 17: Adjusted Data Lake Entity

5.1.3. Insights Gathered on the Data Lakehouse Entity

There were a few rejections for three assumptions regarding the data lakehouse entity. The first is for assumption A27 which states that a data lakehouse reduces data redundancy. This was rejected by Respondent 4, and Respondent 1 suggested making an adjustment. According to Respondent 1, this benefit only counts when comparing the data lakehouse to a data lake. Whereas the level of data redundancy would be the same for a data lakehouse and data warehouse. This is because the data stored in a data warehouse is structured in such a way that it directly satisfies the desire for a specific analysis or report. Whereas data is being dumped in a data lake by anyone in whatever format. Hence, it is almost inevitable to have data stored that will not be used by anyone. However, who will figure this out and decides to remove the data from the data lake to reduce data redundancy? Nobody will. Therefore, a data lakehouse is indeed reducing data redundancy compared to a data lake but not compared to a data warehouse. This view is supported by Respondent 4, although his viewpoint was even stronger, and Respondent 4 finds it even a weak assumption, and thus it should be rejected according to him. The other respondents accepted this assumption and did not bring any counterargument to why it should not be included. Although this is not the most vital advantage of having a data lakehouse in place, it is right to assume that the data is less redundant than the data in a data lake. Therefore, the assumption will be adjusted to 'data lakehouses are likely to have less data redundancy than data lakes'.

Another assumption that received one rejection is assumption A33. This assumption claims that the data lakehouse provides high-security assurance. However, it has been rejected by Respondent 5, it was adjusted by Respondent 4, and Respondent 2 indicated that it depends. Respondent 5 has rejected this

assumption because, according to his experience, the security level of a lakehouse is just the same as that of a data lake. This means that a lakehouse can also not provide fine-grained security in the same way a data warehouse provides it. Respondent 4 took a milder point of view and argued that there are quite some possibilities in terms of transaction security which is more about the utilization of data but not particularly in accessing the data. Some examples of transaction security include 'rollbacks' and the integration with Azure Active Directory. Rollbacks are a functionality that whenever an operation on the data results in errors and bugs, the operation can be reversed to bring the data back to its previous state. This helps to keep the data integer. The second example, integrating with Azure Active Directory, is a service from Azure where you can regulate and set up a single sign-on, multifactor authentication, and conditional access to be protected against cybersecurity attacks. Despite these security measures, the fact remains that it is not possible to have fine-grained security in terms of denying access on a record level. Hence, this is something that Respondents 4 and 5 have in common. Lastly, Respondent 2 stated that the security level depends on the data lakehouse's implementation. However, a high level of security is not a specific feature of the data lakehouse. It is not there out-of-the-box, and it still has to be implemented. Therefore, Respondent 2 argued that the level of security depends on the implementation but having high-security assurance is not something a data lakehouse can guarantee. Based on all these findings, we conclude that, at this moment, the lakehouse does not provide the same fine-grained security as data warehouses do. Hence, assumption A33 will be changed into a disadvantage stating that the data lakehouse cannot provide fine-grained security.

The last assumption that three respondents rejected is assumption A39. This assumption states that a disadvantage of the data lakehouse is that it requires training to obtain new skills. However, Respondents 1, 3, and 4 disagreed with this claim. According to Respondent 1, new skills are not very much required. Utilizing the data lakehouse will not be too difficult if someone has worked either with a data warehouse or a data lake. However, it is indeed required to understand some new basic principles that are unique to the data lakehouse. This view is shared by Respondents 3 and 4, who also indicated that some new knowledge is required to implement a data lakehouse. Nonetheless, they do not think this means people must gain new skills to understand the data lakehouse concept. On top of that, with every 'new' technology, it is required to understand some principles or new configurations for the implementation. Therefore, the respondents believe this is not a disadvantage of the data lakehouse. Since this is shared by 3 out of 5 experts, this assumption will be removed from the model.

Then there were a few assumptions for which the respondent suggested that the advantage or disadvantage depends on or should be adjusted. First of all, one respondent indicated that assumption A29 was not an obvious advantage. The assumption states that the data lakehouse delivers business intelligence and supports decision-making processes. Respondent 1 argues that this advantage is user dependent. Creating business intelligence is a process that the user should carry out. If this user is not capable of doing this process correctly, then this advantage is not really there. However, yes, it is an advantage that a data lakehouse can be used to obtain business intelligence. Due to the fact that all the other respondents agreed with this assumption, it will still be included in the model.

Secondly, two respondents shared that assumption A32 is also a dependent advantage. The assumption claims that metadata management mechanisms are in place in a data lakehouse. Respondents 4 and 5 imply that this advantage depends on how a data lakehouse is configured. Respondent 4 argues that the metadata management mechanisms are not there by default, and it still has to be implemented. Whereas Respondent 5 shared that, in theory, the data lakehouse enables metadata management mechanisms. However, it is not fully there yet in practice. The other respondents implied this advantage is there, although it would not be the strongest advantage of a data lakehouse. Despite that, it is still seen as an advantage. Therefore, it will remain in the model.

Another dependent advantage is assumption A35, according to Respondent 4. The assumption states that a data lakehouse supports optimization techniques like caching, auxiliary data, and data layout. The reason why Respondent 4 was slightly hesitant about this assumption has to do with the fact that these might be vendor-specific terms to Databricks. The other respondents indicated that all those optimization techniques are possible in a data lakehouse and did not mention anything about it being

vendor specific. Given this concern, we searched for more resources besides the Databricks Whitepaper to see what is said about optimization techniques in a data lakehouse. A blog published on an unbiased website stated that query performance optimizations are probably the biggest challenge of a data lakehouse (Panyapiang, 2022). This is clearly caused by the fact that the data is not rigid and well-structured. It also depends on what latency is acceptable and which resources are available per use-case. Unfortunately, there is no ‘one-size-fits-all’ solution, although it is possible to implement optimization techniques to enhance performance. Another blog, published by AWS, which is also a vendor but different from Databricks, also implied that there are possibilities for implementing query optimizations they only do not share specific examples (Kava & Gong, 2021). The fact remains that first of all, Databricks has introduced the Lakehouse and is the first and only vendor. Second of all, the Lakehouse is still very new, so there is not much published about this architecture and its capabilities. Moreover, it is open-sourced hence, it is possible to personalize your configuration and implement specific optimization techniques. We believe that Databricks has mentioned caching, auxiliary data, and data layout as examples of possible optimization techniques that are supported in a data lakehouse. Therefore, for clarification, we will reformulate the assumption by stating that these are possible techniques.

Furthermore, Respondent 1 indicated that assumption A36 is not a natural advantage of the data lakehouse. The assumption says that a lakehouse supports management and performance features. The response to this statement was that it is valid only when the Databricks implementation is used. If some third-party tool is utilized, then the management and performance features are not automatically supported. All the other respondents agreed with this assumption. However, some indicated that it is not really a strong one since it does not say what management features or performance features include. Nonetheless, this assumption was generally accepted and will remain unchanged in the model.

Then, it was suggested by three respondents to adjust the phrasing of assumption A37. As of now, the formulation of the assumption is that a data lakehouse is not very mature. They believe the data lakehouse architecture is new but not necessarily immature. The idea is very good and well-developed. However, it is still very new and has not been adopted by many. Moreover, according to their knowledge, there are not many downsides to the current implementations of the data lakehouse. Thus, the technology is very well-developed and not at an immature level. Therefore, Respondents 2 and 3 suggested replacing the term ‘not very mature’ with ‘new’. Whereas Respondent 4 implied that the assumption could be rephrased to data lakehouses are still immature in terms of market adoption. We will combine these viewpoints by rephrasing the assumption that “data lakehouses are very new and not fully adopted in the market yet”.

Concludingly, a total of 9 out of 17 assumptions about the data lakehouse were fully accepted by all respondents. Four assumptions about the data lakehouse have been adjusted, and one has been rejected. A descriptive overview of the changes made is shown in Table 19, and the adjustments to the entity of the data lakehouse are shown in Figure 18.

Assumption	Action	Result
A27: Data lakehouses reduce the level of data redundancy	Rephrased	Data lakehouses are likely to have less data redundancy than data lakes
A33: Data lakehouses provide high security assurance	Rejected & Rephrased	Data lakehouses cannot provide fine-grained security
A35: Data lakehouses support optimization techniques like caching, auxiliary data, and data layout	Rephrased	Data lakehouses support optimization techniques (e.g., caching, auxiliary data, and data layout)
A37: Data lakehouses are not very mature	Rephrased	Data lakehouses are still very new and not fully adopted in the market yet
A39: Data lakehouses require training for new skills	Rejected	-

Table 19: Overview of Adjustments of Assumptions Related to the Data Lakehouse

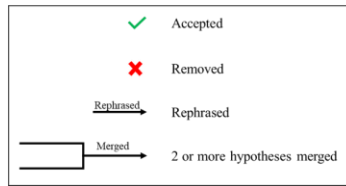
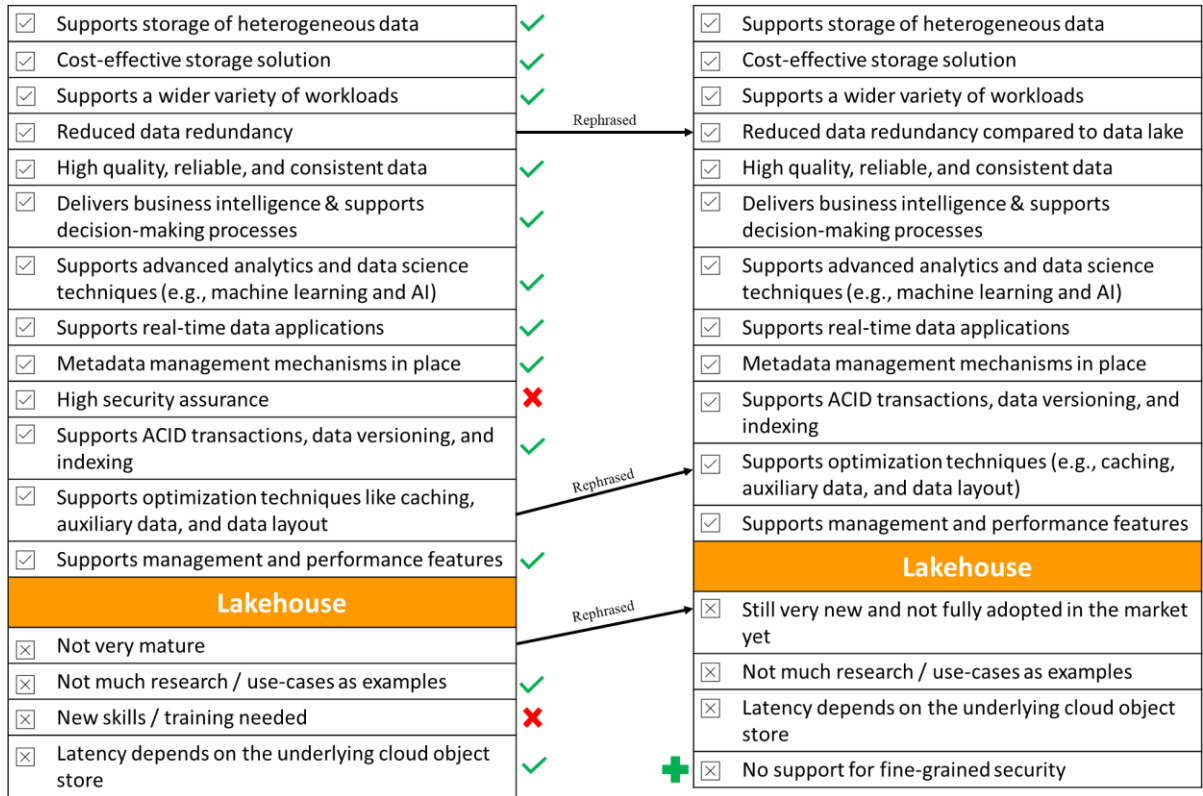


Figure 18: Adjusted Data Lakehouse Entity

5.2. Redefined Data Storage Evolution Model

During all the interviews, numerous suggestions were given to improve the model further. During the interview, the respondents were specifically asked to evaluate the advantages and disadvantages of each entity, whether some advantages or disadvantages could be added, and how the findings from the literature were modelled. Not only was the evaluation focused on entity-level but also model-level. This particularly includes the way in which the entities were put into context and how the relationships between the entities were modelled.

5.2.1. Suggestions of Experts on Entity-Level

The previous sections focussed on whether the respondents accepted, rejected, or adjusted the assumption or argued why it is a dependent advantage or disadvantage. In addition to that, the respondents were also asked to share any advantage or disadvantage that was not included yet and should be added according to their knowledge. The elaborate version of these findings can be found in Appendix C. The following paragraphs will present the suggestions in a compact and reformulated way. Next to presenting the results, an elaboration is provided on why the suggested assumption has been included in the model. The evaluation will be structured by examining what all respondents have to say per entity.

5.2.1.1. Suggestions for Data Warehouses

Most additions that were suggested to be added to the model were related to the data warehouse. In total, six advantages and four disadvantages were suggested to be included in the model that were not there yet. In Table 20, the proposed advantages are presented. The suggestions are structured the same way the assumptions that were obtained from the literature are formulated.

Advantages	Respondents				
	1	2	3	4	5
Data warehouses support foreign keys	✓				
Data warehouses performances are optimized for small data sets		✓			
Data warehouses performances are optimized for structured data			✓		
Data warehouses can show query results to a local session and perform rollbacks			✓		
Data warehouses enable data governance practices				✓	✓
Data warehouses apply quality frameworks to have high-quality data					✓

Table 20: Suggested Advantages for Data Warehouses

According to Respondent 1, a solid advantage of the data warehouse is the ability to implement foreign keys. This is something that cannot be properly implemented in a data lakehouse or data lake, which is why Respondent 1 argues this advantage should be included in the model, given that it is unique to the data warehouse. A foreign key is used to link two tables together by referencing a record from one table and using it in another table. Even though we believe this feature is valuable, we have chosen not to include this feature in the model. This is because it is such a specific technical implementation that might not be relevant for every use case. Also, since only Respondent 1 has shared this advantage, we have chosen not to include this advantage in the model.

Two respondents indicated that the performance of data warehouses is optimized for structured and small data. These two advantages were not considered by our findings in the literature, most likely due to taking on a different perspective on this matter. You could argue that the fact that data warehouses only work well with small or structured data is actually a disadvantage because it indicates that it is not very flexible and cannot handle big data. However, the experts shared that for some use-cases, companies do not have a lot of data or work with a lot of structured data. Evidently, the fact that the performance of data warehouses is excellent for these use-cases is a considerable advantage. Therefore, these two advantages are added to the model.

The fourth advantage was shared by Respondent 3 while discussing an advantage that was included in the model which stated that data warehouses allow versioning and access to historical data. Respondent 3 shared that when you perform some operation that manipulates the data, it is possible to view the result in your local session before it is pushed and updated in the actual database. This feature is called a 'rollback' and is beneficial because it eliminates the chance of pushing bugs and incorrect data to the actual database. Interestingly, another respondent also mentioned this feature when the security of the data was discussed in a data warehouse. However, Respondent 3 believes this feature should be mentioned explicitly, given that this is not supported in a data lake or data lakehouse. Since this is related to versioning data or accessing historical data, Respondent 3 suggested adding this feature or even dedicating a separate assumption to this advantage. Given that this feature is an outstanding capability of the data warehouse and relates to enhancing security, we have chosen to include this feature as a distinct advantage of the data warehouse.

Next, two respondents implied that a data warehouse enables data governance practices which is an advantage of this technology. Data governance has become increasingly important throughout the years, and the experts mentioned that this is still one of the biggest challenges. Before it was discussed what data governance practices are enabled and supported by data warehouses, a definition of data

governance was searched for. The following definition of data governance was found in the literature: “Data governance specifies a cross-functional framework for managing data as a strategic enterprise asset. In doing so, data governance specifies decision rights and accountabilities for an organization’s decision-making about its data. Furthermore, data governance formalizes data policies, standards, and procedures and monitors compliance (Abraham et al., 2019)”. A well-defined data governance framework typically answers questions about “how decisions related to data are made, who makes the decision, who is held accountable, and how the results of decisions are measured and monitored (Rifaie et al., 2009)”. According to Respondent 5, data governance with a data warehouse is pretty well organized since quality frameworks are in place that upholds the principle of managing data as a strategic enterprise asset by maintaining and enhancing its quality. Moreover, Respondent 4 shared that data warehouses are typical “schema-on-write”, meaning the data is transformed into a specific structure before storing it in the data warehouse. Hence, there are some standards and policies on how and what is done to the data. On top of that, in data warehouses, users are generally assigned to groups and each group is granted specific permissions on certain data sets. This helps with knowing who is responsible for what data, who is the owner, and who is held accountable. Lastly, the conceptual model that was created with the insights obtained from literature already included the fact that data warehouses support metadata management practices. This is a very critical element of data governance because it helps with managing the data in general, access control, secure privacy-sensitive data, and data quality management. In conclusion to these insights, the respondents suggested including a distinct advantage that data warehouses can incorporate data governance practices very well.

The last advantage that was a suggested addition to the model was provided by Respondent 5, which states that data warehouses apply quality frameworks to have high-quality data. The respondent shared that having high-quality data in your data warehouse is of utmost importance, given that it is used for business intelligence. Consequently, when developing a data warehouse, ensuring quality is one of the foundations of the requirement analysis. According to the experiences of Respondent 5, this is done by defining and utilizing quality frameworks. As explained, a data warehouse works with a ‘schema-on-write’ method. Hence, before the data is entered into the system, it is already known what the structure of the data is going to be. When this schema-on-write structure also incorporates a quality framework, it is assumed that from the moment the data enters the data warehouse, it adheres to a minimum level of quality standards. Since ensuring quality is so vital for a data warehouse, practices should be implemented from an early stage of the data warehouse development. According to Respondent 5, this is done with these quality frameworks, which is why he argued this should be included in the model as a distinct advantage.

To summarize, based on the experiences and knowledge the experts have shared, the data warehouse entity has grown with five new advantages. They have been added since they were not yet incorporated into the findings from the literature. And given the explanations provided by the experts who have based their views on real-life examples, it was agreed that the additions are valuable advantages of a data warehouse. Thus, the data warehouse entity is now revised and updated, as presented in Figure 19.

Besides the suggested advantages of a data warehouse, some recommendations were also given regarding disadvantages that were not included in the model. An overview is provided in Table 21, where it is shown that four new disadvantages were suggested. A remarkable point of the disadvantages suggested is that three disadvantages are supported by two experts, and one is even supported by all experts. This proves that the proposed disadvantages should be part of the model even more.



Figure 19: Adjusted Data Warehouse Entity 2.0

Disadvantages	Respondents				
	1	2	3	4	5
Data warehouses' storage capacity is linked with the processing capacity	✓			✓	
Data warehouses are very difficult to scale	✓	✓	✓	✓	✓
Data warehouses have proprietary data formats			✓	✓	
Data warehouses are not suitable for modern workloads (e.g., streaming and data science use-cases)			✓	✓	

Table 21: Suggested Disadvantages for Data Warehouses

The first disadvantage was proposed by Respondents 1 and 4. They mentioned that a huge disadvantage of the data warehouse is that the storage capacity is linked to the processing capacity. This is the cause for the second disadvantage that all experts proposed. Because storage and processing capacity are linked to each other, it is very difficult to scale your data warehouse. This is also one of the reasons why the implementation and maintenance costs are so high. Also, Respondent 1 shared that a lot of times, when a data warehouse is scaled up, you do not actually increase the amount of data being processed per day. Still, you only increase the historical data that you have in your storage. However, suppose it is desired to store more historical data and thus expand the storage capacity. In that case, you are also tied to increasing your processing capacity even when this is not desired. We conclude that it is highly important to stress that storage and processing capacity in a data warehouse are linked to each other. This is not the case for data lakes and data lakehouses. Therefore, we have chosen to include this as well in our model.

There was a disadvantage included that stated that data warehouses lack flexibility. In Section 5.1.1. we have discussed this side, and as shown in Table 17, where we present the adjustments made to the existing assumption, there is now a disadvantage stating that data warehouses are difficult to upscale and out scale. When looking at Table 21, this appears to be a considerable drawback of the data warehouse, given that all the respondents had indicated adding this disadvantage to the model. Luckily, this drawback has already been discussed and added to the model.

The third disadvantage was proposed by Respondents 3 and 4, which states that data warehouses have proprietary data formats. This means that the file format is customized so that only data warehouses can read it. This is perceived as a significant disadvantage, given that this will result in data silos. One could argue that this is beneficial in terms of security since the data cannot be read elsewhere but in that data warehouse. However, in general, this is disadvantageous because within the data warehouse data silos can be created, making it difficult for business analysts to combine data from, for instance, two different departments that each have their data silo. This story already came forward when one disadvantage was discussed that was included in the model. Just as for the previous disadvantage, in Section 5.1.1. a discussion can be found where proprietary data formats in a data warehouse were discussed. According to Respondent 3, it is tied to the lack of flexibility of a data warehouse. Therefore, we had already concluded this should be a new disadvantage for the data warehouse.

Two respondents pointed out that the data warehouses do not support modern workloads like streaming, machine learning, or data science use-cases. This aspect was most likely not included in the model because of the focus on what the data warehouse is intended for. Since there are advantages included stating that data warehouses are very good for delivering business intelligence and reports, this does not directly imply that it is not at all suitable for the more modern use-cases. Although, these use-cases almost always include unstructured or semi-structured data. That does not mean, however, that the obvious should be left out. Hence, this disadvantage is now included in the model.

Finally, from the four proposed disadvantages, we already included two in the model when we evaluated each assumption, and a discussion can be found in Section 5.1.1. Besides the two already added, we concluded that the remaining two disadvantages should also be included in the model. Hence, in Figure 20, an updated version of the data warehouse entity is shown.

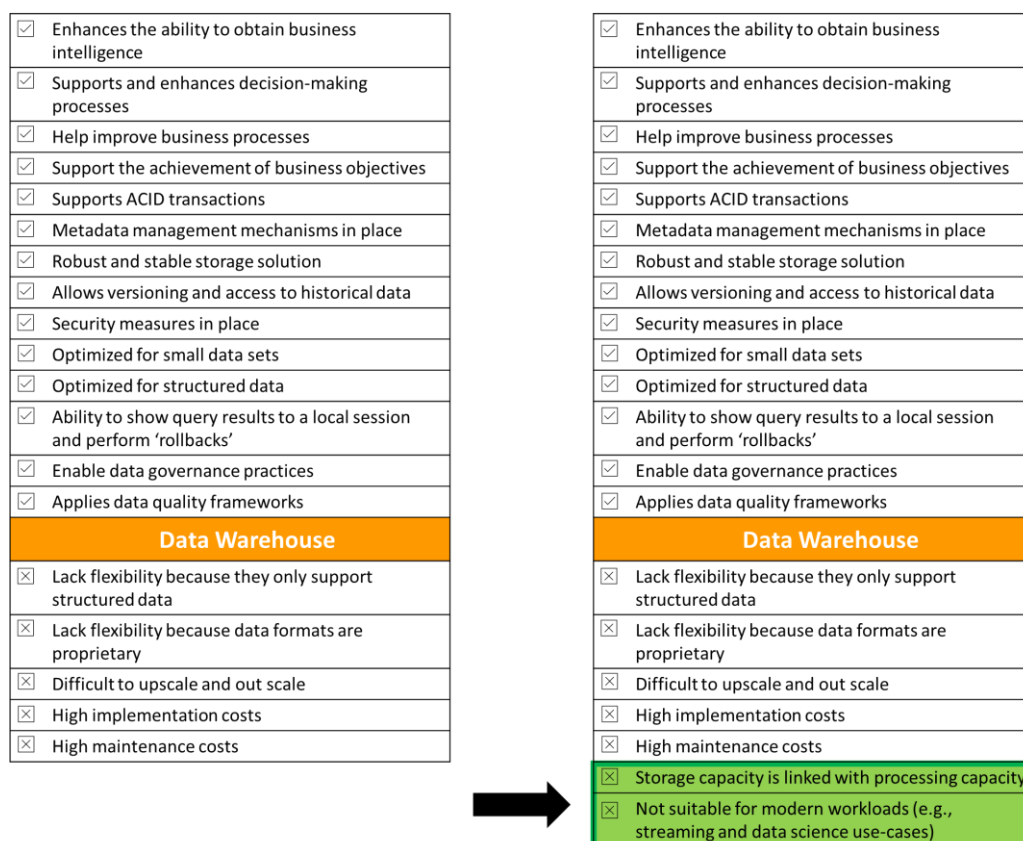


Figure 20: Adjusted Data Warehouse Entity 3.0

5.2.1.2. Suggestions for Data Lakes

For the data lake, three advantages were suggested to add to the model. Again, the suggestions are structured the same way as the assumptions are structured that were already included in the model. Table 22 provides an overview of the suggested advantages.

Advantages	Respondents				
	1	2	3	4	5
Data lakes are open-sourced	✓		✓		
Data lakes provided by Azure have a native integration with Azure Active Directory			✓		
Data lakes have the storage and processing capacity decoupled	✓			✓	

Table 22: Suggested Advantages for Data Lakes

Respondents 1 and 3 shared the first advantage, which indicated that the data lakes are open-sourced. This means that anyone can implement and adjust a data lake's source code. According to the respondents, this is advantageous given that it is possible to personalize the data lake according to your liking if you have the right data engineers for that. However, there is a downside to this story, according to Respondent 1. The fact that anyone can create and adjust a data lake causes there to be numerous different implementations which are pretty chaotic and overwhelming. There are so many creations, hence, it makes it challenging to choose one that best fits your use case and also delivers the best performance. Nevertheless, the advantage outweighs this risk/downside of a data lake, and both respondents believe this advantage should be included in the model, and their suggestion has been carried out.

The second advantage was provided by Respondent 3, who highlighted that this advantage only counts if the focus is on data lakes offered by Azure. The proposed advantage states that the Azure Data Lakes have a native integration with Azure Active Directory. This is beneficial since it enhances the security of the data in the data lake. We were a bit hesitant about whether this should be included for this advantage. On the one hand, this research is performed in collaboration with an IT-consultancy firm that only works with Azure resources. Whereas on the other side of the story, this research presents a model where literature and practice are combined to generate a high-over model that explains the value of each storage architecture. On top of that, this research is also performed for the university and science, hence, we prefer not to be biased. Given that this advantage only counts for Azure Data Lakes, we have chosen not to include this in our model because of the bias towards one data lake provider.

The last advantage was suggested by Respondents 1 and 4, who mentioned that the storage and processing/computing capacity are decoupled. We have seen that this is not the case for data warehouses and is perceived as a significant disadvantage. That explains why a data lake, the successive generation of a data warehouse, has these two capacities decoupled. This means that you can have, for instance, two computing platforms operating on the same data lake. Each computing platform can be used for specific operations like one to run your entire network on and another little platform for all the data transformations. Besides, it is very convenient to upscale your storage capacity when the amount of data increases rapidly or when it is desired to store more historical data. Hence, there are multiple reasons why it is advantageous to have the storage and compute/process capacity decoupled. Therefore, this has been included in the model.

Finally, we have included two proposed advantages out of three in total. An updated version with regard to the advantages of the data lake entity is presented in Figure 21.

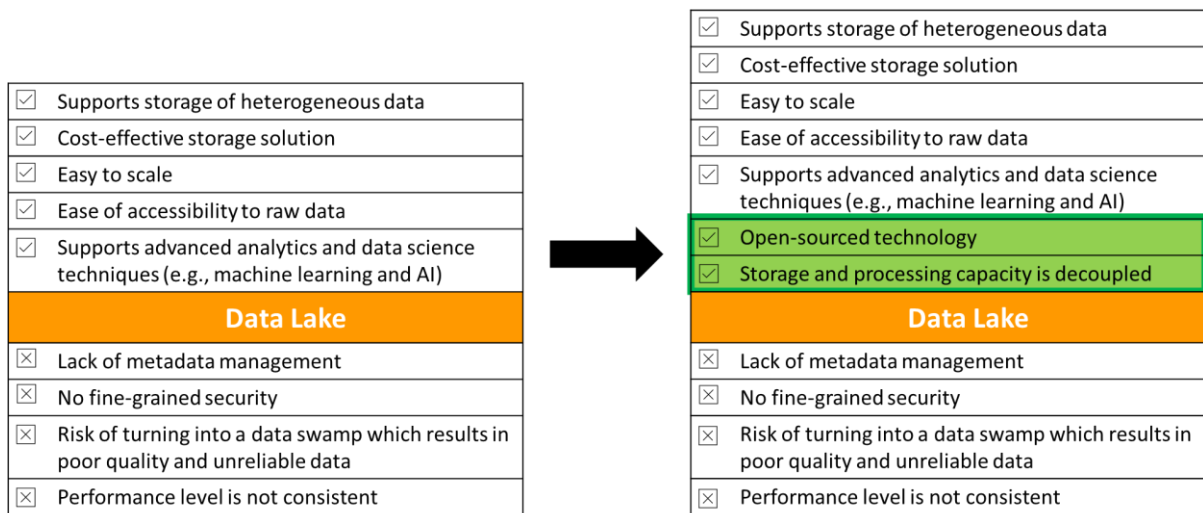


Figure 21: Adjusted Data Lake Entity 2.0

Besides the suggested advantages, some recommendations were also given regarding disadvantages that were not included in the model, of which an overview is provided in Table 23. In total, five disadvantages were proposed to include in our model.

Disadvantages	Respondents				
	1	2	3	4	5
Data lakes' latency is dependent on the underlying storage	✓				
Data lakes do not provide auditing and logging out-of-the-box			✓		
Data lakes do not support backup solutions out-of-the-box			✓		
Data lakes are not capable of implementing good data governance practices		✓	✓	✓	✓
Data lakes have difficulties complying with GDPR and PII rules				✓	✓

Table 23: Suggested Disadvantages for Data Lakes

The first disadvantage was presented by Respondent 1, who indicated that the latency of a data lake depends on the underlying storage object. This disadvantage was not considered for the data lake entity, although it was included in the lakehouse entity. No literature was found that made this claim that this is the case for data lakes. However, after discussing this disadvantage with Respondent 1, understandably, the latency depends on the underlying object storage. Object storage is a storage architecture that handles large amounts of unstructured data. In other words, this is the architecture of a data lake, whereas a data warehouse can be seen as file storage, given that data is stored in files or folders that are named and tagged and organized under a hierarchy of directories and subdirectories (IBM Cloud Education, 2019). The disadvantage is that different object stores are available as options to choose from. Other vendors provide them, and each object store can have different specifications and thus differs in costs as well. In the end, the latency is likely to vary the most, according to Respondent 1. Therefore, Respondent 1 believes this should be included in the model. Since this was already included as a disadvantage for the data lakehouse, and a data lakehouse has a data lake as its object store, it is self-evident that this advantage will also be added to the data lake entity.

The second disadvantage was proposed by Respondent 3, who shared that auditing and logging cannot be done out-of-the-box with a data lake. For this, another resource from Azure should be connected to the data lake to be able to do auditing and logging. In order to first consider whether this is of importance, the definition of auditing and logging was discussed. The respondent explained that auditing typically tracks and records domain-level events, e.g., a transaction is created or a user is performing an action. Auditing focuses on who did what and possibly also why it was carried out. Whereas logging is focused on recording events at the implementation level, which dives into the details of, for instance, which methods get called and which objects are created. So, logging is basically focused

on what is happening and is of interest to data engineers/programmers. Recording these events is particularly important for troubleshooting problems because it can pinpoint what happened, which event caused the bug, and how to resolve this issue. This helps to protect the systems' integrity and is, therefore, a very important feature for storage technology. Consequently, this disadvantage has been included in the model due to the value and importance of providing auditing and logging out of the box.

The third disadvantage was also shared by Respondent 3, stating that data lakes do not support backup solutions out-of-the-box. This is a serious disadvantage because people now have to build custom backup solutions that copy data into different storage accounts. Automatic point-in-time snapshots are not enabled in data lakes by default which is used as a method for data protection. This feature automatically takes snapshots of the data so that in the event of failure, it is possible to restore your data to the most recent snapshot before the failure. Although people build custom backup solutions, the risk here is that this is not developed properly, which can cause serious issues when an event of failure occurs, and the custom-made backup solution does not work. Given the severity of the consequences of lacking proper backup solutions, we have included this disadvantage.

Another disadvantage that was proposed was that data lakes are not able to adhere to good data governance practices. This is confirmed by four respondents who all shared that data governance is a serious problem for data lakes. As mentioned before, a data warehouse is typically schema-on-write, but a data lake is the opposite of this and is typically schema-on-read. This means the data does not have to be structured and adhere to some data model. Instead, data can be ingested into the data lake in any format, and a schema is parsed when the user wants to read and interpret the data. This allows for extreme flexibility in terms of storing different types of data. However, it takes more effort to handle and identify each data piece, given that the schema used to read the data constantly varies. A major downside of having different data formats is that it is difficult to have consistent data governance measures in place. When the data format is unknown, it is challenging to know how to govern this data. This includes not knowing how to carry out metadata management, write policies and standards on how and what can be done with the data, and it is hardly possible to govern data ownership in storage where data of all sorts is loaded and processed that first needs to be read with specific algorithms to interpret the data and give meaning to the data. All in all, it can be concluded that data lakes cannot perform good data governance. Thus, this disadvantage will be included in the model.

The last disadvantage was proposed by both Respondents 4 and 5. They claimed that a data lake has difficulties adhering to GDPR and PII rules. PII stands for Personally Identifiable Information, meaning that this type of information can identify an individual. The moment of discovering sensitive information in the data lake is after the data has been read and interpreted by, e.g., data engineers or data scientists. This means that vulnerable information was stored in the data lake before the data was actually being read and interpreted. This makes a data lake very vulnerable to security threats. Unfortunately, it is very difficult to implement standards, policies, or algorithms to comply with the GDPR rules and to protect PII. Users have likely built some custom practices to try and tackle this challenge. However, the fact remains that it is very difficult for data lakes to support practices that help to comply with GDPR or PII rules. Hence, this disadvantage is added to the model.

In conclusion to the suggested disadvantages from the experts, all the suggested disadvantages have been added to the model. A revised entity to represent the disadvantages of the data lake can be found in Figure 22.

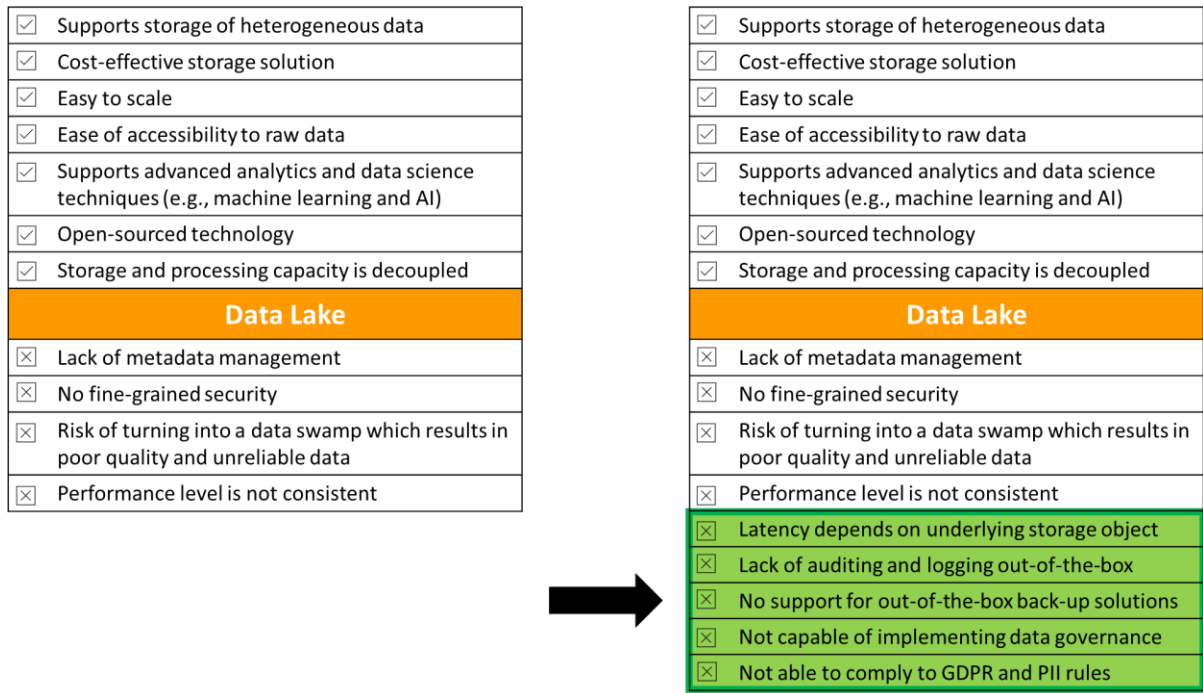


Figure 22: Adjusted Data Lake Entity 3.0

5.2.1.3. Suggestions for Data Lakehouses

The last entity discussed was the data lakehouse, and again the experts were asked to share any advantage or disadvantage that they thought should be included in the model. In the end, three distinct advantages were suggested and can be found in Table 24.

Advantages	Respondents				
	1	2	3	4	5
Data lakehouses are open-sourced	✓		✓		
Data lakehouses support the use of different computing languages interchangeably			✓		
Data lakehouses have the storage and processing capacity decoupled	✓	✓		✓	
Data lakehouses have the potential to support data governance out of the box with a unity catalog	✓	✓	✓	✓	✓

Table 24: Suggested Advantages for Data Lakehouses

Respondents 1 and 3 suggested the first advantage, which indicated that the data lakehouse is open-sourced. This means anyone can obtain the lakehouse's source code, build their implementation, and adjust it to their liking. This is perceived to be an advantage given that it provides flexibility in terms of custom-build solutions for the lakehouse, and it provides transparency into what and how the data lakehouse works. This advantage also counts for the data lakes. As we explained there, being able to develop your lakehouse and add some personalization to make it perfectly fit your use-case or company is very advantageous. Therefore, this advantage has been added to the model.

The second disadvantage was shared by Respondent 3, who mentioned that the data lakehouses support multiple different computing languages interchangeably. This is an advantage given that it provides the user with the flexibility to use any computing language the user is capable of, e.g., Python, Scala, Java, SQL, or other languages. The interesting fact about this advantage is that Respondent 1 also shared that different computing languages could be used. However, it could have a negative effect and make users reluctant to use the data lakehouse. According to him, the more senior programmers are less willing to learn those newer and more advanced programming languages. Given that there are so many possibilities for doing advanced analyses, this can be a deterrent which is disadvantageous. However,

SQL is still enabled by the data lakehouse, although it might be less convenient to do SQL operations than a data warehouse where all the data is stored in a structured manner. Despite this risk of having reluctant users, the fact that you can adopt any computing language is still seen as a big advantage of a lakehouse due to its flexibility. Thus, this advantage will be included in the model.

Thirdly, three respondents indicated that a data lakehouse's storage and processing capacity is decoupled. This is advantageous because it makes the data lakehouse architecture flexible. When it needs to be scaled, storage is not tied to processing and the other way around. Since this benefit was also mentioned for the data lake, we refer to the fourth paragraph in section 5.2.1.2 for a more in-depth explanation of why decoupling the storage and processing capacity is advantageous.

The last advantage is a very remarkable one, given that all the respondents shared it. This only happened for one disadvantage of the data warehouse and now for one advantage of the data lakehouse. The respondents claimed that the lakehouse has the potential to support data governance practices out-of-the-box with a unity catalog. They all shared their knowledge and interpretations of what the unity catalog is going to be capable of, and even some sources were shared that were published by Databricks to validate their claims. At the Data and AI Summit 2021, Databricks announced the Unity Catalog which is a unified governance solution for data and AI and will be natively built into the Databricks Lakehouse Platform (Zaharia et al., 2022). The vision for this catalog is to enable unified governance for which standard governance models can be used that apply to all different types of data. Databricks has shared numerous promising features of the unity catalog. First of all, automatic end-to-end data lineage will be provided to easily obtain an overview of how data flows in the data lakehouse and impact analysis of data changes. Second of all, Databricks desires to enable deep integrations between Unity Catalog and existing catalogs and governance solutions that are already implemented at the client. Another promising feature is the native integration with Delta Sharing, which is the world's first open protocol for data sharing. This enables you to add fine-grained governance and data security controls. As a result, sharing data will be more convenient and secure for internal and external sharing. A fourth key feature is the built-in data search and discovery feature which helps to search and reference relevant data sets quickly. A fifth feature is enabling fine-grained governance with Attribute Based Access Controls (ABACs). This feature allows you to mark multiple columns as PII and manage access to all those columns with one single rule. Next to that, with the Unity Catalog, you can configure unified data access and obtain detailed audit reports on how data is accessed and by whom. This is essential to monitor data compliance and adhere to security requirements. Finally, it will also be possible to accomplish centralized metadata management and user management. All in all, there are a lot of promising features. Unfortunately, the unity catalog is still in its infancy stage, and it is still under development. It has been made available as a preview to customers from Databricks. However, the fact remains that the unity catalog is not there yet. Still, it sounds very promising and as soon as it is further developed, this will be a unique selling point of the data lakehouse and surpass the other storage architectures. Therefore, we have chosen to include this into the model but present it differently than the other advantages of the data lakehouse.

In summary, the entity representing the data lakehouse will be expanded with four new advantages, of which one advantage is differently visualized than the others. As explained, there is one advantage that is very promising and will be very beneficial. However, it is still under development. Despite that, all respondents have indicated that this feature will be a distinguished selling point of the data lakehouse. Therefore, we have decided to include this potential advantage in the model, but we have chosen to fade the colour of the box to indicate there is something special with this advantage in the final model. A visualization of this can be seen in Figure 23.

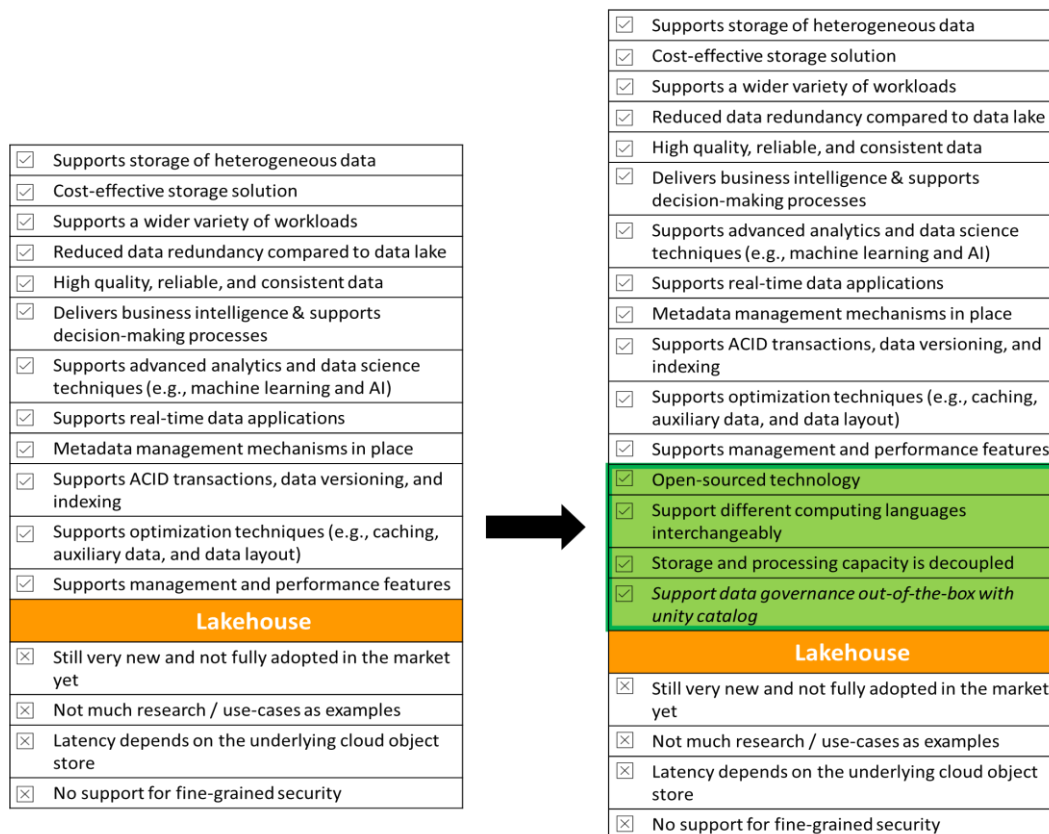


Figure 23: Adjusted Data Lakehouse Entity 2.0

Besides the suggested advantages of a data lakehouse, some recommendations were also given regarding disadvantages that were not included in the model. An overview is provided in Table 25, where it is shown that four new disadvantages were suggested.

Disadvantages	Respondents				
	1	2	3	4	5
Data lakehouses are only provided by one vendor (Databricks)	✓			✓	
Data lakehouses do not have a native integration with Azure Active Directory			✓		
Data lakehouses have nothing build-in yet for data governance			✓		✓
Data lakehouses do not provide fine-grained security				✓	✓

Table 25: Suggested Disadvantages for Data Lakehouses

The first disadvantage was shared by Respondents 1 and 4. At the moment, Databricks is the only vendor that provides the data Lakehouse since they are the ones who have developed this storage architecture. According to Respondent 4, the lakehouse is even a vendor-specific term for this concept that Databricks has invented. The disadvantage is that when you adopt the data lakehouse, the client is locked with Databricks as their vendor. In general, vendor lock-in is perceived as a disadvantageous situation. Therefore, two respondents shared this as a disadvantage for the data lakehouse. Despite the fact that the lakehouse is open-sourced, it is still a new technology. Hence, there is no second large provider yet of the data lakehouse. Therefore, as of now, this is a disadvantage of the data lakehouse and will be included in the model.

The second disadvantage was proposed by Respondent 3, who indicated that the data lakehouse does not have a native integration with Azure Active Directory. The same respondent shared this as a suggested advantage for the data lake. However, just as we argued then, this is a vendor-specific characteristic, and it is preferable to be as unbiased as possible. When a data lakehouse is implemented, there are numerous configurations possible, and Azure Active Directory is not the only service that can

be used to configure security controls. Given that this research also intends to contribute to the literature, we have chosen to keep this disadvantage out of the model.

The third disadvantage mentions that, as of now, there is nothing implemented for data governance. This was mentioned by Respondents 3 and 5 and indirectly by all the other respondents as well. When discussing the suggested advantages, all respondents indicated one advantage, which implied that the unity catalog would provide data governance out-of-the-box. However, the unity catalog is still under development and has not yet been made available to the wider audience. Hence, at this moment in time, there is nothing yet in place. Therefore, it is suggested to include this disadvantage since that applies to the current situation. Given that it is currently applicable, we have included this disadvantage in the model.

Finally, the last disadvantage was proposed by Respondents 4 and 5, who claim that the data lakehouse does not provide fine-grained security. This suggestion was discussed at the same time when one of the assumptions was evaluated. The model had one assumption that claimed that the lakehouse provides high-security assurance. However, Respondents 4 and 5 disagreed with this and proposed to add an assumption stating that the lakehouse does not provide fine-grained security. This has already been discussed in Section 5.1.3 and was already added to the model.

To conclude, after evaluating the four suggested disadvantages, we chose to include two disadvantages, exclude one disadvantage, and discovered one suggested disadvantage was already discussed in Section 5.1.3 when the assumptions that were included in the model were evaluated. The adjusted entity of the data lakehouse is shown in Figure 24.

<input checked="" type="checkbox"/> Supports storage of heterogeneous data	<input checked="" type="checkbox"/> Supports storage of heterogeneous data
<input checked="" type="checkbox"/> Cost-effective storage solution	<input checked="" type="checkbox"/> Cost-effective storage solution
<input checked="" type="checkbox"/> Supports a wider variety of workloads	<input checked="" type="checkbox"/> Supports a wider variety of workloads
<input checked="" type="checkbox"/> Reduced data redundancy compared to data lake	<input checked="" type="checkbox"/> Reduced data redundancy compared to data lake
<input checked="" type="checkbox"/> High quality, reliable, and consistent data	<input checked="" type="checkbox"/> High quality, reliable, and consistent data
<input checked="" type="checkbox"/> Delivers business intelligence & supports decision-making processes	<input checked="" type="checkbox"/> Delivers business intelligence & supports decision-making processes
<input checked="" type="checkbox"/> Supports advanced analytics and data science techniques (e.g., machine learning and AI)	<input checked="" type="checkbox"/> Supports advanced analytics and data science techniques (e.g., machine learning and AI)
<input checked="" type="checkbox"/> Supports real-time data applications	<input checked="" type="checkbox"/> Supports real-time data applications
<input checked="" type="checkbox"/> Metadata management mechanisms in place	<input checked="" type="checkbox"/> Metadata management mechanisms in place
<input checked="" type="checkbox"/> Supports ACID transactions, data versioning, and indexing	<input checked="" type="checkbox"/> Supports ACID transactions, data versioning, and indexing
<input checked="" type="checkbox"/> Supports optimization techniques (e.g., caching, auxiliary data, and data layout)	<input checked="" type="checkbox"/> Supports optimization techniques (e.g., caching, auxiliary data, and data layout)
<input checked="" type="checkbox"/> Supports management and performance features	<input checked="" type="checkbox"/> Supports management and performance features
<input checked="" type="checkbox"/> Open-sourced technology	<input checked="" type="checkbox"/> Open-sourced technology
<input checked="" type="checkbox"/> Support different computing languages interchangeably	<input checked="" type="checkbox"/> Support different computing languages interchangeably
<input checked="" type="checkbox"/> Support data governance out-of-the-box with unity catalog	<input checked="" type="checkbox"/> Storage and processing capacity is decoupled
	<input checked="" type="checkbox"/> Support data governance out-of-the-box with unity catalog
Lakehouse	Lakehouse
<input type="checkbox"/> Still very new and not fully adopted in the market yet	<input type="checkbox"/> Still very new and not fully adopted in the market yet
<input type="checkbox"/> Not much research / use-cases as examples	<input type="checkbox"/> Not much research / use-cases as examples
<input type="checkbox"/> Latency depends on the underlying cloud object store	<input type="checkbox"/> Latency depends on the underlying cloud object store
<input type="checkbox"/> No support for fine-grained security	<input type="checkbox"/> No support for fine-grained security
	<input type="checkbox"/> Databricks is the only Lakehouse provider (vendor lock-in)
	<input type="checkbox"/> No providence of build-in data governance

Figure 24: Adjusted Data Lakehouse Entity 3.0

5.2.2. Suggestions of Experts on Model-Level

Next to evaluating each entity individually, the respondents were also asked to evaluate how the model was constructed. In particular, the way in which the entities were put into context and how the relationships have been modelled. A detailed overview of the suggestions made by the experts is presented in Appendix D. When looking at the suggestions made by the five experts and the general themes of their suggestions, we discovered six distinct themes. An overview is provided in Table 26.

Themes		Respondents				
		1	2	3	4	5
1	Colouring	✓				✓
2	Highlight your message	✓	✓	✓	✓	✓
3	Technical vs. Process/People	✓				
4	Storage vs. Compute		✓	✓		
5	Evolution from left to right		✓		✓	
6	Reverse relationships from disadvantage to advantage				✓	
7	1 high-level model and 1 detailed model					✓

Table 26: Overview of the Overarching Themes of the Suggestions for Improving the Model as a Whole

The first overarching theme is ‘Colouring’. In total, two respondents implied that the model's colouring could be improved. Respondent 1 indicated that the plusses and minuses should be better highlighted. They argued that when you just take a glance at the model, you cannot immediately see what the boxes entail and what is being visualised. Only when you take the time to read the model, it will become clear. This is also what Respondent 5 shared, and according to him, the colours now used seem to be random and do not say anything.

While Respondents 1 and 5 were talking about colouring, they stressed that this could help convey the message better. This is the second overarching theme that all respondents mentioned. They all suggested emphasising the message the model is trying to convey and visualising this. All the experts understood that the model attempts to show how certain advantages of one technology solve certain disadvantages of another. However, this was not entirely clear without an explanation or reading the entire model. This is a very significant point of discussion since bringing across a message is the primary purpose of a model.

The third and fourth suggestions are somewhat related to each other. Respondent 1 indicated that some advantages and disadvantages are technical-related, and others are more process-related or people-related. Whereas Respondents 2 and 3 indicated that some advantages are storage-related, and others are compute-related. They all recommended making a distinction in the model between the aspects they had mentioned. However, we have chosen not to do this for two reasons. First of all, we believe the model will become too complex if the advantages and disadvantages were grouped according to being technical-related, people-related, storage-related, or compute-related. In some cases, they are probably related to both technical and storage, or computing and people, which can easily become very complex and difficult to understand. Another argument is that a few formulated assumptions indicate that storage and compute capacity are decoupled, whereas for a data warehouse this is linked to each other into one entity. Besides that, the fact that some advantages or disadvantages are people-related is also taken into account within the formulation of the assumptions. Therefore, we believe it is not necessary to visualize these distinctions as it will not improve the model in terms of understandability.

Another suggestion was to visualize the evolution from left to right. Given that the storage architectures are developed in sequential order, it would be a logical representation. This view was strongly shared by Respondent 2, whereas Respondent 4 indicated that he would expect a sequential order. But he

understood what the model was trying to achieve in terms of showing how the advantages of the data lakehouse can solve the disadvantages of a data warehouse and a data lake, given that the data lakehouse is a combination of both architectures. We agree that this evolutionary process is linear.

The second-last suggestion was about the relationships that showed which advantage solves which disadvantage. Respondent 4 indicated that he prefers to see the starting point to be the disadvantage and then have an arrow pointing towards the advantage that is supposed to solve it. Instead of going from positive to negative, the personal preference of Respondent 4 is to have the relationship from negative pointed toward positive. We believe this is a valid point, even though it is based on a preferred personal interpretation. Also, the message of the relationship is that some advantages of the second generation solve the disadvantages of the first generation. And some disadvantages of the second generation are solved by some advantages of the third generation. Hence, by modelling the relationships using the disadvantage of one technology, the reason why the second technology was developed will be touched upon in this way. Therefore, we have chosen to incorporate this suggestion into our model.

Finally, Respondent 5 thought of the idea of creating two models, one model visualizing on a higher level the relationship between the three storage architectures and one model on a detailed level where you have all the assumptions for the storage architectures. The purpose of the first model is to provide an easy overview of having three different storage architectures and emphasise that the data lakehouse has overlapping aspects with both a data warehouse and a data lake. In this way, the audience will immediately grasp the essence of the data lakehouse, which is a combination of the two previously developed architectures. Evidently, an explanation should be given that starts with a timeline indicating that the data warehouse was the first generation, after which the data lake was developed, and finally the data lakehouse. Perhaps each storage architecture can be displayed in a coloured object, which should also be used for the more detailed model. This second model will explain how each generation tries to solve the previous generation's problems with the formulated assumptions. With the other overarching themes in mind, the last idea indirectly agrees that the model should picture the storage architectures in sequential order since the message is that the following generation solves the problems of the previous generation.

To summarize, after evaluating the suggestions and their reasoning, we have already concluded that two suggestions will not be incorporated into the model. We have chosen to ignore the suggestion of grouping the advantages and disadvantages according to technical- or people-related. On top of that, we have also decided to do the same with the suggestion of presenting advantages and disadvantages related to storage in one box and computing in another. This is not followed since we believe it does not contribute to bringing across the message, making the model even more complex and harder to understand. As for the other suggestions, an explanation is given on how they have been incorporated into the revised model after the model is first presented in the next section.

5.2.3. Final Version of the Data Storage Evolution Model

To summarize all the improvements that were made, the following list of requirements was formulated for the design of the revised model.

- **Colouring:** we decided to visualize each storage solution with a different colour. This is because it is now more convenient to visualize which advantage of a data warehouse and a data lake is incorporated into a data lakehouse.
- **More emphasis on the message:** to put more emphasis on the message, we have changed the icon displayed in front of the advantages and disadvantages. It is now more apparent what the advantages and disadvantages are by having thicker and clearer icons representing a check for the advantages and a cross for the disadvantages.

- **Improve relationships:** to improve the relationships between the entities, we have chosen to use a disadvantage as a starting point. From there, an arrow is drawn to the advantage of the following storage architecture that will tackle and solve the disadvantage. In that way, the message will also better come across because the emphasis is now put on how the subsequent storage architecture tries to solve the issues of the preceding architecture.
- **Evolution from left to right:** based on the insights provided by the experts, we have chosen to display the model from left to right instead of a circle. This way, it will be more apparent that there is an evolutionary process. Moreover, the message of showing how a subsequent storage architecture solves problems of the preceding architecture will be better expressed when the model is displayed from left to right.
- **1 high-level model, 1 detailed model:** we have chosen to design one high-over model to visualize the idea behind the data lakehouse. Namely, it is a combination of the data warehouse and the data lake. This is not a requirement for the revised model. However, given that the first version of our model was visualized in a cyclical form, the idea rose to present a high-over model that serves as an introduction to the data lakehouse concept.
- **Place namings on top of the entity:** this improvement was recommended to enhance clarity since an entity is generally read from top to bottom. Hence, it is more convenient to have the name of the storage architecture on top instead of in between the advantages and disadvantages.

Based on the suggestions provided during the interview, a high-over model is presented in Figure 25. This model indicates the overlapping strengths between the data warehouse and a data lakehouse, and between a data lake and a data lakehouse. The colours by which each storage architecture is represented are also used for the revised Data Storage Evolution Model. The strengths of the data lakehouse that overlap with the particular strengths of a data warehouse are now shown in the colour representing the data warehouse. The same is done for the overlapping strengths between the data lake and a data lakehouse. The revised Data Storage Evolution Model is presented in Figure 26.

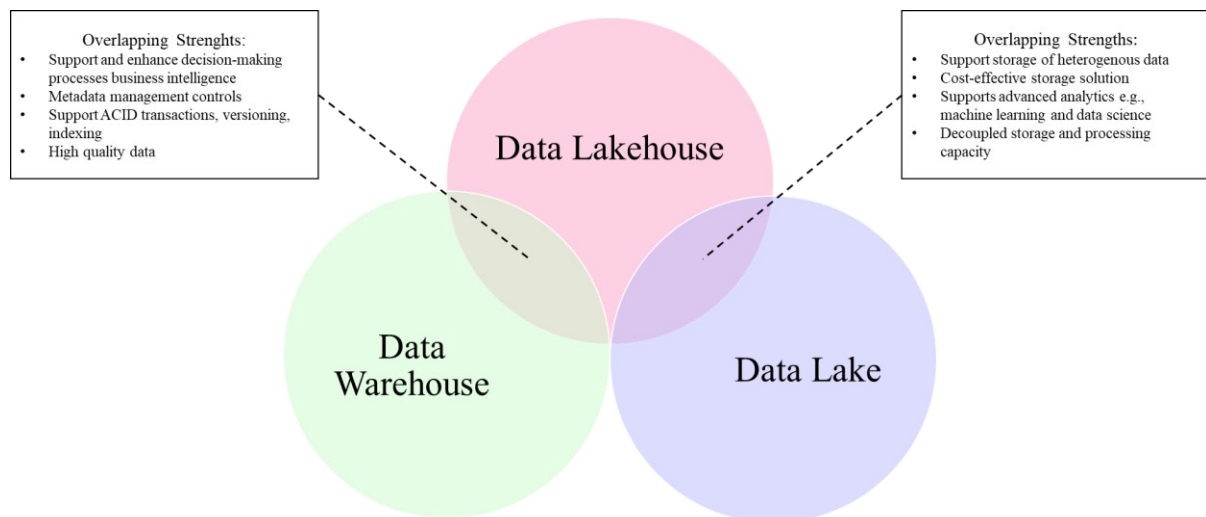


Figure 25: High-Level Model Showing the Idea Behind a Data Lakehouse



Figure 26: Revised Data Storage Evolution Model

5.3. Interview Results on Data Lakehouse Architecture

The second part of the interview focused on the data lakehouse's architecture. In Section 4.3. we evaluated the architectures of the data warehouse, data lake, and data lakehouse. In sub-section 4.3.3. we have evaluated three distinct architectures that all represent the data lakehouse. The experts were asked to evaluate all three of them and share their views on the core parts of the lakehouse architecture and how the presented architectures could be improved if necessary. An overview of all the interview questions can be found in Appendix A.

In the following sections, we will evaluate the insights gathered during the interviews. Afterwards, with the help of the gathered insights from the experts and literature, a list of requirements is created that define the key elements of a data lakehouse architecture. Finally, a sophisticated architecture will be presented to describe the data lakehouse's core idea and visualise the data flows and processes in a data lakehouse architecture.

5.3.1. Insights on Existing Architectures

In Section 4.3.3, three distinct architectures representing the data lakehouse were discussed. They were all evaluated based on the explanations provided in the sources from which the architectures were extracted. During the interviews, the experts were asked to evaluate them and explain which one represents the data lakehouse best. The architectures were discussed in the order that is shown in Figure 27.

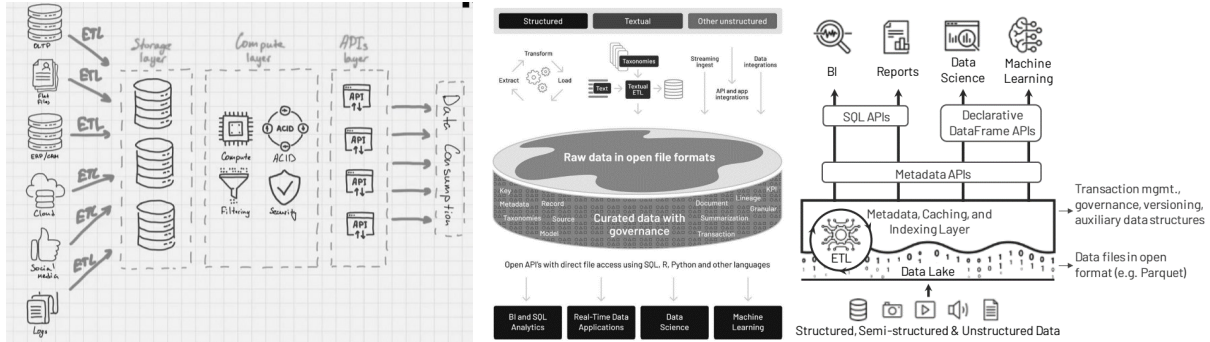


Figure 27: Data Lakehouse Architectures (Armbrust et al., 2021; B. Inmon et al., 2021; Lavrentyeva & Sherstnev, 2022)

An elaborated overview of what the experts shared during the interview is provided in Appendix E. When looking at the observations made by Respondent 1, the first thing that came to mind was that each architecture takes a different perspective. Since they all have different perspectives, their function is also different. However, Respondent 1 clearly indicated that the first architecture is the preferred architecture that best describes how the data lakehouse should be constructed. According to Respondent 1, this architecture contains all the essential parts necessary to build a data lakehouse architecture and clearly indicates the layers of the lakehouse architecture. As for the second and third architecture, they represented more the data lake and not the lakehouse. Although in both architectures, an extra layer is put on top of the data lake, it is not sophisticated enough to explain what and how the construct represents the idea of the data lakehouse.

On the contrary, Respondent 2 shared that the preferred architecture is the third picture with a sidenote that it needs some refinement. Despite a different preference, as Respondent 1 mentioned, Respondent 2 shared that all the architecture is designed from a different perspective. The first takes on a data flow perspective, including all the processes the data goes through. The second is more from a data format perspective because it is more of a breakdown of the data pipeline. And the third is a combination of the previous perspectives, so both the data flow and the data format perspective are utilized here. As a result, that is why the third architecture is preferred by Respondent 2. However, there is one thing lacking, which is a more refined representation of the data lake that consists of different data zones. Moreover, different use-cases utilize data from different zones. Hence it would be a more fine-grained architecture when those zones are included, and the different APIs draw arrows from a specific zone to a particular use case. This architecture is the best for the rest because the metadata, caching, and indexing layer is your lakehouse. The layer with APIs on top is not part of the lakehouse because that is just a computing layer, and below this layer, you have your data lake, which is just your storage layer.

According to Respondent 3, the third picture visualises the data lakehouse best. When the first picture was evaluated, Respondent 3 expressed that the architecture was fine but did not suffice as a data lakehouse architecture. This is because it does not show how the lakehouse can be the best of the data warehouse and data lake. Since the lakehouse is known for being the best of both worlds, this architecture is not precise enough, according to Respondent 3. As the architecture is designed right now, the storage layer can practically be any object storage. While the purpose of the lakehouse is to have a data lake implemented where the data management and governance layer is built on top. Then, the second architecture is pretty vague and very minimalistic. The architecture has too little information to

be clear and understandable. Moreover, it does not clearly show the concept of the lakehouse. Contrary to the previous two, the third architecture is straightforward and clear regarding the lakehouse concept. The only thing that is missing is the data zones in the data lake. As Respondent 2 pointed out, these data zones should be incorporated into the architecture, according to Respondent 3.

Respondent 4 preferred the first architecture, just as Respondent 1 did. According to Respondent 4, the architecture clearly shows the separation between storage and computing, which is the data lakehouse's most critical point. Moreover, this architecture is complete because it describes the data flow and process flow. The respondent perceives the second architecture as ambiguous and even strange. First, the architecture has a top-down approach which is not a usual way to visualize an architecture. Moreover, it is a very minimalistic view of what the lakehouse is. Without reading what the curated data and governance layer contains, the respondent would say it is a picture of the data lake. However, after analysing what features are included in this layer, the architecture is perceived slightly better than at first. Still, it does not suffice enough to be a worthy architecture of the data lakehouse. Lastly, the third architecture is very vendor specific. The remarkable point is that everyone recognized the third architecture as the original architecture of Databricks. However, Respondent 4 is the only one who criticises this architecture for being vendor specific. The reason for this is mainly because particular APIs are provided as the recommended APIs to work with. The respondent points out that the first architecture implies that different APIs can be utilized with the data lakehouse architecture. Whereas this architecture already suggests which APIs should be used. In addition to this, Respondent 4 is not so sure about what the unique selling point of the lakehouse is when looking at this architecture. The architecture itself does not explain the benefits of the governance layer built on top of the data lake, which is supposedly there to represent the data lakehouse concept. But in fact, it seems as if this is a data lake with something on top to take into account some governance practices. Therefore, this architecture would need some finessing to represent a data lakehouse instead of a data lake with a layer built on top.

Finally, the conclusion of Respondent 5 is that none of these three best represents the data lakehouse architecture. This is remarkable given that two respondents indicated the first is the best, and two preferred the third architecture. However, according to Respondent 5, all the architectures fail to answer the question 'what is the data lakehouse concept?'. The first architecture is acceptable in terms of showing the implemented layers, and it is clear how the data flows through the architecture. However, no emphasis is put on what part of the architecture is uniquely related to the data lakehouse. The second picture shows that different data formats can be put into some storage, which can then be used via different APIs for machine-learning purposes. This is a very minimalistic view and not fine-grained at all. Most importantly, it does not show what part of the architecture is lakehouse-specific. Lastly, the third architecture is a very high-level overview of having different data formats and use-cases that utilize the data. However, it does not clearly show how this architecture can be distinguished from a data lake. Hence, Respondent 5 believes all the architectures are not sufficing enough to explain the data lakehouse concept.

5.3.2. Additional Architectures Presented by the Experts

Next to evaluating the three architectures this research provided as an example, a few respondents shared the architectures they are familiar with. Respondent 1 shared two architectures (see Figure 28 and Figure 29), Respondent 3 also shared two architectures (see Figure 30 and Figure 32), and finally, Respondent 5 shared three architectures (see Figure 31, Figure 33, and Figure 34). In Appendix E, they are all presented. When comparing and evaluating these architectures, the following observations were made.

First of all, the architectures presented in Figure 28, 30, and 31 are very vendor specific. The first architecture is published by Microsoft, and evidently, it wishes to support the services that are provided by its cloud platform Microsoft Azure (Figure 28). The interesting part, however, is the layers they have included. There are in total five layers: ingest, store, process, serve, and monitor and govern. The flow within the architecture is a little bit complicated due to all the arrows. Nevertheless, one strong aspect of this architecture is how it shows that data is obtained from the data lake to utilize in the process stage. However, whenever the data is processed, it needs to be stored again. Hence, there is supposed to be some loop between these layers. This is what Respondent 2 had indicated during the interview, and this is exactly what Microsoft has implemented in their designed architecture. Another key strength of this architecture is the inclusion of the bronze-silver-gold data zones in the data lake. Moreover, the architecture shows that specific processing applications utilise data from specific zones to show that different use-cases use different data types. In addition, the monitor and govern layer indicates that this should be implemented and adhered to at all layers. Unfortunately, all the services presented there are from Microsoft Azure and thus vendor specific. However, the idea of what this layer should be capable of is definitely emphasized. We believe that these observations are pretty valuable. Even though this architecture is vendor-specific, it still adds value to the design of a conceptual reference architecture that this research will produce.

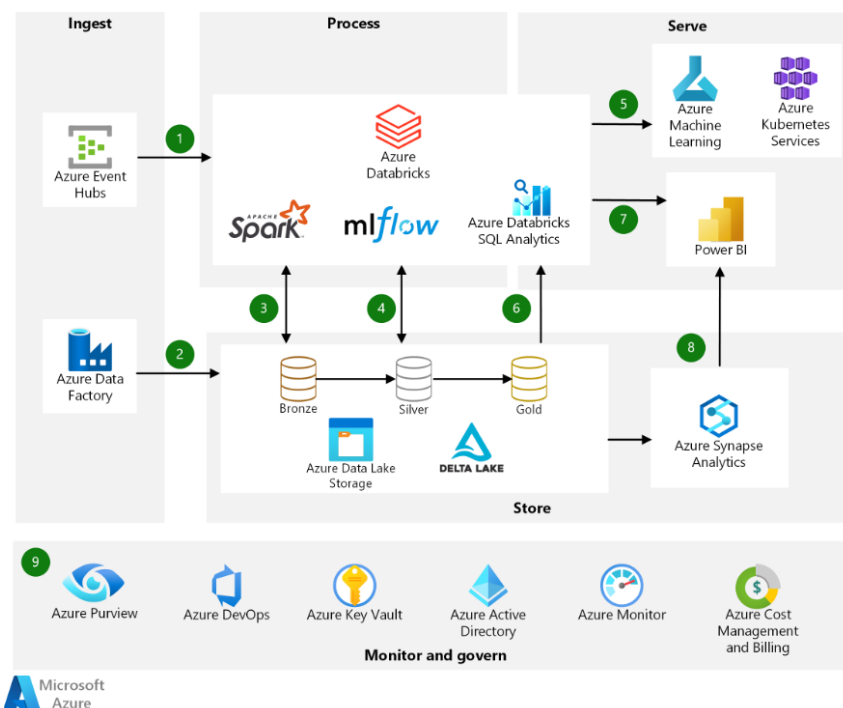


Figure 28: Proposed Data Lakehouse Architecture by Respondent 1 (Microsoft Azure, 2021)

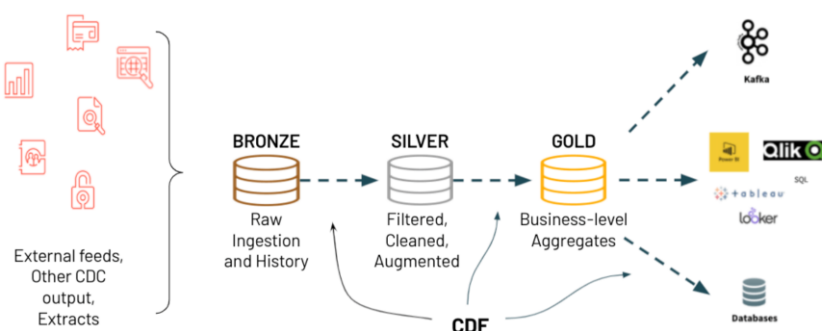


Figure 29: Proposed Data Lakehouse Architecture by Respondent 1 (Databricks Blog, 2020)

Next to that another remarkable observation is the fact that this architecture has no specific API layer, whereas the three architectures taken from our literature research have all implemented this. This is not per se a downside of the architecture, although it is remarkably missing. Additionally, there is no specific link between a processing application and a serving application utilized by the stakeholders when looking at the arrows going from the process stage to the serve stage. The last observation was related to the arrow pointing from the ‘Azure Data Lake Storage’ and ‘Delta Lake’ to the box where ‘Azure Synapse Analytics’ is displayed. Here, the assumption can be made that Azure Synapse Analytics represents a data warehouse in the data lakehouse. Hence, the data that is being stored here should be structured data. However, the architecture does not indicate whether only ‘gold data’ is stored in Azure Synapse Analytics. It now seems as if all the data stored in the data lake and delta lake is also stored in the Azure Synapse Analytics store. Thus, this part could be slightly more refined.

When looking at the two architectures presented in Figure 30 and Figure 31, the first observation is that they represent almost the same but with different visuals. This can be explained by the fact that they are both published by the same vendor: Databricks. Although there is one difference, the architecture in Figure 31 has included a specific layer called the unity catalog, which is not displayed in the other architecture (Figure 30). The only reason is that the right architecture is the most recent version. When Databricks announced the data lakehouse concept a few years ago, they had not announced the catalog yet. Therefore, the picture on the left is slightly outdated, given that it does not incorporate the governance layer yet. All in all, these two architectures do not provide newer or relevant insights next to the three architectures we have found and presented to the experts.

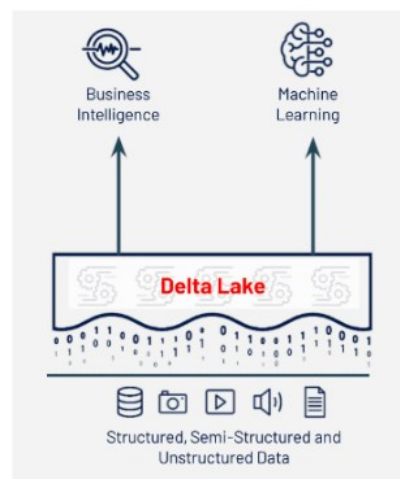


Figure 30: Proposed Data Lakehouse Architecture by Respondent 3 (Databricks, 2020)

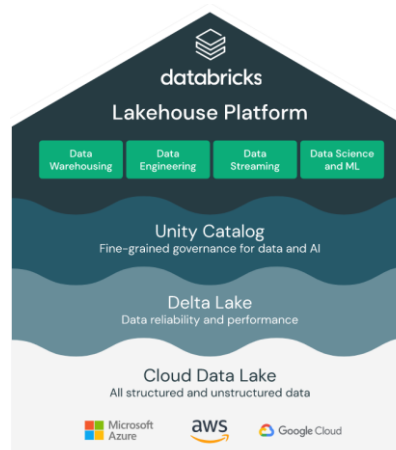


Figure 31: Proposed Data Lakehouse Architecture by Respondent 5 (Databricks Blog, 2021)

Then, Respondents 1 and 3 shared another architecture very similar to each other. The difference between the two is that the first (Figure 29) is more conceptually focused, and the second (Figure 32) is more practical-oriented by naming concrete storage solutions and use-cases. At first, it was unclear why the respondents shared this architecture, given that it seems incomplete when looking at the other architectures. However, the respondents indicated that this is a sophisticated representation of how the storage layer should be organized. The storage layer should have three data zones: bronze, silver, and gold. The bronze layer typically stores the data in its raw format, the silver layer stores data that has been filtered, cleaned, and enriched, whereas the gold layer stores structured data so it can be used for BI and reports immediately. That is the essential message that this architecture brings across.



Figure 32: Proposed Data Lakehouse Architecture by Respondent 3

Finally, Respondent 5 expressed his concern that the architectures extracted from the literature did not emphasize the lakehouse concept. Consequently, Respondent 5 shared architectures that have highlighted what the data lakehouse concept entails. The first that was shared (Figure 33) shows a very fine-grained architecture explaining different types of services that can be exploited in each layer. The structure of the architecture takes a bottom-to-top approach, with the data sources layer as the first layer. This layer clearly indicates the data types supported and it is even grouped according to which tool ingests which type of data. The second layer of the architecture presents different ingestion tools to showcase that there are multiple possibilities. This layer was not shown in the other architectures, but it is a nice refinement and valuable for depicting the process flow. Then the third layer is the storage layer which consists of two storage objects. One is a typical data warehouse storage object, and the other is a data lake storage object. Having two storage objects contradicts the idea of having one central repository in your data lakehouse. Implementing two storage objects is the same as having a two-tier architecture. Still, this architecture claims that the storage and catalog layers form the data lakehouse concept. This observation was raised during the interview. The expert argued that this architecture differs from the two-tier architecture because of the native integration between the two storage objects. This is possible since Redshift and S3 are both services provided by Amazon Web Services (AWS). Hence, if there is no native integration between the data warehouse storage (Redshift) and the data lake storage (S3), the architecture cannot serve as a data lakehouse architecture. Moreover, this architecture indicates that the catalog layer is also covering the data lake, which argues that this architecture does not represent a data warehouse and also not a two-tier architecture. Still, it is highly preferred to have only one type of storage in the data lakehouse architecture, given that this decreases the possible occurrence of integration errors.

On top of the storage layer, there is a catalog layer. It is not entirely clear what this layer holds from the architecture itself. However, this research has evaluated a catalog several times, and we know now this layer contains the data management and governance tools. The combination of the storage layer and catalog layers is how the lakehouse concept can be defined. This is highlighted in the figure by this orange box around the two layers.

Then the two remaining layers are the processing and consumption layer. First, the processing layer indicates the different processes that can be run on the data in a data lakehouse. Again, the processes seem to be grouped according to the different types of use cases. Finally, the consumption layer shows different tooling applications that can be used to consume the data. They are indirectly linked to the

different processing techniques shown in the previous layer. The only critique we have on this architecture's design is that it claims that having two types of storage services can still be part of the data lakehouse concept, which is quite confusing. The expert explained that this is only the case if there exists a native integration between the two storages. Still, it contradicts the definition of the data lakehouse and the architectures found in the literature. Therefore, when following the definitions we found, we do not agree that this architecture represents a data lakehouse architecture. However, we do believe that it is attempting to incorporate the data lakehouse idea, and it has highlighted this well.

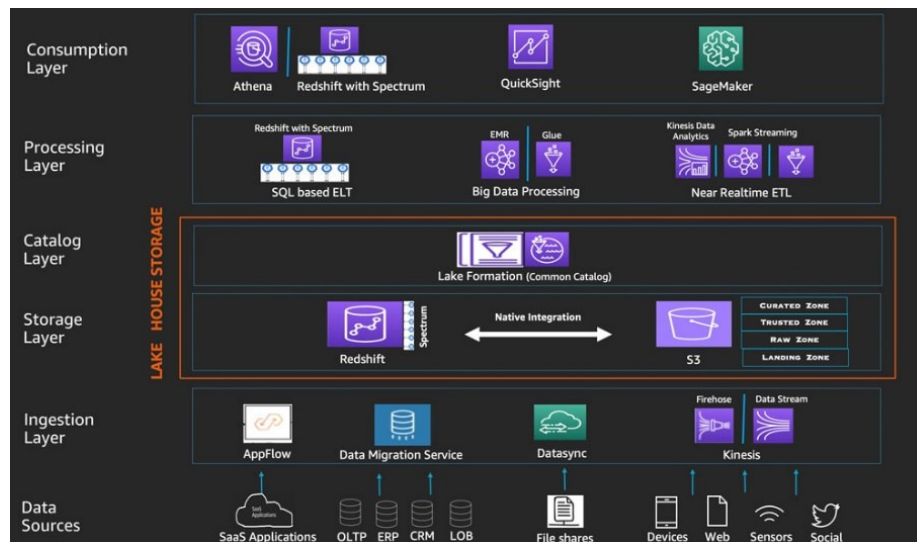


Figure 33: Data Lakehouse Architecture from AWS (Kava and Gong, 2021)

Respondent 5 shared another architecture where the data lakehouse concept is highlighted (Figure 34). Although, at first sight, the architecture does not look as fine-grained as the previous architecture showcased, it is still valuable to see what part of the architecture can be called the data lakehouse concept. First of all, we clearly see that the architecture is structured from left to right, and it starts with a data sources layer. Here, different data formats are presented to indicate a lakehouse's flexibility. After that, the architecture consists of 3 more layers and shows a very high overview of what further happens with the data. First, the data is stored in some storage object, of which three examples are depicted. Then there is a processing layer where anything can be done with the stored data. And finally, the data can be consumed in external BI/AI applications. The key message is that the data lakehouse consists of the storage and processing layer according to this architecture. The emphasis is a good point of this architecture, however, for the rest, we believe it is too high-over to explain on a fine-grained level how the data flows and how it is being processed in a data lakehouse architecture.

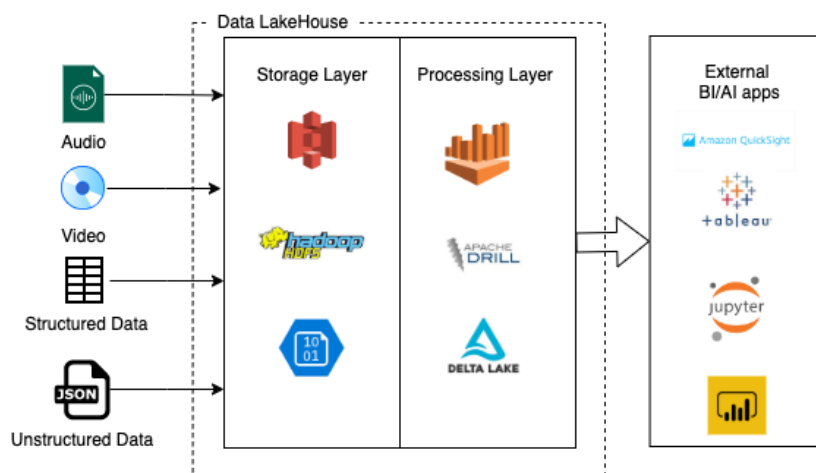


Figure 34: Proposed Data Lakehouse Architecture by Respondent 5

5.3.3. Design of the Data Lakehouse Architecture

After evaluating all the architectures proposed by the respondents and the ones that this study presented, the goal was to capture an overview of the essential parts of a lakehouse architecture. All the respondents were asked to share their views on how they would visualize a lakehouse architecture. Consequently, this led to the following list of requirements:

- **Distributed Storage:** focus on the fact that storage and compute are decoupled
- **Different Data Zones:** implement different data zones in your data lake. Preferably have four zones: landing zone, raw/bronze zone, silver zone, and gold zone. The landing zone can be used to dump any data. The raw/bronze zone is a 1-on-1 copy of the landing zone except for duplicates. So, you have all the data still in their original format and the complete history of the data. However, it is deduplicated in this zone. The silver zone is where transformations take place to clean, filter, update, and enrich the data. Finally, the gold zone is where you have aggregated and structured tables that serve specific business purposes.
- **Different use-cases:** the data lakehouse supports BI analyses and reports and the more modern workloads like data science, machine learning, and streaming use-cases.
- **Data management and governance layer:** the idea of the data lakehouse is to have a data management and governance layer on your data lake. This allows you to manage and govern your data like in a data warehouse, and it distinguishes the data lakehouse from a data lake.
- **Continuous loop between storage and compute:** it is not always clearly shown, but the data is stored, then it is used and processed, then the processed data is stored again, which can be used and processed again, and the processed data is again stored. Thus, this is a loop that happens within the architecture.
- **Link use-case with data zone:** typically, the data engineers utilize data stored in the raw/bronze zone, whereas the business analysts typically utilize the data stored in the golden zone. Hence, for clarification, it would be nice to have this somehow linked to each other.
- **Do not use vendor-specific terms:** to make a generic reference architecture, there should be no vendor-specific services or terms in the architecture.
- **API layer:** the nice thing about the data lakehouse is the possibility of connecting all different kinds of APIs to your platform.

This list of requirements is a summary of what has been discussed during the interviews. In addition to the abovementioned requirements, we decided on the following additional requirements based on the evaluation of 10 architectures (Figure 27 and Appendix G). First of all, we believe that the representation of all the different data types is essential. This contributes to emphasizing the flexibility a data lakehouse offers. Hence, a particular layer is dedicated to the different data formats. Secondly, we prefer to display the storage layer as a data lake to visualize how the data lakehouse incorporates the data lake. Because of this visualization choice, the third decision we made was to structure the architecture from bottom to top. This structure is more convenient when displaying a data lake that takes up space horizontally. Fourthly, Respondent 5 made us aware of the fact that it is important to highlight the data lakehouse concept. Therefore, we have decided to highlight this just as the architectures that Respondent 5 had shared with us.

To summarize, in addition to the eight requirements that we formulated based on the viewpoints of the experts, a total of four requirements are added to the list.

- **Data source layer:** the data lakehouse can handle structured, semi-structured, and unstructured data. Since this part highlights the flexibility of a data lakehouse, a special layer should be dedicated to the different data formats that are supported.

- **Data lake as a storage object:** the data lakehouse concept is a combination of the data warehouse and the data lake. Hence, we think it is necessary to visualize the storage object as a data lake to indicate that the lakehouse utilizes the flat architecture a data lake provides.
- **Bottom-to-top structure:** after evaluating ten different architectures, we have decided to design the architecture from bottom-to-top. This seemed the most logical and preferable way of reading and understanding the flow of a platform architecture.
- **Highlight the data lakehouse concept:** in order to understand that this architecture is specific to a data lakehouse, it is important to highlight what part of the architecture is unique to the lakehouse.

All these requirements and decisions have led to the design of a generic architecture presented in Figure 35. The design of this architecture is based on the viewpoints shared by the experts and our analysis of the ten different architectures. As described in the list of requirements, one of the requirements was to have the use-case linked to the data zone from which data is typically utilized. We have visualized it so that machine learning and data science use-cases typically obtain data from the bronze and silver zone. Whereas the business intelligence and reports use-cases typically utilize data from the gold zone. This is based on a general assumption that all the experts shared. Evidently, this can deviate depending on a specific use case. Thus, business analysts can use data from the silver zone for their reports. The architecture is a representation of how it is in general.

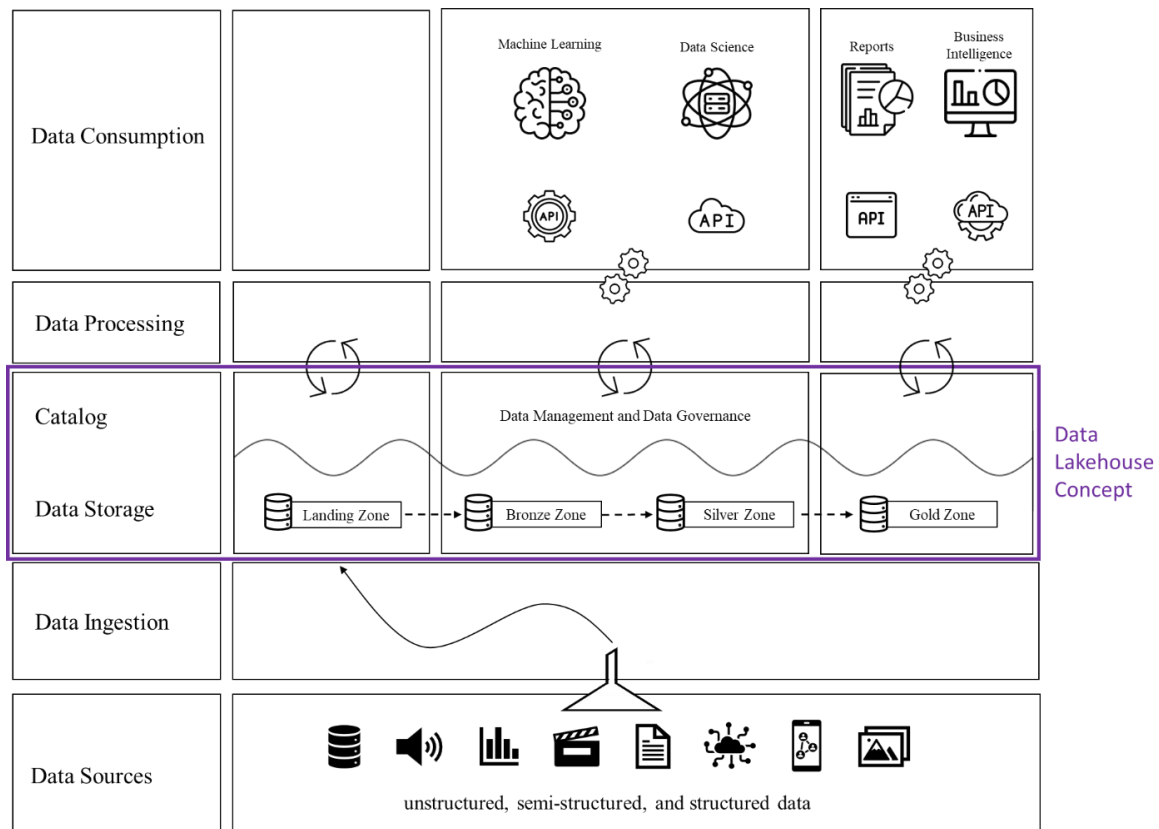


Figure 35: Reference Architecture for a Data Lakehouse

5.3.4. Challenges and Current Shortcomings of the Data Lakehouse

Besides the promising features of a data lakehouse architecture, the experts were also asked to evaluate any shortcomings or challenges they have encountered with the current architecture of the data lakehouse. Their experiences can be classified according to people-related challenges and shortcomings and technical-related challenges and shortcomings. An overview is provided in Table 27.

People-Related Challenges	Technical-Related Challenges	Technical-Related Shortcomings
<ul style="list-style-type: none"> • Slow adoption at the clients • Reluctant to obtain new skills and capabilities • Bad skills = bad implementation • Good understanding of the business problem 	<ul style="list-style-type: none"> • Latency is higher compared to the data lake • Latency is higher due to distributed storage • Latency is higher with small data compared to data warehouse • New technology, no proof and track records 	<ul style="list-style-type: none"> • Integration of governance tools • Implementing security

Table 27: Overview of Challenges and Shortcomings of Data Lakehouses

5.3.4.1. People-Related Challenges

The first people-related challenge that was identified is related to the adoption of the data lakehouse at the client. Multiple respondents indicated that clients are often quite reluctant to change and, in this case, a new technology. Clients are usually apprehensive when it comes to new technologies because they expect new skills and knowledge are needed, and changes will be made to their existing systems. As for data lakehouse, one of the changes is the enablement of using any programming language to query and analyse the data. SQL was used as an official query language in a data warehouse, and generally, everyone is familiar with this. However, this is not always mastered on the client-side regarding the more modern and advanced programming languages. As a result, the clients tend to be reluctant to use the technology because they fear not having the right expertise and skills or that the new technology will not live up to its promises. After all, new skills are required, or they are unwilling to learn new skills. Consequently, it is challenging to advise them when they do not see the value in new technology just because they are reluctant to change and obtain new skills and knowledge.

The second people-related challenge is having people with the right skills and capabilities. As mentioned in the previous paragraph, with new technologies often, new knowledge and skills are required to work with them. As for the data lakehouse, this is also the case to some extent. You would typically need to master SQL and some scripting knowledge for a data warehouse and data lake. Whereas for a data lakehouse, most scripts are running in Python. However, Databricks has tried to make it convenient for every user to utilize the data lakehouse and supports using different programming languages. Despite the flexibility of using different computing languages, most senior programmers are not entirely comfortable with the more advanced programming languages. Hence, finding the right people to be involved with the project is challenging, given that there is often some reluctance to learn new programming languages.

Another challenge identified by the experts is that whenever programmers are involved that do not have the required skills and knowledge, the implementation of the data lakehouse will be of poor quality. However, the challenge here is to have the client realize that this is caused by the lack of skills and knowledge and that it is not a faulty technology. It is more convenient to blame an unsuccessful implementation on the technology than to admit that the right expertise was unavailable in your project team. Hence, it is also quite challenging to make them aware that lacking the right skills will lead to bad implementation. As a matter of fact, the performance of a data lakehouse is the main advantage of a data lakehouse, provided that it is well implemented. Databricks has tried to make it convenient to implement the lakehouse architecture. However, there are still a lot of parameters and settings that need to be configured. This is not yet fully optimized; hence the implementers must understand all the concepts of a data lakehouse to have a successful implementation.

Lastly, a challenge was mentioned that is perhaps not unique to the data lakehouse, although a very common challenge when implementing new technology. Business stakeholders often desire to become more data-driven and thus want to store and analyse the data they possess efficiently. However, this desire does not explicitly say what problem needs to be solved. Consequently, when a data lakehouse is implemented, it will never be as successful as it could have been because it will not solve a specific

problem. This problem was encountered by multiple experts with other technologies as well. However, they noticed this is also the case with new technologies that sound very promising. The new technology promises all different kinds of benefits which seem nice to have and contribute to becoming data-driven. However, it is not solving a specific business problem. Therefore, it is always challenging to identify a client's explicit goals and see how the storage architecture should be developed and implemented accordingly.

5.3.4.2. Technical-Related Challenges

As for the technical-related shortcomings, the biggest challenge is dealing with the latency of a data lakehouse. The challenge of dealing with latency comes in three parts. First of all, compared to a data lake, there is a higher latency with a data lakehouse when records are created and written to the data lakehouse. Secondly, latency also comes in when maintaining distributed storage. Lastly, compared to a data warehouse, the latency in a data lakehouse will be higher when it deals with smaller data sets.

Concerning the comparison between the data lakehouse and the data lake, the latency can be explained by the fact that no transaction logs are made in a data lake. Whereas in a data lakehouse, every transaction that takes place is recorded in a log to make it convenient for people to track any change that was recently made. It can be done instantly when a record is created or updated in a data lake. However, the downside is that no one will know this record was created or updated. Moreover, when a user is looking for specific data, they will have to search through the entire data lake because there is no log stating what data can be found in which dataset. On top of that, there is also no log stating what changes were made to the data. Hence, for any analyses, it is common practice to process the data from the last 24 days to see whether any transactions or changes were recently made. For a data lake, there is no other way to find out. So technically, a data lakehouse will pay itself off quickly since the transaction log can precisely indicate what file has been modified or created and on which day, making it easy to navigate to the needed data.

Secondly, latency comes in with having distributed storage. This aspect has been considered a disadvantage of the data lakehouse. We have already argued that this entirely depends on the underlying storage object and whether the company is distributed worldwide. When the data is stored in a data centre in Northern Europe, and a company's department is located in Australia and wants to query data from this data centre, the latency will be slightly higher than having a data centre closer by. Therefore, the challenge here is implementing a data lakehouse while achieving all the client's needs and keeping the latency in mind.

Lastly, with regard to latency, the latency of a data lakehouse will be higher when processing small data compared to the latency of a data warehouse. However, one could also question why a company has implemented a data warehouse when only small datasets are being processed. Evidently, it depends on what the company is trying to achieve. However, small data sets can also quickly be processed in, for instance, Microsoft Excel. Hence, in that sense, the company would have invested a lot of money in a data warehouse to process small data sets. Anyhow, the fact remains that a data warehouse is more optimized for processing small data than a data lakehouse, and reducing the latency in a data lakehouse is challenging.

Finally, the last technical challenge is not unique to a data lakehouse, although it is now the biggest challenge for implementing a data lakehouse. Namely, when a new technology is introduced, there are no or just a few use-cases to fall back onto and see how specific errors or bugs were dealt with. However, when a new technology is introduced, it always works perfectly on paper, though when it is implemented, there are always new discoveries of issues that need to be fixed or worked around. When the market has adopted the technology for a little longer, there will be more proof and track records to help implement the technology correctly. However, the data lakehouse is still very new; hence these

track records do not exist yet. This is a challenge that has to be dealt with when the lakehouse is implemented.

5.3.4.3. Technical-Related Shortcomings

As of now, there are two shortcomings to the current lakehouse architecture, according to the experts. The first is that there are no build-in governance practices. Consequently, custom solutions are being built to regulate data governance however the integration between this tool and the data lakehouse platform is not optimized and very convenient to implement. In addition to custom-build solutions, third-party solutions like Azure Purview regulate data governance and data lineage that could be integrated. Unfortunately, again the integration between this tool and the lakehouse platform is not as convenient as one would wish. Concludingly, it is a massive shortcoming of the data lakehouse architecture not to have any governance practices build-in. However, Databricks, the only lakehouse vendor, has released multiple articles about working on a ‘unity catalog’ that will be directly integrated with the data lake to allow data governance practices out of the box. This feature sounds very promising however, at the moment, it is still very immature and only available as a private preview for selected clients of Databricks.

The last shortcoming is related to the implementation of security controls. This seems to be a challenge from the moment the big data wave started. Unfortunately, the data lakehouse is also not a magical architecture that suddenly solves all the problems related to securing the data. Still, Databricks has attempted to provide support for typical security measures that a traditional data warehouse brings in, to have more advanced security controls compared to what a data lake can offer. Although, the fact remains that the current architecture of a data lakehouse does not support fine-grained security in terms of having security rules on column-level or row-level.

5.3.5. Future Perspectives of the Data Lakehouse

Two directions were examined to evaluate the future perspectives of the data lakehouse. First of all, during the interview, the final question was whether the expert believed a data lakehouse could replace a data warehouse and/or data lake. The results are presented in Figure 36. All respondents were convinced that the data lake could be replaced by a data lakehouse, given that nothing would be lacking. All the strengths of a data lake are incorporated into the data lakehouse. Hence, the conclusion for whether a data lakehouse can replace the data lake is that it is indeed possible.

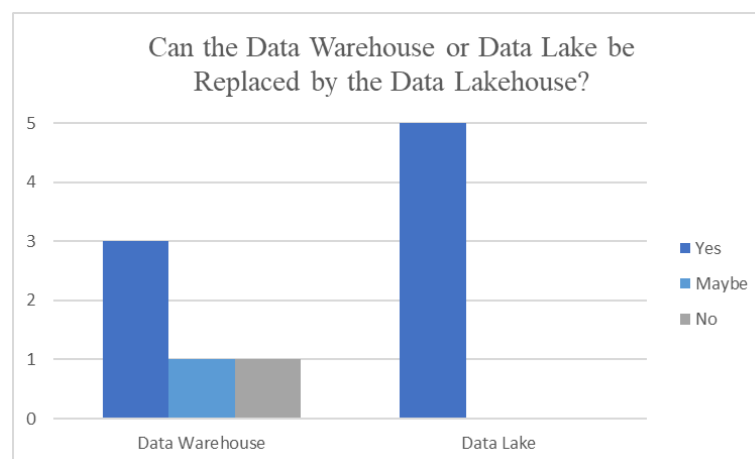


Figure 36: Results on the Question of Whether the Data Lakehouse Can Replace a DW or DL

As for the data warehouse, the experts were divided. One respondent answered with ‘no’ because a data lakehouse is not always better than a data warehouse. For some instances, a data warehouse is more suitable. If more flexibility is required, the respondent indicated that the hybrid two-tier architecture works well enough if it is implemented correctly. Another respondent suggested that the data lakehouse can replace a data warehouse in some cases. Although, it still needs quite some improvement to be able to do so. Hence, this respondent indicated that it is maybe possible in the future. For instance, data

warehouses are optimized for smaller data sets, whereas the data lakehouse is not. Also, converting files in a data warehouse to the right format so they can be stored in the data lake of a lakehouse can be pretty challenging when talking about replacing existing data warehouses. In addition to that, one respondent shared that the data in a data warehouse has likely been transformed into another structure. Consequently, replacing the data warehouse with a data lakehouse requires you to either obtain the data from the original source or go through the history of changes in the data and reverse them. However, that would be too time-consuming and complex to do. Finally, three respondents believe that the data lakehouse can replace a data warehouse from now on and in the future. They acknowledged the fact that it is indeed a challenge to replace data warehouses that are already implemented. However, they believe the data lakehouse can deliver the benefits that a data warehouse would also deliver.

The second direction that was examined for the future perspective of the data lakehouse was examining the assumed following evolutionary step. The concept ‘data mesh’ has gained quite some popularity among researchers in the technology field. Also, the experts that were interviewed mentioned this concept, however, no one has seen a practical implementation of it as of yet. The concept was introduced by Zhamak Dehghani (2019), who presents the idea by underpinning four fundamental principles (Dehghani, 2020):

- **Domain-oriented decentralized data ownership** → this principle mandates that each domain team is responsible for their data. According to this principle, data should be grouped per domain according to the different domains in which teams operate. The reason for this is that all the knowledge about the domain-specific data lies with these domain teams. Hence, they are the best fit to maintain the quality and reliability of the data.
- **Data as a product** → this principle projects a mindset of seeing data as a product. This product thinking philosophy contributes to realizing that there are consumers for the data beyond the business-domain team. Hence, the business-domain team is responsible for satisfying the needs of teams outside their domain and maintaining quality.
- **Self-serve data infrastructure platform** → the idea behind this principle is to adopt platform thinking to data infrastructure. The data infrastructure platform enables other domain teams to consume and create data products seamlessly by considering domain-agnostic functionality, tools, and systems.
- **Federated computational governance** → this principle ensures that interoperability across all data products is achieved in the whole data mesh. The primary purpose of federated governance is to create a data ecosystem that adheres to the rules and regulations of the company and industry.

The rationale behind a data mesh as a platform is that it is “distributed data products oriented around domains and owned by independent cross-functional teams who have embedded data engineers and data product owners, using common data infrastructure as a platform to host, prep, and serve their data assets. The platform is an intentionally designed distributed data architecture, under centralized governance and standardization for interoperability, enabled by a shared and harmonized self-serve data infrastructure (Dehghani, 2019)”. Hence, the idea of what data management means for organizations is redefined (Strenghtolt, 2022). Instead of centrally managing data, specific domain teams become responsible for governing their data.

The way the data lakehouse fits in this story is that the design of the data lakehouse lends itself easily to the idea of having distributed structures and data products. This is supported because all datasets are directly accessible from the storage object, specifically a data lake, without connecting users to the same compute resources (Armbrust et al., 2021). Hence, sharing data is very straightforward and does not depend on which teams produce or consume it. In essence, the data mesh could consist of multiple data lakehouses that are connected. Therefore, the assumption is that the data mesh will be the next storage architecture evolution.

6. Conclusion and Recommendations

This chapter will provide all the main insights that were gathered during the research by answering the research questions. Then, a list of practical recommendations for practitioners will be presented. This is followed by an explanation of threats to validity and how this research has contributed to science and practice. Lastly, the limitations of this study and the future directions will be discussed.

6.1. Research Questions

First, the main research question will be repeated and answered, after which the same will be done for all the sub-questions.

RQ2 “What is the added value of a data lakehouse architecture in your data management platform?”

This study successfully presented a conceptual model that explains the storage architectures' evolution process and each storage architecture's strengths and weaknesses. Moreover, a fine-grained conceptual architecture was developed to show how a data lakehouse is constructed, indicate particular strengths, and explain the uniqueness of the data lakehouse concept. Based on this, the added value of a data lakehouse solution is that it incorporates best practices from data warehouses and data lakes next to the implementation of the Unity Catalog layer. In essence, it supports the storage of all data formats and incorporates best practices of typical database management features to support data management, data governance, and securing the data.

In order to go more into depth on what the added value of a data lakehouse solution is, several sub-questions were formulated. The following paragraphs present a summary of the results for each sub-question.

SQ2.1. “What are the different architectures for data storage solutions?”

Currently, there are three distinct storage architectures: the data warehouse, data lake, and data lakehouse. The evolution of the first storage architecture took place in the early 1980s, when the data warehouse was first introduced. The purpose of this storage architecture was to have a central repository for data coming from different sources and utilize the stored data to enhance and support decision-making processes. A few decades later, society has been making a digital shift, and a Big Data wave started to emerge. Soon after, the very first International Conference on the Internet of Things was held in 2008. Due to the emergence of Big Data and various new data sources, the traditional data warehouse could no longer handle the large volume and variety of data. As a result, the data lake was developed with the rationale of being able to store data in its raw state. Now that unstructured and semi-structured data was also stored, it was possible to perform more modern workloads like machine learning and data science use-cases. As the years progressed, more and more implementations of the data lake faced challenges with handling the reliability and integrity of the stored data. The data lake lacked mature data management, governance and security controls. As a result, the data lakehouse was introduced in 2020 with the rationale of combining best practices of the data warehouse and the data lake.

SQ2.2. “What are the strengths of Data Warehouses, Data Lakes, and Data Lakehouses?”

A data warehouse's unique strengths are that it can process structured data from numerous sources, maintain quality, integrity, and reliability, and support generating reports and business intelligence. Then, a data lake was developed with the rationale of providing everything a data warehouse lacked. Therefore, the data lake is typically known for processing and storing unstructured, semi-structured, and structured data in their raw format in one central repository. As a result, the data lake allows more advanced workloads given that data engineers and scientists can utilize raw data with their preferred analytical tool or framework. In addition to generating business intelligence and reports from the data, it is now possible to do more advanced analyses like performing predictive analytics, finding new patterns while mining through historical data and suggesting prescribed actions to achieve an optimal result. Lastly, the key strength of a data lakehouse is that it combines the best practices of a data

warehouse and a data lake. This means that it has the same flexibility a data lake offers by supporting the storage of various data formats. And it provides the basic data management, governance, and security measures that are also implemented in a data warehouse. This is done by implementing a metadata layer on top of the storage object in the lakehouse, which is, in essence, a data lake.

SQ2.3. “What are the weaknesses of Data Warehouses, Data Lakes, and Data Lakehouses?”

In terms of weaknesses of a data warehouse, the biggest weakness is the lack of flexibility. It cannot process semi-structured, unstructured, and big data, and scaling it up or out is expensive and inconvenient due to the coupled storage and processing capacity. As for the data lake, the main weakness is that there is little to no ability to manage and govern the data properly, and the level of security is considerably weak. Finally, the newest storage architecture was introduced recently, and there are not many evaluation records of the implementation of a data lakehouse. Therefore, one of the main weaknesses of the data lakehouse is that it is still very new. Thus, it has not been fully adopted into the market. Additionally, there are still developments going on to provide, for instance, out-of-the-box data governance or data security tools. Hence, there are numerous benefits and potential benefits to the newest architecture. However, it still needs time to prove itself and fulfil its promises.

SQ2.4. “How can the evolution of the storage architectures and the value of each architecture be explained in a conceptual model?”

Based on the strengths and weaknesses of each storage architecture extracted from literature, a Data Storage Evolution Model was produced. The main distinct strengths and weaknesses are described in the previous two paragraphs. In total, ten advantages and three disadvantages were identified for the data warehouse, five advantages and five disadvantages were identified for the data lake, and thirteen advantages and four disadvantages were found for the data lakehouse. Given that each storage architecture can be seen as a stand-alone object, they were represented in three separate entities in the model. Moreover, the evolution of the storage architectures is entirely based on the rationale of solving and making up for the weaknesses of the precedent architecture. Hence, the model shows how the strengths of a data lake solve the disadvantages of a data warehouse and how the strengths of a data lakehouse solve the disadvantages of a data lake. As a result, the model that was shown in Chapter 4, Figure 10, is a cyclical model with three entities, each representing a storage architecture. On top of the entity, the strengths are displayed, and underneath the weaknesses.

SQ2.5. “How can the designed artefact be improved?”

Five interviews were conducted with experts from the field to improve the Data Storage Evolution Model. By evaluating all the strengths and weaknesses that were included in the model, insights were gathered on whether the experts agreed with what was in there based on their experiences. Additionally, the experts were asked to share their experiences with the design and implementation of each storage architecture to see if novel strengths and weaknesses could be discovered. Lastly, the respondents evaluated the model as a whole and reflected upon the way the relationships between the entities were modelled and how the three entities were put into a cyclical model. Table 28 presents an overview of the number of assumptions that were accepted, adjusted, rejected, and added to the model. The outcome of this evaluation is discussed in Chapter 5, and as a result, a revised model is presented in Section 5.2.3.

	Accepted	Adjusted	Rejected	Added
Data Warehouse	10	2	1	9
Data Lake	6	4	0	7
Data Lakehouse	12	3	2	7

Table 28: Overview of the Evaluation of Assumptions

SQ2.6. “What challenges are faced when utilizing the lakehouse architecture?”

To answer this sub-question, the insights gathered during the interview were used. The lakehouse architecture is still new, hence their experience was only limited to a few use-cases. However, eight

challenges and two shortcomings were identified based on their insights. The challenges were categorized according to 4 people-related challenges and four technical-related challenges, and the two shortcomings are technical-related. The people-related challenges showed that the reluctance to change is quite a big challenge which slows down the adoption of the data lakehouse. Moreover, there is some reluctance to obtain new skills and knowledge, leading to improper implementation of the data lakehouse. This is often blamed on the technology rather than the lack of expertise within the development team. As for the technical-related challenges, they were all related to dealing with the latency of a distributed storage system. Also, given that the data lakehouse combines best practices of a data warehouse and data lake, it tries to present itself as a 'one-size-fits-all' solution. This sometimes leads to lower quality functionality than those solutions built for specific use-cases. Hence, trade-offs must be made, and careful thought must be given to specific design choices. On top of that, given the novelty of this technology, there are no proof and track records to refer to when certain issues occur. Hence, it is always a challenge when new problems and errors occur. Finally, the two technical-related shortcomings are the lack of native integration with governance tools and the lack of fine-grained security.

SQ2.7. "What are the future perspectives with regard to the data lakehouse?"

The answer to this sub-question is two-fold. First of all, during the interviews, the experts were asked to evaluate whether a data lakehouse could replace the data warehouse and the data lake. The respondents all agreed that the data lakehouse could replace the data lake. The currently implemented data lakes can and should be converted, and in the future, the data lakehouse should be consulted instead of the data lake. For the data warehouse, the opinions were slightly divided. One expert does not believe the lakehouse is capable enough to replace the data warehouse. This was mainly based on the fact that a data warehouse outperforms the data lakehouse for specific use-cases. Moreover, this expert believed that the two-tier architecture is sufficient and capable enough to implement the idea of the data lakehouse. Another expert answered that the data lakehouse might be able to replace the data warehouse. Although, the data lakehouse would need a lot of improvements and updates before it can deliver the benefits as good as a data warehouse would deliver. Finally, three experts optimistically answered that a data lakehouse could replace future data warehouse implementations. Converting existing data warehouses is probably too complex. However, in the future, the data lakehouse has the potential to be chosen over a data warehouse. Based on the insights gathered, we believe that the data lakehouse is already capable of replacing the data lake. However, as of now, the data lakehouse is not yet ready to replace the data warehouse as well. When a data lakehouse can provide fine-grained security, reduce its latency when processing small data, and prove to be able to implement adequate data management and governance controls, it will replace the data warehouse.

The second part of the answer to this sub-question focused on the assumed future developments around the data lakehouse and the evolution process of the data storage architectures. The 'data mesh' concept has gained much popularity during the last year, and practitioners are optimistic about this phenomenon. The idea behind the data mesh is to have domain-specific teams governing domain-specific data and see data as a product. This idea fits with the architecture of the data lakehouse; hence the idea is that the data mesh concept can be implemented by implementing multiple data lakehouses, each representing a domain-specific data team. All these lakehouses would then be connected in the 'mesh'. This idea is still very new and conceptual. None of the respondents has seen an actual implementation of this concept. However, the assumption is that the data lakehouse will eventually be used to create a data mesh. We believe it is still too early to speculate about the next generation, given that the data lakehouse is still being improved and not yet fully adopted in the market. We speculate that the data lakehouse will first replace all the data lakes. Next to that, with the introduction of the Unity Catalog and future updates to improve the data lakehouse, we believe the data lakehouse will also become a robust and stable storage architecture. Finally, when the data lakehouse has been fully developed and recognized as a robust and stable storage architecture, it can be used to develop data meshes. Consequently, this might lead to the introduction of the fourth generation of storage architectures: the data mesh.

SQ2.8. “How should a fine-grained architecture for the data lakehouse be constructed to explain its way of working and value?”

Based on the evaluation of ten different architectures and the viewpoints that were gathered from the experts during the interview, a fine-grained conceptual architecture is presented in Section 5.3.3. It consists of six layers: data sources, data ingestion, data storage, catalog, data processing, and data consumption. The data source layer indicates the flexibility of the lakehouse by visualizing different data formats that are either unstructured, semi-structured, or structured. Then the data is ingested through a data ingestion layer and stored in the data storage layer. The storage object for a lakehouse is a data lake which is visualized in the architecture. In this data lake, there are 4 data zones (landing zone, bronze zone, silver zone, and gold zone) that each store data in a different state. On top of this data lake, a data management and governance layer is built, which is presented by the catalog layer. The data storage and the catalog layer together form the data lakehouse concept. Then, there is a processing layer where data can be processed for different use-cases, which are finally presented in the data consumption layer. By connecting different types of APIs, data can be utilized for modern workloads like machine learning and data science or more traditional workloads like business intelligence and reports.

Finally, this research had three research objectives:

Research objective 1: Examine the added value of a data lakehouse architecture

Research objective 2: Develop a conceptual model showing the evolution and value of each storage architecture

Research objective 3: Present a fine-grained conceptual architecture for the data lakehouse

The first research objective is achieved through answering all the sub-questions based on a literature study and conducting interviews. The second research objective was achieved by first developing a conceptual model based on the findings from the literature, after which the model was validated by the experts. As a result, a revised conceptual model of the evolution and strengths and weaknesses of each storage architecture is presented in Figure 25. Lastly, the third research objective was achieved through studying and evaluating ten different architectures and gathering insights and viewpoints from the experts during the interview. As a result, a fine-grained conceptual architecture is presented in Figure 30.

6.2. Recommendations

Given that this research was conducted in collaboration with Avanade, the following recommendations are formulated to provide helpful advice for further consultancy opportunities concerning storage architectures.

First of all, this research has provided many insights and information on the evolution of the storage architectures, particularly the potential of the newest architecture: the data lakehouse. Based on the findings, it can be concluded that implementing the data lakehouse in a data management platform is valuable for the company. When selecting experts for our sample to interview, we noticed a limited group of eligible people due to their knowledge of this topic. Therefore, we recommend Avanade to invest time and resources into this concept. This entails facilitating and providing reimbursement for workshops that their employees can follow. Moreover, they should motivate their employees, or smaller groups of people, to keep invested in this topic. This can be done by allowing employees to go to summits and technology-related events to stay updated with the latest developments. Since Avanade and Databricks are first-degree partners, Avanade already has the privilege to hear about the newest upgrades earlier than their competitors. Additionally, this provides great access to learning paths and certifications that Databricks has to offer with regards to this topic. This can be used to let Avanade employees become experts on this newest storage architecture.

In addition to investing time and resources into enhancing the knowledge of the employees from Avanade, we specifically recommend running pilots and studying different case studies to discover the

potential and challenges of implementing a data lakehouse. Given the interest of the experts that were included in our sample and the enthusiasm and optimism that is shared in published articles and web blogs, it is of high importance to become more familiar with the implementation of the data lakehouse. Knowing what it entails and its benefits are crucial, as well as actual practical experience.

Thirdly, we recommend Avanade to develop a change management plan that will facilitate a smooth transition from existing data lakes to a data lakehouse. Given the unanimous answer to the question of whether a data lakehouse is capable of replacing a data lake, we believe it is of high value to have a complete advisory report on how the transition process works. This possibly enhances and strengthens their current client relations. For the data warehouse, it is slightly more complicated, and not all experts are convinced that a data lakehouse is capable of replacing the data warehouse. Therefore, we recommend Avanade to keep themselves informed of further developments around the data lakehouse. By staying up to date, they will be able to react immediately to certain developments and gain a competitive advantage by having more knowledge.

Finally, it is recommended to utilize this study as a foundation for understanding the differences between the three storage architectures and their value. By knowing how to put each of them into context and understanding the evolution process, the practitioners at Avanade will be better aware of the strengths of each architecture. This will enhance their consults for future clients on which storage architecture to choose.

6.3. Threats to Validity

This research was completed with the utmost integrity, and we minimized the threats to the validity within our power. Nonetheless, we address possible threats to validity in the following paragraphs.

We acknowledge that the sample for this study might be biased because they all have a very similar profile and background and work for the same company. However, the bias has been mitigated in the following ways: 1) by using a structured set of interview questions, 2) by systematically analysing the results with a narrative approach and structuring the results in tables, 3) by validation checks by the researcher during the interviews, and 4) by recording the interviews to ensure the interview results were not based on the recollection of the researcher. This is how the threat to validity regarding the qualitative approach has been mitigated.

A second threat is the validity of the designed Data Storage Evolution Model. By trying to avoid vendor-specific characteristics, extracting information from sources that were perceived as trustworthy, and performing validation interviews with experts whose experience in this field ranges from 6 to 16 years, we have tried to mitigate this threat. However, this threat is not eliminated given that the sample was relatively small and not very diverse. Moreover, only limited resources are published with regards to the data lakehouse. Hence, despite our best efforts to minimise it, there is still a threat to the validity of the Data Storage Evolution Model.

Lastly, there is also a threat to the validity of the produced fine-grained reference architecture. Since experts have not validated this design, it cannot be ensured that this study has presented a valid data lakehouse architecture. However, during the interviews three data lakehouse architectures were evaluated, and seven architectures were introduced by different experts. By combining the insights gathered from the experts and a critical analysis of all ten architectures by the researcher, a new fine-grained architecture was developed. Concludingly, all design choices that led to the creation of the architecture as it is, were validated by five experts and based on the analysis of ten architectures, which mitigates this validity threat to the best of our power.

6.4. Contribution to Science

Currently, the research into the data lakehouse is very limited in the literature. We found five scientific papers and two books that were published about the data lakehouse. These studies referenced each other and articles and blogs that were posted on technology forums. Moreover, no in-depth comparative study examines the evolution of the storage architectures and the rationale behind the evolution process. Next to that, there was no fine-grained architecture of the lakehouse presented in the studies that could be used as a generic reference architecture. First of all, this was due to referencing vendor-specific services. And second of all, the architecture found was a very high-over representation, but it did not emphasize the strength and concept of the lakehouse architecture.

Concludingly, there were numerous knowledge gaps to which this thesis contributes in several ways. First of all, an in-depth comparison is made between the three existing storage architectures. The findings are presented in a conceptual model that visualizes the evolution process and how the weaknesses of a preceding storage solution are solved in the next generation. This model can be used to explain the rationale behind the evolution process, and it can be used for future research to explain the next evolutionary step. Secondly, this study contributes to science by presenting a generic reference architecture that can be used to explain the lakehouse concept on a fine-grained level. Finally, this study also focussed on the challenges and shortcomings of the lakehouse architecture as it is now. This contributes to laying a foundation for future research to see which aspects a data lakehouse could improve.

6.5. Contribution to Practice

Next to contributing to science, this study also contributes to practice. First of all, practitioners can make use of a high overview model that explains the strengths and weaknesses of each storage architecture. This is typically convenient when they are to consult a client on a new project, and there is some hesitation about choosing which architecture. Besides, the generic reference architecture that is presented can be used as a starting point for explaining how a data lakehouse is constructed to both the client and employees who are not familiar with this architecture yet. This contributes to having foundational knowledge that will be necessary to finally build a lakehouse architecture with the Microsoft Azure resources that are available to Avanade. Finally, based on all the findings, this thesis contributes to practice by presenting a set of recommendations. We highly recommend investing time and resources to raise awareness and educate employees. This can be achieved by motivating employees to attend summits and technology-related events that will cover the data lakehouse and encourage them to obtain certifications in this topic. Secondly, we recommend performing pilots of the implementation of a data lakehouse architecture to analyse the performance and discover potential implementation challenges. Finally, this study serves as a starting point to create change management plans on how to perform the transition of a data lake to a data lakehouse.

6.6. Limitations and Future Directions

One of the limitations of this study is the qualitative approach that was chosen. The main challenge with a qualitative study is to minimize bias as much as possible. This is challenging given that interviews are focused on the viewpoints and experiences of the respondents, and the results are evaluated by one researcher who may interpret certain expressions differently than intended. Nevertheless, different measures have been taken like defining structured interview protocols, validation checks during the interview, and recording the interview so that the research did not have to recall everything that was discussed. This was all done to reduce the bias to a minimum. Sections 3.4 and 6.3 give a more in-depth evaluation of how the bias has been reduced to a minimum with our qualitative research approach.

A second limitation is that our sample consisted of five experts. They were interviewed to validate the Data Storage Evolution Model that was designed based on the findings from the literature. However, the revised model has not been validated by the same experts or a completely new sample. Therefore, for future directions, we recommend doing a second round of validation interviews to examine whether the revised model is clear and complete.

In addition to that, the fine-grained reference architecture that was developed has also not been validated through validation interviews. Therefore, it is highly recommended to perform validation interviews with experts who have not been included in the sample for this research. This will reduce the bias of the experts, and they will perhaps come up with new insights that were not shared by the respondents that were included in the sample for this research.

This study provided a comparative view of the strengths and weaknesses of each storage architecture to better understand the rationale behind the newly developed architecture. Given the novelty of the data lakehouse architecture, not many scientific studies were found. Hence, this study has also referred to multiple blogs on technology forums, whitepapers, and blogs published by vendors. This stresses the importance for practitioners to further study the lakehouse concept, which can go in many different directions. For future research, we recommend remaining up to date with the latest publications. In the near future, studies performing case-study analyses might become available. This is especially interesting when the the performance of a data warehouse implementation is compared to the performance of a data lakehouse implementation on the same data. Furthermore, it is likely that this study can be complemented with new information on future developments of the data lakehouse.

Another future research direction would be researching the potential of a data mesh and how this is related to the data lakehouse. Since the assumption is that the data mesh will be the next evolution in the storage architecture, it is highly important to fully understand its specifications and way of working. As a result, our Data Storage Evolution Model can be expanded with an entity describing the data mesh. In that way, a better context is provided on why the data mesh is expected to be following a data lakehouse.

Lastly, based on the insights gathered from literature and the interviews it can be concluded that each technology brings along new challenges. Hence, it should be noted that the data lakehouse is not a magical solution that will not bear any shortcomings or challenges. Therefore, future research should also focus on the challenges and shortcomings we have identified during the interviews. Since one of the shortcomings will probably be resolved by the newest upgrade announced in August 2022, it is highly important to keep up to date with the latest developments and adjust the model and architecture accordingly. The model and architecture that are presented in this research can serve as a foundation and starting point to build further on in future research.

7. References

- [1] Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. In *International Journal of Information Management* (Vol. 49, pp. 424–438). Elsevier Ltd. <https://doi.org/10.1016/j.ijinfomgt.2019.07.008>
- [2] Adelman, S. (2021). *Data Warehouse Costs*. EW Solutions - DataManagementU. <https://www.ewsolutions.com/data-warehouse-costs/>
- [3] Al-Okaily, A., Al-Okaily, M., Teoh, A. P., & Al-Debei, M. M. (2022). An empirical study on data warehouse systems effectiveness: the case of Jordanian banks in the business intelligence era. *EuroMed Journal of Business*. <https://doi.org/10.1108/EMJB-01-2022-0011>
- [4] Alrehamy, H., & Walker, C. (2015). Personal Data Lake With Data Gravity Pull. *IEEE Fifth International Conference on Big Data and Cloud Computin.* <https://doi.org/10.13140/RG.2.1.2817.8641>
- [5] Apache HBase Project docs. (2022, April 18). *Welcome to Apache HBase*. <https://hbase.apache.org/>
- [6] Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Łuszczak, A., Świtakowski, M., Szafranski, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., ... Zaharia, M. (2020). Delta lake: High-Performance ACID Table Storage over Cloud Object Stores. *Proceedings of the VLDB Endowment*, 13(12), 3411–3424. <https://doi.org/10.14778/3415478.3415560>
- [7] Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). *Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics*.
- [8] Avanade Inc. (2022). *Avanade - Our Story*. <https://www.avanade.com/nl-nl/about-avanade/story>
- [9] Berbís, J. M. G., Chagüendo, J. M., Seco, A. A., Sánchez-Segura, M. I., & Domínguez, F. M. (2019). INDIGO: Industrial IoT Data Management and Control Platform based on Semantics. *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, 4240–4246.
- [10] Bourbonnais, P. L., & Morency, C. (2018). A robust datawarehouse as a requirement to the increasing quantity and complexity of travel survey data. *Transportation Research Procedia*, 32, 436–447. <https://doi.org/10.1016/j.trpro.2018.10.054>
- [11] Boyd, R. (2021, February 4). *How Data Lakehouses Solve Common Issues With Data Warehouses*. Engineering Blog on Databricks. <https://databricks.com/blog/2021/02/04/how-data-lakehouses-solve-common-issues-with-data-warehouses.html#:~:text=Data%20staleness%20Because%20the%20data%20warehouse%20is%20populated,out-of-date%20data%2C%20according%20to%20a%20recent%20Fivetran%20survey.>
- [12] Cartledge, C. (2016). *How Many Vs are there in Big Data?*
- [13] Charmaz, K. (2006). *Constructing Grounded Theory: A practical guide through qualitative analysis*. Sage Publications.
- [14] Chen, J., Chen, S., & Rundensteiner, E. A. (2002). A transactional model for data warehouse maintenance. *International Conference on Conceptual Modeling*, 247–262.
- [15] Choi, A., & Shin, H. (2018). *Longitudinal Healthcare Data Management Platform of Healthcare IoT Devices for Personalized Services*.
- [16] Creswell, J. W. (2007). *Qualitative Inquiry & Research Design : Choosing Among Five Approaches*. Sage Publications.
- [17] Databricks. (2020). *Data Lakehouse*. Databricks - Glossary. <https://databricks.com/glossary/data-lakehouse>
- [18] Databricks. (2022, May 10). *ACID Transactions*. <https://databricks.com/glossary/acid-transactions>
- [19] Dehghani, Z. (2019, May 20). *How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh*. <https://martinfowler.com/articles/data-monolith-to-mesh.html>

- [20] Dehghani, Z. (2020, December 3). *Data Mesh Principles and Logical Architecture*. <https://martinfowler.com/articles/data-mesh-principles.html>
- [21] Delgado-Clavero, Á., Gómez-Berbís, J. M., Amescua-Seco, A. de, Sánchez-Segura, M. I., & Medina-Domínguez, F. (2019). Optyfy: Industrial iot-based performance and production optimization based on semantics. *Communications in Computer and Information Science*, 1124 CCIS, 164–177. https://doi.org/10.1007/978-3-030-34989-9_13
- [22] Deng, H., Zhu, Z., & He, Y. (2019). Framework of information data management platform for integrated logistical support of UAS based on military trade mode. In *International Journal of Performability Engineering* (Vol. 15, Issue 5, pp. 1360–1370). Totem Publishers Ltd. <https://doi.org/10.23940/ijpe.19.05.p12.13601370>
- [23] Dixon, J. (2010). *Pentaho, Hadoop, and Data Lakes*. <https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/>
- [24] Dworkin, S. L. (2012). Sample size policy for qualitative studies using in-depth interviews. In *Archives of Sexual Behavior* (Vol. 41, Issue 6, pp. 1319–1320). Springer Science and Business Media, LLC. <https://doi.org/10.1007/s10508-012-0016-6>
- [25] Elmeleegy, H., Li, Y., Qi, Y., Wilmot, P., Wu, M., Kolay, S., Dasdan, A., & Chen, S. (2013). Overview of Turn Data Management Platform for Digital Advertising. *Proceedings of the VLDB Endowment*, 1138–1149.
- [26] Emeakaroha, V. C., Healy, P., Morrison, J. P., & Fatema, K. (2016). Towards a generic cloud-based sensor data management platform: A survey and conceptual architecture. *ResearchGate*. <https://www.researchgate.net/publication/289343405>
- [27] Fan, C., Song, J., Wen, Z., Zhang, X., Wu, Y., & ZOu, J. (2011, February 19). *A Scalable Internet Of Things Lean Data Provision Architecture Based On Ontology*.
- [28] Fang, H. (2015). Managing data lakes in big data era: What's a data lake and why has it become popular in data management ecosystem. *IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 820–824.
- [29] Fivetran. (2020). *Data Analysts: A Critical, Underutilized Resource. A Global Survey of Data and Analytics Professionals*.
- [30] FreshTemp. (2022, April 8). *Cloud-based monitoring platform*. <https://freshtemp.com/>
- [31] Fukuda, A., Nakanishi, T., Hisazumi, K., Kaneko, K., Tagashira, S., Mine, T., Arakawa, Y., Ishida, S., Ando, T., Ashihara, S., Ura, M., Nakamura, Y., Nakamura, S., Kong, W., & Li, G. (2018). Toward Sustainable Smart Mobility Information Infrastructure Platform - Current Status - Current S. *Proceedings - 2018 7th International Congress on Advanced Applied Informatics, IIAI-AAI 2018*, 81–85. <https://doi.org/10.1109/IIAI-AAI.2018.00025>
- [32] Gall, M. D., Borg, W. R., & Gall, J. P. (2003). *Educational Research: An Introduction* (7th ed.). Pearson.
- [33] Galloway, J. M. (2013). *A Cloud Architecture For Reducing Costs In Local Parallel And Distributed Virtualized Cloud Environments*.
- [34] García-Gil, D., Luengo, J., García, S., & Herrera, F. (2017). *Enabling Smart Data: Noise filtering in Big Data classification*. <http://arxiv.org/abs/1704.01770>
- [35] Gardner, S. R. (1998). Building the Data Warehouse. In *COMMUNICATIONS OF THE ACM* (Vol. 41, Issue 9).
- [36] Ghemawat, S., Gobioff, H., & Leung Google, S.-T. (2003). *The Google File System*.
- [37] Giebler, C., Gröger, C., Hoos, E., Schwarz, H., & Mitschang, B. (2019). *Leveraging the Data Lake: Current State and Challenges* (C. Ordonez, I.-Y. Song, G. Anderst-Kotsis, A. M. Tjoa, & I. Khalil, Eds.; Vol. 11708). Springer International Publishing. <https://doi.org/10.1007/978-3-030-27520-4>
- [38] Gosain, A., & Arora, A. (2015). Security issues in data warehouse: A systematic review. *Procedia Computer Science*, 48(C), 149–157. <https://doi.org/10.1016/j.procs.2015.04.164>

- [39] Gupta, H., Xu, Z., & Ramachandran, U. (2018). DataFog: Towards a Holistic Data Management Platform for the IoT Age at the Network Edge. *USENIX Workshop on Hot Topics in Edge Computing*.
- [40] Hasegaw, Y., & Yamamoto, H. (2020). *Highly Reliable IoT Data Management Platform Using Blockchain and Transaction Data Analysis*.
- [41] He, W., Tan, E. L., Lee, E. W., & Li, T. Y. (2009). A solution for integrated track and trace in supply chain based on RFID & GPS. *ETFA 2009 - 2009 IEEE Conference on Emerging Technologies and Factory Automation*. <https://doi.org/10.1109/ETFA.2009.5347146>
- [42] IBM Cloud Education. (2019). *Object Storage*. IBM Cloud Learn Hub. <https://www.ibm.com/cloud/learn/object-storage>
- [43] Inmon, B., Levins, M., & Srivastava, R. (2021). *Building the data lakehouse* (J. Hoberman, Ed.; First Edition). Technics Publications.
- [44] Inmon, W. H. (2002). *Building the Data Warehouse* (R. Elliot, E. Herman, & J. Atkins, Eds.; Third Edition). John Wiley & Sons, Inc.
- [45] Jäger, M., Kamm, L., Krushevskaja, D., Talvik, H.-A., Veldemann, J., Vilgota, A., & Vilo, J. (n.d.). *Flexible Database Platform for Biomedical Research with Multiple User Interfaces and a Universal Query Engine*.
- [46] Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (2002). *Fundamentals of Data Warehouses*. Springer Science & Business Media.
- [47] Kava, P., & Gong, C. (2021, April 28). *Build a Lake House Architecture on AWS*. AWS Analytics Blog. <https://aws.amazon.com/blogs/big-data/build-a-lake-house-architecture-on-aws/>
- [48] Keith D. Foote. (2018, April 19). *A Brief History of the Data Warehouse*. Information Management Articles on Dataversity. <https://www.dataversity.net/brief-history-data-warehouse/#>
- [49] Khine, P. P., & Wang, Z. S. (2018). Data lake: a new ideology in big data era. *ITM Web of Conferences*, 17, 03025. <https://doi.org/10.1051/itmconf/20181703025>
- [50] Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The complete guide to dimensional modeling* (R. Elliot, E. Herman, & J. Atkins, Eds.; Second Edition). John Wiley & Sons Inc.
- [51] Krataithong, P., Anutariya, C., & Buranarach, M. (2021). A data management platform for taxi trajectory-based tourist behavior analysis. *ACM International Conference Proceeding Series*, 15–21. <https://doi.org/10.1145/3444757.3485104>
- [52] Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020). An overview of iot sensor data processing, fusion, and analysis techniques. In *Sensors (Switzerland)* (Vol. 20, Issue 21, pp. 1–23). MDPI AG. <https://doi.org/10.3390/s20216076>
- [53] Kutay, J. (2021, November 15). *Data Warehouse vs. Data Lake vs. Data Lakehouse: An Overview of Three Cloud Data Storage Patterns*. Striim Blog. <https://www.striim.com/blog/data-warehouse-vs-data-lake-vs-data-lakehouse-an-overview/>
- [54] Lavrentyeva, Y., & Sherstnev, A. (2022, February 23). *Cutting through the confusion: data warehouse vs. data lake vs. data lakehouse*. <https://itrexgroup.com/blog/data-warehouse-vs-data-lake-vs-data-lakehouse-differences-use-cases-tips/#>
- [55] Li, X., Zhou, C., Gao, X., Han, Y., & Yang, M. (2020). Architecture Design of Cryptographic Data Management Platform Based on Hadoop. *IEEE International Conference on Artificial Intelligence and Computer Applications*, 691–694.
- [56] Li, X., Zhou, C., Gao, X., Han, Y., & Yang, M. (2021). Architecture Design of Joint Operation Data Management Platform for Space-Ground Integrated Information Network. *Proceedings of 2021 IEEE International Conference on Data Science and Computer Application, ICDSCA 2021*, 862–865. <https://doi.org/10.1109/ICDSCA53499.2021.9650196>
- [57] Liu, K., & Dong, L. J. (2012). Research on cloud data storage technology and its architecture implementation. *Procedia Engineering*, 29, 133–137. <https://doi.org/10.1016/j.proeng.2011.12.682>

- [58] Lorica, B., Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2020, January 30). *What is a Data Lakehouse?* Databricks Engineering Blog.
- [59] Loxton, G. (2022, March 18). *The Data Lakehouse, the Data Warehouse and the Modern Data Platform Architecture*. Microsoft Tech Community .
<https://techcommunity.microsoft.com/t5/azure-synapse-analytics-blog/the-data-lakehouse-the-data-warehouse-and-a-modern-data-platform/ba-p/2792337>
- [60] Lu, J., Liu, Z. H., Xu, P., & Zhang, C. (2016). *UDBMS: Road to Unification for Multi-model Data Management*. <http://arxiv.org/abs/1612.08050>
- [61] Ma, R., Li, W., Ma, N., Zhang, X., & Zhang, H. (2020). Design and Research of Big Data Platform Framework for Power Enterprises. *IOP Conference Series: Earth and Environmental Science*, 529(1). <https://doi.org/10.1088/1755-1315/529/1/012009>
- [62] Madera, C., & Laurent, A. (2016). The next information architecture evolution: The data lake wave. *8th International Conference on Management of Digital EcoSystems, MEDES 2016*, 174–180. <https://doi.org/10.1145/3012071.3012077>
- [63] Marks, E. A., & Lozano, B. (2010). *Executive's guide to cloud computing*. Wiley.
- [64] Mason, M. (2010). *Sample Size and Saturation in PhD Studies Using Qualitative Interviews*. <http://www.qualitative-research.net/>
- [65] McNamara, C. (2006). General Guidelines for Conducting Research Interviews. In *Field Guide to Consulting and Organizational Development: A Collaborative and Systems Approach to Performance, Change and Learning*. Authenticity Consulting, LCC.
- [66] Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., Tekes, S., & Riekk, J. (2019). Implementing big data lake for heterogeneous data sources. *Proceedings - 2019 IEEE 35th International Conference on Data Engineering Workshops, ICDEW 2019*, 37–44. <https://doi.org/10.1109/ICDEW.2019.00-37>
- [67] Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*.
- [68] Microsoft Azure. (n.d.-a). *Azure Data Explorer*. Retrieved April 21, 2022, from <https://azure.microsoft.com/en-us/services/data-explorer/#overview>
- [69] Microsoft Azure. (n.d.-b). *NoSQL Database - What is NoSQL?* Retrieved April 21, 2022, from <https://azure.microsoft.com/en-us/overview/nosql-database/>
- [70] Microsoft Azure. (2022). *What is Cloud Computing?* . A Beginner's Guide. <https://azure.microsoft.com/en-gb/overview/what-is-cloud-computing/#:~:text=Simply%20put%2C%20cloud%20computing%20is%20the%20delivery%20of,faster%20innovation%2C%20flexible%20resources%20and%20economies%20of%20scale.>
- [71] Microsoft Corporation. (2022, April 13). *What is a data management platform (DMP)?* <https://dynamics.microsoft.com/en-us/ai/customer-insights/what-is-a-data-management-platform-dmp/#:~:text=What%20Is%20a%20Data%20Management%20Platform%20%7C%20Microsoft,sources%20and%20put%20it%20into%20a%20usable%20form.>
- [72] Microsoft Docs. (2021a, January 28). *What is Apache HBase in Azure HDInsight*. <https://docs.microsoft.com/en-us/azure/hdinsight/hbase/apache-hbase-overview>
- [73] Microsoft Docs. (2021b, October 7). *Introduction to Azure Data Lake Storage Gen2*. <https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction>
- [74] Microsoft Docs. (2021c, November 12). *Choose a Big Data Storage Technology in Azure*. <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-storage>
- [75] Microsoft Docs. (2022a, March 16). *Introduction to Azure Blob Storage*. <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>
- [76] Microsoft Docs. (2022b, April 10). *Welcome to Azure Cosmos DB*. <https://docs.microsoft.com/en-us/azure/cosmos-db/introduction>
- [77] MicroStrain. (2022, April 8). *MicroStrain SensorCloud Platform*. <http://www.sensorcloud.com/>

- [78] Motta, G., Sacco, D., Ma, T., You, L., & Liu, K. (2015). Personal mobility service system in urban areas: The IRMA project. *Proceedings - 9th IEEE International Symposium on Service-Oriented System Engineering, IEEE SOSE 2015*, 30, 88–97. <https://doi.org/10.1109/SOSE.2015.15>
- [79] Nind, T., Galloway, J., McAllister, G., Scobbie, D., Bonney, W., Hall, C., Tramma, L., Reel, P., Groves, M., Appleby, P., Doney, A., Guthrie, B., & Jefferson, E. (2018). The research data management platform (RDMP): A novel, process driven, open-source tool for the management of longitudinal cohorts of clinical data. In *GigaScience* (Vol. 7, Issue 7). Oxford University Press. <https://doi.org/10.1093/gigascience/giy060>
- [80] Obrutsky, S. L. (2016). Cloud Storage: Advantages, Disadvantages and Enterprise Solutions for Business. *Conference: EIT New Zealand*. <https://www.researchgate.net/publication/305508410>
- [81] Oracle. (2022, April 13). *What is data management platform (DMP)?* <https://www.oracle.com/cx/marketing/data-management-platform/what-is-dmp/>
- [82] Orescanin, D., & Hlupic, T. (2021). Data Lakehouse - A Novel Step in Analytics Architecture. *2021 44th International Convention on Information, Communication and Electronic Technology, MIPRO 2021 - Proceedings*, 1242–1246. <https://doi.org/10.23919/MIPRO52101.2021.9597091>
- [83] Ostia. (2022, April 8). *Portus Platform*. <https://www.ostiasolutions.com/index.php/product/platform-overview>
- [84] Panyapiang, T. (2022, February 12). *A Gentle Introduction to Data Lakehouse*. Towards Data Science. <https://towardsdatascience.com/a-gentle-introduction-to-data-lakehouse-fc0f131f90ff>
- [85] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- [86] Peili, Y., Xuezheng, Y., Jian, Y., Lingfeng, Y., Hui, Z., & Jimin, L. (2018). Deep learning model management for coronary heart disease early warning research. *2018 3rd IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA 2018*, 552–557. <https://doi.org/10.1109/ICCCBDA.2018.8386577>
- [87] Philip Chen, C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314–347. <https://doi.org/10.1016/j.ins.2014.01.015>
- [88] Poppendieck, M. (2011). Principles of Lean Thinking. In *IT Management Select*.
- [89] Prelipcean, A. C., Gidófalvi, G., & Susilo, Y. O. (2018). MEILI: A travel diary collection, annotation and automation system. *Computers, Environment and Urban Systems*, 70, 24–34. <https://doi.org/10.1016/j.compenvurbsys.2018.01.011>
- [90] Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud Computing: An Overview. In *LNCS* (Vol. 5931).
- [91] Ravat, F., & Zhao, Y. (2019). Data Lakes: Trends and Perspectives. *International Conference on Database and Expert Systems Applications*, 304–314. <https://doi.org/10.1007/978-3-030-27615-7>
- [92] Ren, L., Zhang, Z., Zhao, C., & Zhang, G. (2020). Cloud-Based Master Data Platform for Smart Manufacturing Process. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 322 LNICST, 163–170. https://doi.org/10.1007/978-3-030-48513-9_13
- [93] Rifaie, M., Alhajj, R., & Ridley, M. (2009). Data Governance Strategy: A Key Issue in Building Enterprise Data Warehouse. In G. Kotsis (Ed.), *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services* (pp. 587–591).
- [94] Ritchie, J., Lewis, J., & Elam, G. (2003). Designing and Selecting Samples. In *Qualitative research methods* (pp. 77–108). CA: Sage.

- [95] Roelofs, E., Persoon, L., Nijsten, S., Wiessler, W., Dekker, A., & Lambin, P. (2013). Benefits of a clinical data warehouse with data mining tools to collect data for a radiotherapy trial. *Radiotherapy and Oncology*, 108(1), 174–179. <https://doi.org/10.1016/j.radonc.2012.09.019>
- [96] Rooney, S., Bauer, D., Garces-Erice, L., Urbanetz, P., Froese, F., & Tomic, S. (2019). Experiences with managing data ingestion into a corporate datalake. *Proceedings - 2019 IEEE 5th International Conference on Collaboration and Internet Computing, CIC 2019*, 101–109. <https://doi.org/10.1109/CIC48465.2019.00021>
- [97] Rosenthal, A., & Sciore, E. (2000). View Security as the Basis for Data Warehouse Security. In M. Jeusfeld, H. Shu, M. Staudt, & G. Vossen (Eds.), *International Workshop on Design and Management of Data Warehouses*. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/>
- [98] Ruan, S., Li, R., Bao, J., He, T., & Zheng, Y. (n.d.). *CloudTP: A Cloud-based Flexible Trajectory Preprocessing Framework*.
- [99] Segal, M., & Team Zuar. (2021, December 28). *Data Mart vs Data Warehouse vs Database vs Data Lake*. <https://www.zuar.com/blog/data-mart-vs-data-warehouse-vs-database-vs-data-lake/#:~:text=Data%20Lake%20vs.%20Data%20Mart%20The%20key%20differences,structure d%20essential%20data%20for%20a%20department%20or%20function.>
- [100] Sheta, O. E., & Nour Eldeen, A. N. (2013). Evaluating a Healthcare Data Warehouse For Cancer Diseases. *International Journal of Computer Science and Information Technology & Security*, 3(3), 2249–9555.
- [101] Shvachko, K., Kuang, H., Radia, S., Presented, R. C., & Pokluda, A. (2010). THE HADOOP DISTRIBUTED FILE SYSTEM. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10.
- [102] Singh, J. (2021, March 5). *History of Cloud Computing*. <https://www.geeksforgeeks.org/history-of-cloud-computing/>
- [103] Strengtholt, P. (2022, May 24). *Data Mesh: Topologies and Domain Granularity*. Towards Data Science. <https://towardsdatascience.com/data-mesh-topologies-and-domain-granularity-65290a4ebb90>
- [104] TempoDB. (2022, April 8). *The Time Series Database Service*. <https://tempo-db.com/>
- [105] Turner III, D. W., & Hagstrom-Schmidt, N. (2010). *Qualitative Interview Design: A Practical Guide for Novice Investigators: Vol. 15(3)*.
- [106] Vaduva, A., & Vetterli, T. (2001). Metadata Management for Data Warehouse: An Overview. *International Journal of Cooperative Information Systems*, 10(3), 273–298. www.worldscientific.com
- [107] Vishnu, B., Manjunath, T. N., & Hamsa, C. (2014). An Effective Data Warehouse Security Framework. *International Journal of Computer Applications*.
- [108] W3Schools. (2022). *History of Cloud Computing*. <https://www.w3schools.in/cloud-computing/history-of-cloud-computing>
- [109] Wang, J., Fang, J., & Han, Y. (2013). A multi-source data organization and management method for intelligent transportation. *Proceedings - 2013 10th Web Information System and Application Conference, WISA 2013*, 324–327. <https://doi.org/10.1109/WISA.2013.67>
- [110] Wang, J., Liu, J., Li, J., Zhang, J., & Lei, Q. (2017). *Design and Implementation of Medical Data Management System* (B. Zou, M. Li, H. Wang, X. Song, W. Xie, & Z. Lu, Eds.; Vol. 727). Springer Singapore. <https://doi.org/10.1007/978-981-10-6385-5>
- [111] Wang, L., & von Laszewski, G. (2008). Scientific Cloud Computing: Early Definition and Experience. In *2008 10th Ieee International Conference on High Performance Computing and Communications*, 825–830.
- [112] Watson, H. J., Goodhue, D. L., & Wixom, B. H. (2002). The benefits of data warehousing: Why some organizations realize exceptional payoffs. *Information and Management*, 39(6), 491–502. [https://doi.org/10.1016/S0378-7206\(01\)00120-3](https://doi.org/10.1016/S0378-7206(01)00120-3)
- [113] Watson, T. W. (1997). *Guidelines For Conducting Interviews*.

- [114] Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer.
- [115] Xue, C. T. S., & Xin, F. T. W. (2016). Benefits and Challenges of the Adoption of Cloud Computing in Business. *International Journal on Cloud Computing: Services and Architecture*, 6(6), 01–15. <https://doi.org/10.5121/ijccsa.2016.6601>
- [116] Yang, S., & Hu, J. (2021). Research on Intelligent Management Platform for Printing Equipment Fault. *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer, ICFTIC 2021*, 665–668. <https://doi.org/10.1109/ICFTIC54370.2021.9647288>
- [117] Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., & Vasilakos, A. v. (2016). Big data: From beginning to future. In *International Journal of Information Management* (Vol. 36, Issue 6, pp. 1231–1247). Elsevier Ltd. <https://doi.org/10.1016/j.ijinfomgt.2016.07.009>
- [118] You, L., Zhao, F., Cheah, L., Jeong, K., Zegras, P. C., & Ben-Akiva, M. (2020). A Generic Future Mobility Sensing System for Travel Data Collection, Management, Fusion, and Visualization. *IEEE Transactions on Intelligent Transportation Systems*, 21(10), 4149–4160. <https://doi.org/10.1109/TITS.2019.2938828>
- [119] Zaharia, M., Keller, J., Pappa, Z., & Thakur, S. (2022, April 20). *Announcing Gated Public Preview of Unity Catalog on AWS and Azure*. Databrick Platform . <https://databricks.com/blog/2022/04/20/announcing-gated-public-preview-of-unity-catalog-on-aws-and-azure.html>
- [120] Zhang, Z., & Li, F. (2019). Research on the Construction of Big Data Management Platform of Shuohuang Railway Locomotive Operation and Maintenance. In Y. Qin, L. Jia, B. Liu, Z. Liu, L. Diao, & M. An (Eds.), *Lecture Notes in Electrical Engineering 639 Proceedings of the 4th International Conference on Electrical and Information Technologies for Rail Transportation (EITRT) 2019 Rail Transportation System Safety and Maintenance Technologies*. <http://www.springer.com/series/7818>

8. Appendices

Appendix A: Interview Questions

Intro

- Permission for recording & Results will be anonymized
- Introduction
 - Master thesis for MSc Business Information Technology
 - Topic data management platform, specifically storage solutions
 - Scope: research the potential of a lakehouse
- Purpose interview: evaluate findings from literature and discover insights from practice

Data warehouse

1. What are the main advantages of a data warehouse according to you?
2. Do you agree with what is included in the model?
3. What are the main disadvantages of a data warehouse according to you?
4. Do you agree with what is included in the model?
5. What would you change/add/eliminate from the model?

Data lake

6. What are the main advantages of a data lake according to you?
7. Do you agree with what is included in the model?
8. What are the main disadvantages of a data lake according to you?
9. Do you agree with what is included in the model?
10. What would you change/add/eliminate from the model?

Lakehouse

11. Do you agree with the advantages mentioned in the model?
12. Do you agree with the disadvantages mentioned in the model?
13. What would you change/add/eliminate from the model?

Artefact

14. Do you find the model easy to interpret?
15. Do you have any suggestions on changing the relationships?

Lakehouse

Experience

16. What are your experiences with the lakehouse?
17. What challenges have you encountered with the lakehouse architecture?

Part 2 – Architecture

18. Could you describe the layers or components of the lakehouse architecture according to you?
19. Are there any shortcomings to the current lakehouse architecture?
20. Do you think the lakehouse architecture can be improved? (e.g., adding a component, new features, layers, etc.)
21. Which architecture visualizes the architecture of a lakehouse best?
22. When looking at literature, *security* is a fundamental aspect and should be well organised. There seems to be no component or layer in the architecture that deals with this. What are your thoughts on this?
23. Another vital aspect is data governance, more specifically *data lineage*. A lakehouse seems to have taken this into account but doesn't specify how. What are your thoughts on this?
24. When looking at the architectures found in literature, the API layer seems something new and essential, could you describe the benefit of this layer and how it works?
















Future perspective

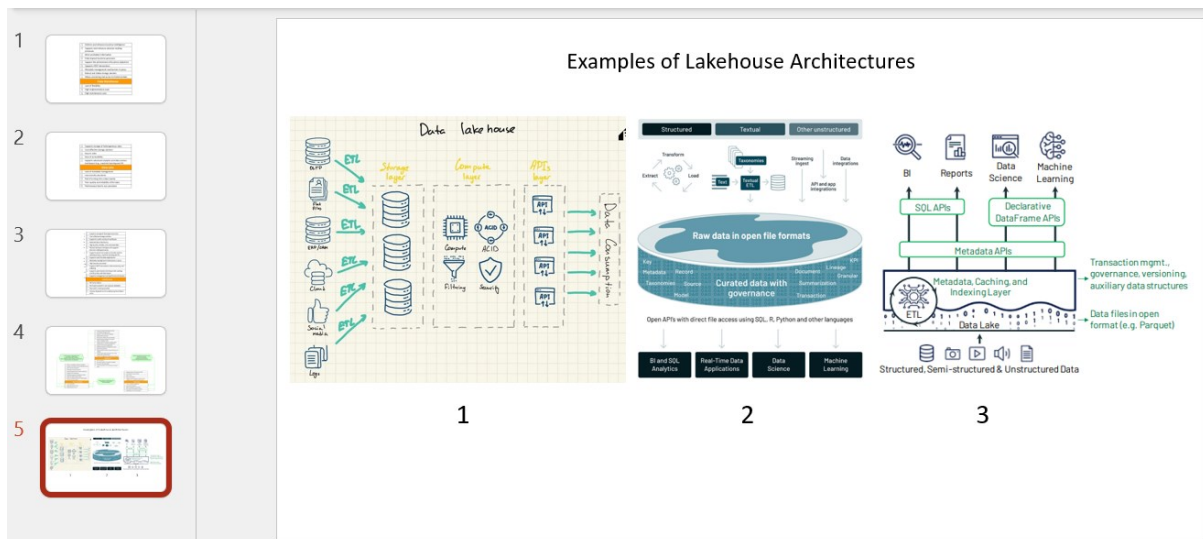
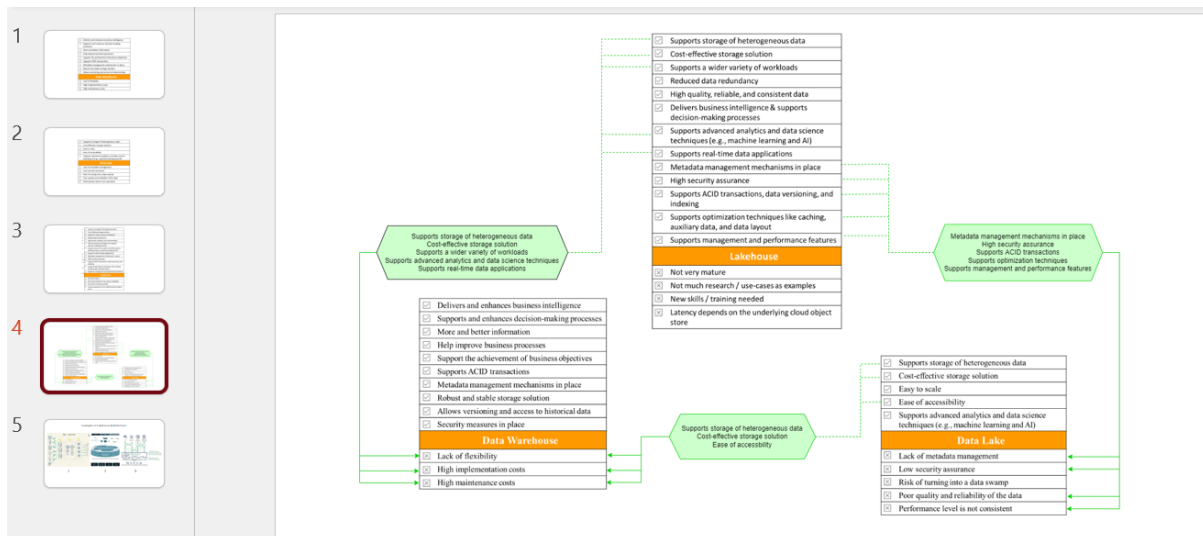
25. Do you think a Lakehouse is able to replace the data lakes and/or data warehouses? If so, how would this migration look like?

Outro

26. Do you have something you want to share that we didn't discuss yet?
27. Do you have any questions for me?

Appendix B: Slides Used During Interviews

1			<input checked="" type="checkbox"/> Delivers and enhances business intelligence <input checked="" type="checkbox"/> Supports and enhances decision-making processes <input checked="" type="checkbox"/> More and better information <input checked="" type="checkbox"/> Help improve business processes <input checked="" type="checkbox"/> Support the achievement of business objectives <input checked="" type="checkbox"/> Supports ACID transactions <input checked="" type="checkbox"/> Metadata management mechanisms in place <input checked="" type="checkbox"/> Robust and stable storage solution <input checked="" type="checkbox"/> Allows versioning and access to historical data <input checked="" type="checkbox"/> Security measures in place
2			
3			
4			
5			
			Data Warehouse <input type="checkbox"/> Lack of flexibility <input type="checkbox"/> High implementation costs <input type="checkbox"/> High maintenance costs
1			
2			<input checked="" type="checkbox"/> Supports storage of heterogeneous data <input checked="" type="checkbox"/> Cost-effective storage solution <input checked="" type="checkbox"/> Easy to scale <input checked="" type="checkbox"/> Ease of accessibility <input checked="" type="checkbox"/> Supports advanced analytics and data science techniques (e.g., machine learning and AI)
3			
4			
5			
			Data Lake <input type="checkbox"/> Lack of metadata management <input type="checkbox"/> Low security assurance <input type="checkbox"/> Risk of turning into a data swamp <input type="checkbox"/> Poor quality and reliability of the data <input type="checkbox"/> Performance level is not consistent
1			<input checked="" type="checkbox"/> Supports storage of heterogeneous data <input checked="" type="checkbox"/> Cost-effective storage solution <input checked="" type="checkbox"/> Supports a wider variety of workloads <input checked="" type="checkbox"/> Reduced data redundancy <input checked="" type="checkbox"/> High quality, reliable, and consistent data <input checked="" type="checkbox"/> Delivers business intelligence & supports decision-making processes <input checked="" type="checkbox"/> Supports advanced analytics and data science techniques (e.g., machine learning and AI) <input checked="" type="checkbox"/> Supports real-time data applications <input checked="" type="checkbox"/> Metadata management mechanisms in place <input checked="" type="checkbox"/> High security assurance <input checked="" type="checkbox"/> Supports ACID transactions, data versioning, and indexing <input checked="" type="checkbox"/> Supports optimization techniques like caching, auxiliary data, and data layout <input checked="" type="checkbox"/> Supports management and performance features
2			
3			
4			
5			
			Lakehouse <input type="checkbox"/> Not very mature <input type="checkbox"/> Not much research / use-cases as examples <input type="checkbox"/> New skills / training needed <input type="checkbox"/> Latency depends on the underlying cloud object store



Appendix C: Suggested Additions of Experts for Each Entity

Respondent	Data warehouse	Data lake	Data lakehouse
1	<ul style="list-style-type: none"> • Add advantage that it is possible to have foreign keys in your data warehouse (this is not properly implemented in lakehouse and data lake). • Add the disadvantage that the storage capacity is linked with the processing capacity. As a result of this, there is a lack of flexibility and high costs are involved. 	<ul style="list-style-type: none"> • Add disadvantage or advantage depending on how you see it: data lake is just storage. The data lake does not take into account the processing. • Add disadvantage or advantage depending on how you see it: there are so many implementations of the data lake. It is like the wild west. Everyone comes up with their creation. • Combine the disadvantage of the data swamp and the quality and reliability of the data. And make 1 disadvantage of this. • Add the disadvantage that the latency depends on the underlying storage object. 	<ul style="list-style-type: none"> • Add disadvantage that explains there is only one implementation of the lakehouse provided by Databricks. Even though it is open-sourced, the adoption is still very new and Databricks is the only official lakehouse provider. So you don't have many other robust solutions for working with a data lakehouse apart from Databricks. • Add disadvantage that it does not implement foreign keys.
2	<ul style="list-style-type: none"> • Add advantage that it is optimized for small data sets. • Add the disadvantage that scalability is not really possible. 	-	-
3	<ul style="list-style-type: none"> • Add advantage that the performance of a data warehouse for structured data is awesome. • Add the advantage that the data warehouse can show the results of your DDL and DML 	<ul style="list-style-type: none"> • Considering a focus on Azure: Add the advantage that the azure data lake has native integration with Azure active directory which helps with managing the permissions. The lakehouse is not there 	<ul style="list-style-type: none"> • Add advantage that it is open-source. The data format Delta is open-sourced. • Add the advantage that you can use any language you like interchangeably.

	<p>queries to a local session.</p> <ul style="list-style-type: none"> • Add disadvantage that it is difficult to upscale and out scale. • Add the disadvantage that the data format in the storage is proprietary. It is not open-sourced. • Add disadvantage that it is not suitable for streaming use-cases. 	<p>yet with this integration.</p> <ul style="list-style-type: none"> • Add the disadvantage that auditing and logging on the data lake are not available out of the box. • Add the disadvantage that a backup solution is not available out of the box on data lakes. To be more specific, snapshots and point-in-time-restore are not there yet. • Add the disadvantage that the data governance is not yet there. 	<ul style="list-style-type: none"> • Perhaps give a thought to this: Databricks has announced the release of the unity catalog. With this, you will have data governance out of the box. It is only not there yet. Perhaps, add this as a potential advantage for the future. • Considering a focus on Azure: add the disadvantage that the lakehouse does not have a native integration with azure active directory to handle security.
4	<ul style="list-style-type: none"> • Add advantage that data warehouses are schema on write. However, this can also be a disadvantage because changing the structure is costly and complex. • Add advantage that data governance can be applied. • Add disadvantage is that it is often proprietary storage. • Add the disadvantage that there is limited support for modern data access like streaming, APIs, and data science. You could also phrase it as limited use-case support for modern workloads. 	<ul style="list-style-type: none"> • Add the advantage that you have a decoupling of storage and compute. • Add the disadvantage that data governance is a huge problem. • Add the disadvantage that GDPR and PII are problematic topics for a data lake. 	<ul style="list-style-type: none"> • Add disadvantage that the lakehouse is a very vendor-specific thing. It is a marketing term from Databricks. And you are immediately choosing Databricks as your vendor.
5	<ul style="list-style-type: none"> • Add the advantage that you can have good data governance 	<ul style="list-style-type: none"> • Add the disadvantage that GDPR rules and regulations are not there in a data lake. 	<ul style="list-style-type: none"> • Add the disadvantage that the lakehouse has nothing build-in yet for data governance.

	<p>in your data warehouse.</p> <ul style="list-style-type: none"> • Add advantage is that when you implement a data warehouse, a data quality framework is implemented. This checks the data quality. • Add the disadvantage that it is difficult to add another piece to your data warehouse. 	<p>The GDPR rules are actually lacking, and the governance part is really bad.</p>	<p>Databricks is busy with a unity catalog but that is not on a mature level yet.</p> <ul style="list-style-type: none"> • Add the disadvantage that you cannot have fine-grained security on your lakehouse. It is the same as for a data lake actually.
--	--	--	--

Table 32: Suggested Additions of Experts for Each Entity

Appendix D: Suggestions of Experts on Improving the Model

	Suggestions	Summarized Theme
Respondent 1	<p>“One thing that would help is making the difference between plusses and minuses more visible or clearer. This is not immediately clear only when you read what is in the model.”</p> <p>“Some plusses and minuses are purely technical, some are dependent on the underlying technology, and some are fully dependent on how people are used to working with it and how skilled they are to work with it. Maybe you could make a distinguishment between technical- and process-related or people-related plusses and minuses”.</p>	<ul style="list-style-type: none"> ❖ Highlight plusses and minuses (colouring) ❖ Technical vs. process/people
Respondent 2	<p>“You should keep in mind that technically, a data lake is basically just storage. Data warehouses are single entities whereas a data lake is a storage part + processing part. So you have to split it among the model. For instance by having two boxes in your model one for storage and one for computing. Some problems of a data lake, for instance, are related to storage and how the data is managed in storage. But you also have problems that depend on performance which is part of the computing part. The same goes for lakehouse, this is also a storage solution + compute solution.”</p> <p>“If you have this separation of storage and compute, I would specify which attribute (advantage or disadvantage) belongs to which technology. And then clearly indicate which plusses of one technology solve the minuses of the other technology.”</p> <p>“The model could maybe be from left to right instead of a circle. Because the data lake is an evolution of the data warehouse, and the lakehouse is an evolution of the data lake.”</p>	<ul style="list-style-type: none"> ❖ Storage vs. compute ❖ Highlight which plusses solve which minuses of the other technology ❖ Evolution from left to right
Respondent 3	<p>“You should keep in mind to include storage AND compute. We are talking about a data management platform. Just with storage, you cannot do something. So we also need the compute part. Technically a data lake is just storage, but also look at the compute part.”</p> <p>“For the rest, the entities are easy to interpret. The relationships, however, are a bit difficult to understand immediately just by looking at the model. But with an explanation it is understandable. Maybe this could be visualised in a better or easier way”</p>	<ul style="list-style-type: none"> ❖ Storage vs. compute ❖ Highlight your message
Respondent 4	<p>“Interesting diagram. This is a relationship I didn’t really consider before. I first thought of it as a linear relationship but I see what you are trying to do”.</p>	<ul style="list-style-type: none"> ❖ Reverse relationships from disadvantage to advantage

	<p>“For me, it would be logical to use a disadvantage as a starting point. So let’s say you have this disadvantage, let’s look at where it will be solved. So the diagram is in reverse for me.”</p> <p>“Think about the message you want to bring across. Perhaps by highlighting certain aspects you can bring the message even better. Which problems are solved by the subsequent generation, and which new problems are introduced in the subsequent generation? Also, in practice, we would never say this client NEEDS a data warehouse and nothing else. The message should say that a data warehouse has these specific shortcomings which would be solved by this generation but that one has its shortcomings. I am not sure if this will lead to a cyclical diagram.”</p> <p>“Try to link the storage architectures based on how they solve which problems over time. For example, a data warehouse tried to solve the problems people had. But over time, the people’s demands became more complex. As a result, the subsequent generation was developed. And so on.”</p>	<ul style="list-style-type: none"> ❖ Highlight your message ❖ Evolution from left to right
Respondent 5	<p>“I think the model is fine, but the message is not coming over. I would recommend doing something with colouring to emphasize your message.”</p> <p>“Maybe you can make a high-level model and a detailed model. A high-over model with just 3 circles that each represent a storage architecture, and then in the overlapping areas indicate what they are overlapping. Use colouring for this. And then have a detailed model with the same colouring where you try to explain how each generation tries to solve problems of the previous generation.”</p>	<ul style="list-style-type: none"> ❖ Highlight your message ❖ Colouring ❖ High-level model and detailed-model

Table 33: Suggestions of Experts on Improving the Model

Appendix E: Expert's Evaluation of Existing Architectures

E.1. Respondent 1

	Observations
General	In general, all three architectures represent a different perspective, so each architecture has a different function.
Architecture 1	This pretty much contains everything that should be in there. It is also pretty similar to another architecture that I have seen. This one best represents the data lakehouse in my opinion.
Architecture 2	This reflects more on the data lake than the data lakehouse. It just says we have different types of data that are put in storage in raw format and it can be read by anything. So mostly, it represents the data lake
Architecture 3	This also sort of represents the data lake. You do have a metadata layer now, so in that sense, it can be a lakehouse because this is what separates a data lake from a lakehouse, but it is still a monolithic thing. It is not a fine-grained architecture
Additional Architecture (Figure 28)	This architecture is very functional and describes all the processes. However, this is published by Microsoft and they want to sell their products. So, the entire architecture consists of services that they provide. That is something I don't like about it. however, the structure and the 'back-bone' so to say is a very good example of the data lakehouse architecture.
Additional Architecture (Figure 29)	<p>This architecture is from Databricks and is very nice because it does not directly say which product you need to use.</p> <p>So you have different sources and all different types of data go into the first zone of the storage layer. It starts with a bronze zone where data is only appended but you never do anything to the data. So you never update or delete rows, you just bring in new data and it might be useful for updating previous tables, however, I will not do that yet in this zone.</p> <p>Then the next zone, the silver zone, is where I will do computational stuff to update, clean, and filter the data.</p> <p>Then the third layer is where I really carry out combinations like joints and aggregations. In the zone, there is also an API layer so that the tables can be delivered to the applications that are used by the business user.</p> <p>This pretty much relates to Architecture 1 which is shown in the slides.</p>

Table 34: Respondent 1 - Evaluation of Existing Architectures

E.2. Respondent 2

	Observations
General	All architectures have different perspectives
Architecture 1	This is from a data flow perspective. This is like a breakdown of the processing logic.
Architecture 2	This is from a data format perspective. It is basically a breakdown of a data pipe. For the rest, this architecture is not really saying anything.
Architecture 3	This one tries to combine the perspectives of the previous two. However, I am missing the data zones in the data lake. And then different APIs access these layers because for certain use-cases you use data from the raw zone, but for others from the gold zone.

Table 35: Respondent 2 - Evaluation of Existing Architectures

E.3. Respondent 3

	Observations
General	-
Architecture 1	This one is okay, but it is not clear to me what kind of storage object you are typically using in a data lakehouse. Because the lakehouse is known for being the best of both worlds (data warehouse and data lake) so the storage object should be a data lake in my opinion.
Architecture 2	This one is a bit vague, to be honest. I don't really understand what the picture is trying to say. There is too little information, and it does not clearly show the concept of the lakehouse
Architecture 3	This one is I believe straight from Databricks, and I think the best. Although, it is missing the zones in the data lake.
Additional Architecture (Figure 30)	This architecture is directly from Databricks and is actually almost the same as Architecture 3 which is shown in the slides. Only this architecture (G3) clearly mentions 'Delta Lake' which is in essence the lakehouse developed by Databricks.
Additional Architecture (Figure 32)	This architecture is also from Databricks and is very important because it shows the design of the lakehouse.

Table 36: Respondent 3 - Evaluation of Existing Architectures

E.4. Respondent 4

	Observations
General	-
Architecture 1	<p>This architecture clearly shows the separation between storage and computing which is critical to point out for the lakehouse.</p> <p>This architecture is very nice and looks also quite complete in terms of visualizing data flows and process flows.</p>
Architecture 2	<p>It is a bit weird that this architecture is from top to bottom. It does not feel logical to me to visualise it that way.</p> <p>I like the way they have incorporated the data governance aspect.</p> <p>For the rest, I think it is a very minimalistic way of visualizing the architecture of a lakehouse. If I were to look at it really fast, just at first glance, it might come across as if I am looking at the architecture of the data lake. Not the lakehouse.</p>
Architecture 3	<p>This architecture is very vendor specific. It also mentions specific APIs that can be connected to the data whereas for the first architecture you can interpret that you can use different APIs.</p> <p>It is not entirely clear what the unique selling point of the lakehouse is. I see the governance layer on top of the data lake. But it does not tell me that this is a very big advantage of the lakehouse and is actually the thing that distinguishes the lakehouse from the data lake. So, this architecture needs some finessing.</p>

Table 37: Respondent 4 - Evaluation of Existing Architectures

E.5. Respondent 5

	Observations
General	-
Architecture 1	<p>It is remarkable that the architecture goes from left to right, I am used to an architecture going from bottom to top.</p> <p>The thing I am missing is the data lakehouse concept. It is now unclear to me what can be defined as THE lakehouse concept.</p>
Architecture 2	<p>This does not explain what the data lakehouse really is. I would not even call this an architecture. It just shows that you have something in the middle where all kinds of data are put in by people, and it is then used for machine learning.</p> <p>Then you have the curated layer with governance, that is fine. However, the question that remains is ‘what is the lakehouse concept?’</p>
Architecture 3	<p>This is a high overview of the architecture. It is fine, but again I am asking myself ‘what is the lakehouse’ concept? What makes this architecture a lakehouse and not a data lake or data warehouse?</p>
Additional Architecture (Figure 33)	<p>This is what I mean by visualizing the lakehouse concept. You clearly have this box around the layers that are supposed to form the lakehouse concept. The rest is very sophisticated on each layer explaining the possibilities and processes that take place here.</p>
Additional Architecture (Figure 34)	<p>This is a simplified version of the lakehouse architecture, however, again it clearly shows the lakehouse concept.</p>
Additional Architecture (Figure 31)	<p>This is the newest architecture from Databricks that was released in February.</p> <p>Basically, Delta Lake is just a way to store data in your lakehouse. It is a storage engine but I would even say it is more a concept that is used by Databricks. If you don’t know Databricks you won’t know the term delta lake.</p> <p>The unity catalog that they visualised in a layer here, is something very promising. However, it is not there yet. And Databricks knows this has to be in there and that they are currently lacking this. Hence, they have put it so clearly in the architecture to let the audience know that it will be part of the lakehouse platform. Though, they are still working on it.</p>

Table 38: Respondent 5 - Evaluation of Existing Architectures

Appendix F: Literature Review Protocol

This appendix describes, evaluates, and justifies the methodological choices that were made for the first literature review. It explains how the literature research was approached and which methodology was used to answer the research questions.

F.1. Literature Review Methodology

This section starts with an explanation of the chosen search strategy. This will be followed by an explanation of which database had been selected to obtain sources from. Next, the inclusion and exclusion criteria are formulated to finally guide the article selection process that was first based on title and abstract, and finally based on full text and references.

F.1.1. Search Strategy

To be able to address and answer the research questions a literature review is performed. To do this systematically, the approach that is used by Bukhsh et al., 2020 and Wienen et al., 2017 has been used as inspiration for this study. We first started with selecting the database(s) we wanted to use for this study. Next, a search query has been formulated that will help to perform focused literature research and find answers to the research questions that have been formulated. After extracting the articles that resulted from the search terms, the title and abstract were read. By defining inclusion and exclusion criteria, articles were selected based on their title and abstract which we thought would fit this study and we would want to read entirely afterwards.

F.1.2. Database Selection

We have chosen to utilize Scopus as the digital database to extract articles. The reason is that it covers a variety of articles from journals and conference proceedings, from almost any discipline, which enables to reach a diverse set of publications. This is beneficial given that we want to evaluate the use of a DMP in multiple sectors. Additionally, Scopus relies on a set of source selection criteria to produce a curated collection of documents. Whereas Google Scholar is very inclusive and gives each article “the chance to rise on its own merit” which possibly leads to some technical errors in the platform (Martín-Martín et al., n.d.). The final reason for this choice is that it is found to be the most comprehensive and user-friendly database, hence this database has been selected (Bukhsh et al., 2020).

After choosing a database, a search query was formulated to perform a focused database search for the most appropriate articles for this study. The following search query was used: “Data management platform”. This resulted in a total of 391 papers.

F.1.3. Inclusion and Exclusion Criteria

In order to select the set of articles we would use for this study, the following inclusion and exclusion criteria were formulated:

- IC1: The paper directly relates to the topic: the data management platform. This should be the main topic or at least one of the multiple research questions the paper tries to answer.
- IC2: The paper addresses the architecture of a data management platform.
- IC3: The paper is in English and available for download.
- EC1: The paper used a data management platform as a tool to answer specific use-case questions and solely focuses on the achieved results.
- EC: The paper is not related to any of the topics described in the sub-questions
- EC2: The paper is a duplicate.
- EC3: The paper is a conference paper and summarizes which papers were admitted to the conference.

F.1.4. Article Selection Based on Title and Abstract

These criteria were used while reading the title and abstracts of the paper that resulted from this search query. In this process, a 5-point Likert colour scale was used from red to dark green (Figure 32). The

reason was that a very broad perspective was taken on which made a difficult to classify an article into two groups, which were either ‘yes we select this paper’ and ‘no we will not select this paper’.

In the end, after going through the abstract of all the 391 articles that resulted from our search query, it resulted in the classification that is shown in Figure 32. We decided to immediately strike 208 papers based on our inclusion and exclusion criteria. For 36 papers we were not sure whether they could be of value however, we have chosen to keep them in mind whenever we could not find any relevant information that would answer a specific research question. Then, 63 papers were classified under “yellow” whenever the abstract would mention a “data management platform”, however, it was not entirely clear in which direction the paper would go. Then 59 papers were classified as “light green” whenever we thought the paper seemed to dive into the topic deep enough. Whereas 21 papers were classified under dark green when we expected the paper to perfectly describe the architecture of a data management platform. We decided to fully read the papers that were classified as dark green and light green. Whereas the papers classified as yellow, and orange were seen as backup papers for when the green and light green papers did not provide enough information to answer the research questions sufficiently.

F.1.5. Article Selection Based on Full Text and References

The list of 80 articles was the reading list for collecting data that was related to the research questions that were formulated in Section F1.2. After downloading and reading all articles, the final list of valuable articles for this study ended up containing 42 articles. The exclusion of articles was due to several reasons. 2 articles were chapters from a book and access was not granted to the full length of the chapter or the book.

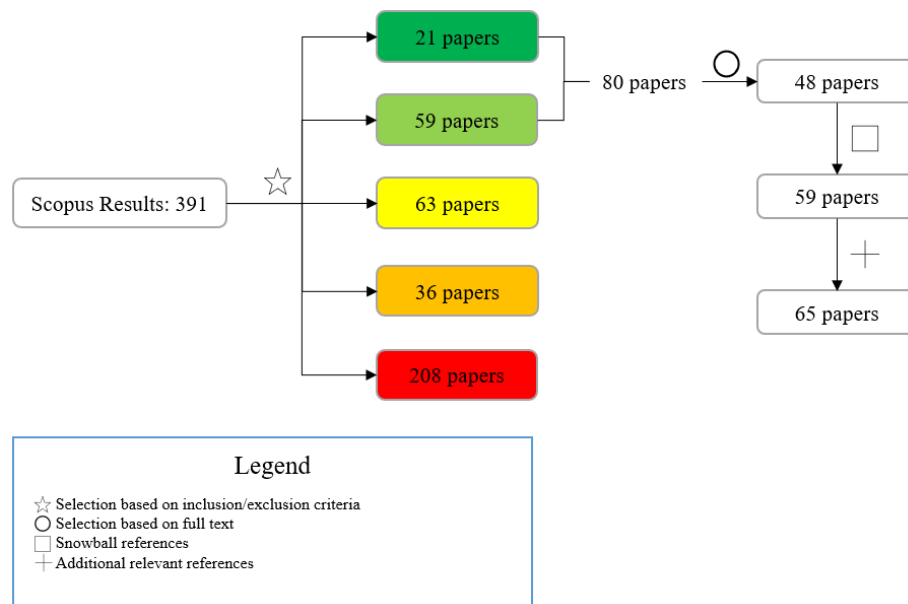


Figure 37: Article Selection Process

Additionally, 5 articles were not available for download hence in total, 7 articles were excluded due to no access. Another reason why one article was excluded, had to do with the fact that only the abstract was available in English, though the rest of the article was written in another language than English. Since no English version could be found, this paper was also excluded from the list. Furthermore, 24 papers were excluded because one of the exclusion criteria was applied. Hence, this selection process resulted in a list of 48 articles that were included in this literature research.

F.1.6. Data Extraction

In the articles, we found numerous examples of how the architecture of a data management platform is built, the results of implementations of such a platform in different sectors, and we have examined the various research streams concerning data management platform solutions. We gathered the publication

details of the articles that were used for this literature review. Furthermore, we made an overview in of which articles respond to one or more of our research questions. Lastly, we gathered all the future research propositions that could be used as inspiration for the research direction in which the thesis would go.

F.2. Literature Review

This chapter provides a literature review guided by the research questions that were formulated in Section F1.3. This literature review can be split into two parts. The first part is related to the first research question which focuses on data management platforms. Whereas the second part is related to the second research question and goes more into depth on the architecture of data storage solutions.

F.2.1. Core Parts of a Data Management Platform

Based on our findings in the literature, the data management platform consists of 5 core parts: a data ingestion module, a data storage module, a data processing module, a data visualization/sharing module, and data governance (Figure 38). The scope of this literature review focuses on the first three parts, given that the data governance and data visualization/sharing module is very important for enterprise-level. Whereas the first three parts are very important on an operational level which is the focus of this literature review. The following sections describe each core part in more detail, current practices that are adopted in a specific module, its relevance, benefits, disadvantages, and challenges. Given the focus of the thesis, the section describing the data storage module is more elaborated compared to the other modules.

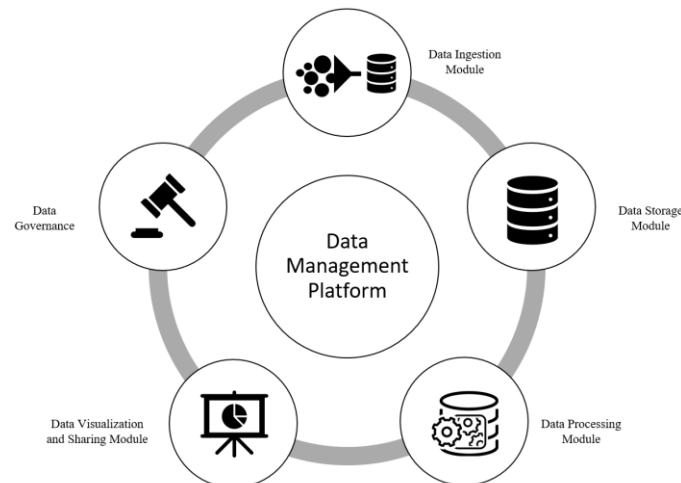


Figure 38: Core Parts of a Data Management Platform

F.2.1.1. Data Ingestion Module

Data ingestion captures the entire process of collecting data from multiple sources and loading it into the data management platform. Elmeleegy et al., (2013) refer to this process as “data integration” and state that next to ingesting data from different sources, this process also takes into account the necessary data cleaning processes and finally linking and merging data before it is stored. To go a little bit more in-depth in this process, Rooney et al., (2019) explain this process by zooming into three tasks: “1) the creation of a data asset, 2) the transfer of data across typically geographically separated systems as efficiently as possible, and 3) the ingestion of data into the data lake such that it is catalogued, governed and made available to the consumer.” This process is the push method and is depicted in Figure 39.

When a new data source is to be included in the data lake (task 1), the source owner copies their data asset to a “drop zone”. This drop zone is not part of the data lake since the source owner still has full control over the data asset and here, they can specify the type of data, optionally define access rules for privacy concerns, change their data asset by e.g. excluding certain columns, and finally they can trigger the transfer of data to the data lake with the ingestion REST API.

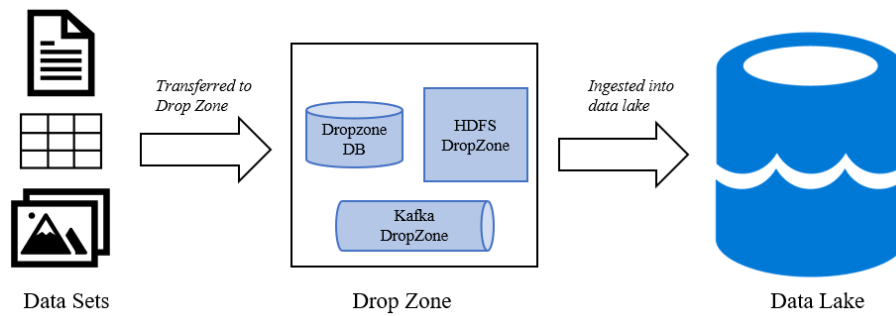


Figure 39: Process Flow with DropZone in Place

The second task is about the way data is being copied into the drop zone, before it enters the data lake. This is done by teams who either write scripts themselves or incorporate commercial extract-transform-load (ETL) and extract-load-transform (ELT) tools to coordinate the data transfer. Lastly, the third task is not primarily focused on copying data, but rather on governing the process of ingesting data into the lake and ensuring it is an auditable process. Information on who brought data into the lake, on which date at which time, from which data source, who may access the data, etc. must be recorded for compliance reasons. Moreover, when a new data asset is ingested, it needs to become part of the catalogue of data available in the data lake. This catalogue shows what and where data assets are available, and policies describing the acceptable usage of the data can be published here (Rooney et al., 2019). Besides the push method, there is also a pull method where data is directly being pulled out of the data sources through for instance an Azure Data Factory.

During the entire ingestion process, Elmeleegy et al., (2013) identified multiple challenges. One of the typical challenges is related to scalability. To ingest online data with high accuracy and in a timely way, multiple data centres are normally spread around the globe to be as close as possible to their client base. Hence, to scale up, more hardware needs to be put in place and perhaps the connectivity and capabilities of data centres that are already in place need to be improved. Also, the platform itself should have a scalable infrastructure in terms of available storage and the power to process increasing amounts of data. Another challenge is ensuring the high quality of the data. Online data is susceptible to fraud in different ways. Thus, the challenge here is to have fraud detection techniques to keep the quality of the data in the data management platform high. This is important for ensuring accurate and reliable analytical results. Lastly, within the marketing domain, the biggest challenge is to correlate different clicking and viewing actions of one user. Linking these events is essential for evaluating the effectiveness of ad campaigns and improving it however very challenging.

F.2.1.2. Data Storage Module

The second core part of a data management platform is data storage which, as the name implies, is a repository to store all the data. With the focus on a cloud-based data management platform, data storage in the cloud can simply be seen as the storage in cloud computing which is equipped with large capacity storage. The large capacity is enabled through the assembly of “different types of storage devices through the application software which is based on the functions of the cluster applications, grid techniques, distributed file systems, etc. (Liu & Dong, 2012).”

Two well-known examples of existing cloud data storage technologies from practice are the Google File System (GFS) and Hadoop Distributed File System (HDFS). GFS is hosted by Google and a GFS cluster consists of “a single master, multiple chunk servers, and is accessed by multiple clients. (Ghemawat et al., 2003)” The master maintains all file system metadata and controls system-wide activities like communicating with the chunk servers. The files that are being stored are divided into fixed-size chunks, and these are stored on the chunk server. When the client wants to read or write data to a specific file, it will ask the master which chunk server it should contact, and then the client will directly interact with the chunk server. (Ghemawat et al., 2003; Liu & Dong, 2012) Then the second example is the HDFS which is very similar to the GFS. The architecture of the HDFS namely also

consists of 3 components. The NameNode, DataNodes, and HDFS Client. The NameNode is the central server that maintains namespace hierarchy and file system metadata. The DataNodes are responsible for storing, updating, removing, and replicating data files. Just as in GFS, the data files in HDFS are also divided into blocks which are then stored in DataNodes. Lastly, the HDFS Client is the user who reads and writes data by communicating with the NameNode and also directly with a DataNode. The reason why HDFS is so similar to GFS is that HDFS was developed as an open-source project that was inspired by Google’s proprietary GFS and the MapReduce framework. The differences between GFS and HDFS lay in terms of implementation and file system operations (Liu & Dong, 2012; Shvachko et al., 2010).

When zooming into the storage solutions specifically for big data offered by Microsoft Azure (Table 29), there are 2 File Storage Solutions namely the Azure Blob Storage and Azure Data Lake Store. The first is the most flexible option for storing files and it supports the storage of basically any type of file from pictures to HTML files to logs and documents. One of the reasons why Azure Blob Storage is a good choice is its ability to support three types of concurrency strategies, optimistic concurrency, pessimistic concurrency, and last-writer-wins. Secondly, it supports account failover for geo-redundant storage accounts so that your data is copied to a second region as part of a disaster recovery plan. Thirdly, Azure Storage automatically encrypts your data with the use of server-side encryption (SSE) for the sake of protecting your data and meeting organizational security and compliance commitments. Lastly, it also supports role-based access control to assign who can view and edit data. The second solution is Azure Data Lake Store, which is “an enterprise-wide hyper-scale repository for big data analytics workloads”, and there is already an upgrade from Azure Data Lake Storage Gen1 to Gen2 which is a combination of Gen1 and Azure Blob Storage. There is no limit on file sizes or the amount of data that can be stored, it is compatible with all Apache Hadoop environments, it is cost-effective since you do not have to transform or move your data before you can analyse it, and it is specifically optimized for big data analytics (Microsoft Docs, 2021b, 2021c, 2022a).

File Storage	NoSQL Databases	Analytical Database
Azure Blob Storage Azure Data Lake Store	Azure Cosmos DB HBase on HDInsight	Azure Data Explorer

Table 29: Big Data Storage Solutions in Microsoft Azure

Then there are NoSQL database solutions that support the storage of any non-relational data, so the data is not stored in tabular relations. The first solution, Azure Cosmos DB, is “a fully managed NoSQL database for modern app development.” Any mobile, web, gaming, and IoT application that deals with massive amounts of data and reads and writes with near-real response times will benefit from this storage solution due to its high throughput, high availability, low latency, and tuneable consistency. The second solution, HBase on HDInsight, is a NoSQL database that is built on Hadoop and supports the storage of large amounts of unstructured and semi-structured data in a schemeless database where data is stored in rows of a table and grouped by column family. This solution can be used as a key-value store and for scenarios like managing message systems as Facebook does or managing tables that are extracted from webpages as WebTable does. Another scenario in which it is beneficial is for capturing sensor data that is incrementally collected that is randomly accessible in real-time (Apache HBase Project docs, 2022; Microsoft Azure, n.d.-b; Microsoft Docs, 2021a, 2021c, 2022b).

Lastly, Azure Data Explorer is an analytical database that supports collecting, storing, and analysing data coming from data streams that are emitted by modern software. It can handle data from any source and can be used e.g., for machine learning, monitoring, diagnostics, or reporting purposes. One of the typical use cases in which this solution is most valuable is for analysing data coming from IoT applications. Not only has Azure Data Explorer low-latency ingestion, but it also monitors remote IoT devices through telemetry data to ensure high performance. Another use case is the analysis of web-

based businesses that generate huge amounts of logging data. Lastly, it is valuable when this solution is embedded in SaaS applications to ingest data in real-time and perform any type of analysis. (Microsoft Azure, n.d.-a; Microsoft Docs, 2021c)

There are many benefits to storing data in the cloud instead of locally. First of all, companies have to pay for the storage that they use. This is cost-saving given that only operational costs are incurred and no capital expenses. Additionally, with the powerful processing capabilities and the huge amount of storage, organizations are even more encouraged to move to the cloud. Secondly, in cases of a disaster, cloud storage is safer than local storage given that a backup in the cloud is better protected than a backup on local data centres in cases of fire or water damage or any other type of disaster that brings down an entire data centre. Thirdly, in case of hardware failure, the cloud vendor is responsible for mitigating any possible hardware failure. Given their expertise and experience, they know how to provide hardware redundancy and automatic storage failover by distributing copies. Fourthly, cloud vendors offer a unified view of storage utilization through an easy export. Another benefit that is closely linked to the first, is that there is limitless storage capacity. Scaling up or down is easily done with some commands, which is a huge advantage compared to local storage. If the company desires to use less storage the costs will decrease, and the other way around. Lastly, access is granted from anywhere that the company allows. Even when you are not on your company's network, it is possible to access the data from any location since the data is consummated by network connectivity and client service. (Galloway, 2013; Obrutsky, 2016; Xue & Xin, 2016)

On the other hand, there are also some disadvantages when storing data in the cloud instead of locally. First of all, there are a lot of cloud vendors however they are not all equally experienced. Some are even still immature compared to others and hence, there is a higher risk of malfunctioning and incompatibilities when storing data in the cloud through these vendors. Secondly, one of the benefits is the fact that no capital expenses are incurred when storing in the cloud. However, one must take into account that cloud storage also has a price. Hence, calculating the cost-effectiveness of cloud storage versus hosting and maintaining data locally is something that should be considered. In some cases, cloud storage might not be financially beneficial. Thirdly, the company is losing some control to a certain extent when they choose to store data in the cloud since a third party is now given the responsibility to store and secure the data. When the cloud vendor is not able to fulfil its responsibilities due to a lack of knowledge and expertise, there is a possibility that the data is viewed by unauthorized people, and it is susceptible to data theft. Even though the cloud vendor can take preventive measures, the more sensitive the data the more risk a company is taking. Hence, governmental institutions could for instance demand extra security measures that the majority of companies would normally not require. It can therefore be that special knowledge is required on how to implement particular security measures. Fourthly, depending on the requirements of the company, it could be possible that the cloud vendor is not able to live up to the needed bandwidth. Bandwidth is "a measure of how much data can be moved from one spot to another in a given amount of time", hence the capabilities of the cloud vendor could be a limitation. Next to bandwidth, another technical limitation could be latency. Cloud-friendly data stores can respond well even when "data has to move, both in terms of physical distance covered and the number of network steps". However, when the responsiveness is low, in other words, low latency, it would be more beneficial to store data locally given the proximity. (Galloway, 2013; Marks & Lozano, 2010; Obrutsky, 2016) The last disadvantage we identified, depends on whether the client has specific requirements in terms of where their data is being stored. When a company wants its data to be stored in a specific region, some cloud vendors might not be able to provide that. An example would be the storage of European companies, they rather not have their data stored in Russia or China.

F.2.1.3. Data Processing Module

The last core part is the Data Processing module which, as the name implies, is all about processing the data by carrying out operations on it to produce meaningful information. Before data can be used for analytical purposes like finding patterns, making predictions, or generating reports with insights, the

data needs to be processed in such a way that it becomes a useable format. There are numerous data processing techniques. Krishnamurthi et al., (2020) propose to include the following data processing techniques when processing data, particularly IoT sensor data: data denoising, missing data imputation, data outlier detection, and data aggregation.

One of the problems with big data sets is the presence of noise. Noise can be defined as “the partial or complete alteration of the information gathered for a data item, caused by an exogenous factor not related to the distribution that generates the data. (García-Gil et al., 2017)” With classification problems, noise can be observed in terms of label noise or attribute noise. The first occurs when input is wrongly labelled and the second refers to erroneous attribute values. In general, the latter type of noise is very common and occurs very frequently. For instance, when referring back to IoT sensor data, the received signal may be modified from the original signal when it was transmitted. The modification is caused by some external factor that led to a change in value during the transmission. Hence, data denoising is essential for reliable results.

The second technique, missing data imputation, is essential when dealing with incomplete data. Learning algorithms and analytical techniques generally assume that the data is complete. This is a serious issue since the results that these techniques and algorithms produce will be inaccurate or unreliable. How Krishnamurthi et al., (2020) propose to solve this, is by estimating the missing value. For this, three tasks need to be implemented. First, the reason for missing data should be identified and the reason can either be 1) missing completely at random, 2) missing at random, or 3) not missing at random. Then, the pattern of missing data should be studied which helps with forming a missing value imputation model to approximate the value for the missing data.

The third technique deals with detecting outliers. Given that the IoT sensor network is widely distributed and heterogeneous, this setup is prone to failure and risks due to external factors resulting in the data being prone to outliers. Therefore, these outliers should be identified before performing any analysis. Numerous techniques can be performed to detect data outliers from which Krishnamurthi et al., (2020) chose to highlight the three popular methods: majority voting, classifiers, and principal component analysis.

The last data processing technique that is presented is the data aggregation method. This method collects and communicated information in a summary form which can then be used for statistical analysis. This method helps to decrease the number of transmissions between objects by summarizing data. Eventually, this will lead to an increase in the lifetime of the network and a decrease in energy consumption. (Krishnamurthi et al., 2020)

F.2.2. Generic Architecture of a Data Management Platform

This literature review examined 10 different studies that presented the architecture of their data management platform. The purpose of this examination was to find similarities and differences between different architectures and to discover a pattern in building a data management platform. As a result, we summarized our findings in one generic architecture. We will first briefly describe each architecture, after which we will dive into the commonalities and the differences. Finally, we will present our view on what a generic architecture of a data management platform should look like.

Zhang & Li (2019) propose a data management platform for a railway operator in China for operational and maintenance support. The architecture of the platform consists of 5 layers: 1) Data Source, 2) Data Collection, 3) Computation & Storage, 4) Data Analysis, and 5) Data Presentation (Figure 40A). Additionally, they built a platform management subsystem that provides operation and management functions of the platform, and it adheres to standards and security protocols through security management. When going from bottom to top, the first layer of the architecture is the data source layer. Here all types of data are captured both structured and unstructured. Then, the layer above is the data collection layer. This is where the data is collected and cleaned with processing techniques to improve

the quality of the data. When this is done, the data will enter the next layer which is the data storage layer. This layer supports the use of all kinds of databases for storing the data. Next, analyses can be performed in the data analysis layer, and finally, the results are displayed with any tool that is supported in the data presentation layer.

Ma et al., (2020) introduce a big data platform for the electric power industry. Enterprises in this industry deal with a large amount of structured and unstructured data of production, operation, and management. With this platform, problems such as insufficient data sharing among systems, data resource redundancy and dispersion, and maximizing data value are being tackled. The architecture consists of 5 horizontal layers: 1) Data Acquire, 2) Data Store, 3) Data Compute, 4) Data Analysis, and 5) Data Service (Figure 40B). Within the first layer data extraction, cleansing, transforming, validating, and loading are realized. Afterwards, the second layer realizes the correct way of storing the data by supporting the use of different types of data stores. Then, the data computing layer “provides a variety of computing engines such as SQL computing, batch computing, streaming computing, and parallel computing. Moreover, it realizes real-time and non-real-time data computing for the data of the system (Ma et al., 2020).” The layer on top of that is the data analysis layer. Here, different types of analytical methods are supported to analyse the data that is stored in the big data centre. The last horizontal layer is the data service layer which can be seen as the business function layer where end-users perform actions related to accessing and using the data. In addition to these horizontal layers, the authors have incorporated 3 vertical layers: 1) Data Asset Management, 2) Data Security Management, and 3) Platform Management. Hence, these practices are taken into account during each horizontal layer. The first, data asset management, concerns itself with the management of metadata and the quality of the data through standards. Secondly, the data security management layer is a basic function of the platform and realizes the security of system data through encryption, verification, desensitization, and authority control methods. The last vertical layer, platform management, is similar to a system admin given the responsibility for the upkeep and configuration of the system and realizing functions such as authority management and system audit.



Figure 40: DMP Architecture Examples A (left) and B (right)

Yang & Hu (2021) introduce a platform design that consists of 6 layers including a foundation layer, storage layer, supporting layer, service layer, access layer, and presentation layer (Figure 41C). The foundation layer is the foundation of the platform like the necessary network, servers, system software, cloud platform, and the configuration of middleware and the operating system. On top of this layer, the storage layer is built where different databases are used to store the data. Then, the support layer is there due to the adoption of microservice architecture. This ensures the independence of each service, hence when one service fails the others are not affected. Moreover, multiple servers provide the same service concurrently and as a result, high availability is provided. Functions like “service registration, service discovery, service configuration, service monitoring, and service routing are some of the functions that are supported in this layer. Next, the service layer is focused on providing basic services like user management, company management, equipment management, and permission and role management.

Then, the access layer acts as the name implies: it provides access to the end-user. Functions that are taken into account are permission verification, routing, load balancing, and traffic control. Finally, the presentation layer entails all functionalities provided in the user interface. From registering and logging in, to accessing data, using analytical services, and viewing reports.

Li et al., (2021) propose an architecture for a platform that stores and manages combat data efficiently to provide effective combat support for modern warfare. This study is the first in which the architecture consists of 3 layers: the physical layer, the data centre layer, and the application layer (Figure 41D). The physical layer, as the name implies, includes the necessary hardware, and this layer acts as the basis of the platform where storage units for data storage are provided. On top of this, the data centre layer is built which is the very core of the whole system. This layer includes multiple entities: the cloud platform, database entity, data service, and external data service. The cloud platform virtualizes physical resources like computing, storage, and network resource pools. This study utilizes the Hadoop distributed database technology and data cluster technology to deal with massive data with parallel storage functions. The database entity can include databases like relational, non-relational, or file databases to ensure the support of storage for all different data types. The data service entity ingests first-party data from all available sources and the external data service entity ingests data from external sources. Lastly, the application layer is on top and this layer provides business functions to the end-user. Here the end-user can choose to display data, perform data analyses and view data statistics. Next to these horizontal layers, the platform incorporates security measures on all levels to protect combat and military data. For the physical layer, isolation is the simplest and safest security measure however the downside of this is that the data has to be networked for transmission and data sharing. By adding security gateways and firewalls, illegal users are locked out and only authorized computers can access the platform. For the data storage layer, protection is provided by encrypting the data, and finally for the application security is ensured through controlling and managing user access. Hence, overall, security is ensured with different services like network security, data security, platform security, authority control, and access control.

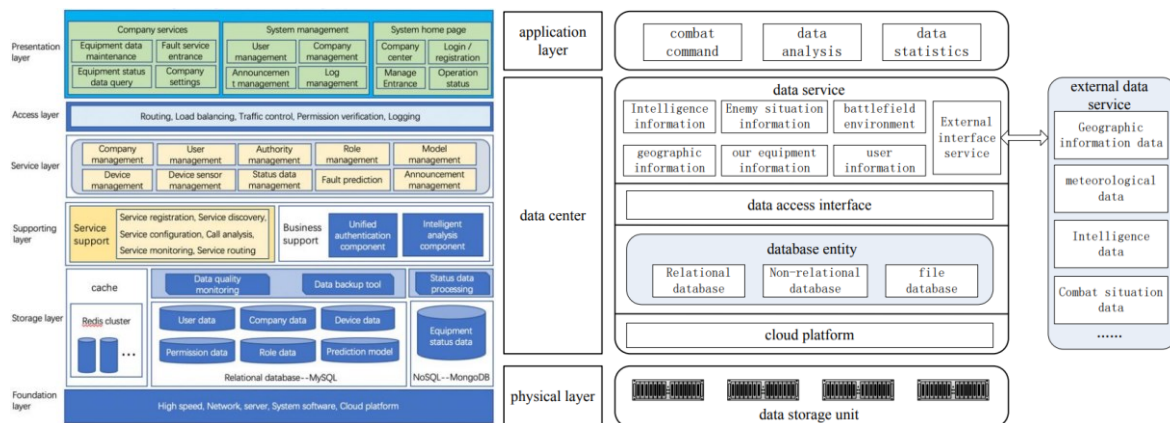


Figure 41: DMP Architecture Examples C (left) and D (right)

Next to the aforementioned study, Li et al., (2020) propose another platform with 3 layers. This platform was specifically designed for storing cryptographic data. The architecture consists of the data storage layer, the data processing layer, and the business application layer (Figure 42E). The first layer, as the name implies, stores the cryptographic data. Storing different formats is supported through different types of databases. The second layer provides main functions like parallel storage, various analysis models, and storing the results of these analyses. Lastly, the business application layer provides basic tools to view data and perform statistical analyses to live up to the business requirements. Next to these layers, the platform also incorporates security practices to ensure the protection of sensitive information. This is done by using SSL/HTTPS protocols to secure the data transmission, building a unified portal

to support single sign-on to control who accesses the platform, and lastly role management to control data access authority.

Deng et al., (2019) propose a data management platform for the integrated logistical support of unmanned aerial systems that are used in the military. Its architecture consists of 5 layers: hardware layer, software layer, data layer, function layer, and interface layer (Figure 42F). The first layer includes all necessary hardware like computers, workstations, and servers. The layer on top is the software layer where all data and functions are managed in a chosen software. Next, the data layer includes the database and data structure design. Here, the way data should be collected, stored, and managed is decided and this layer forms the basis for operational functions that can be performed with the platform. The fourth layer can be divided into 5 modules: platform management, function management, configuration management, mission management, and analysis & evaluation. Hence, this layer carries out all the operational tasks that are needed to create business value. Finally, the results of analyses and management reports can be accessed and viewed in the interface layer where end-users can directly interact with the platform.

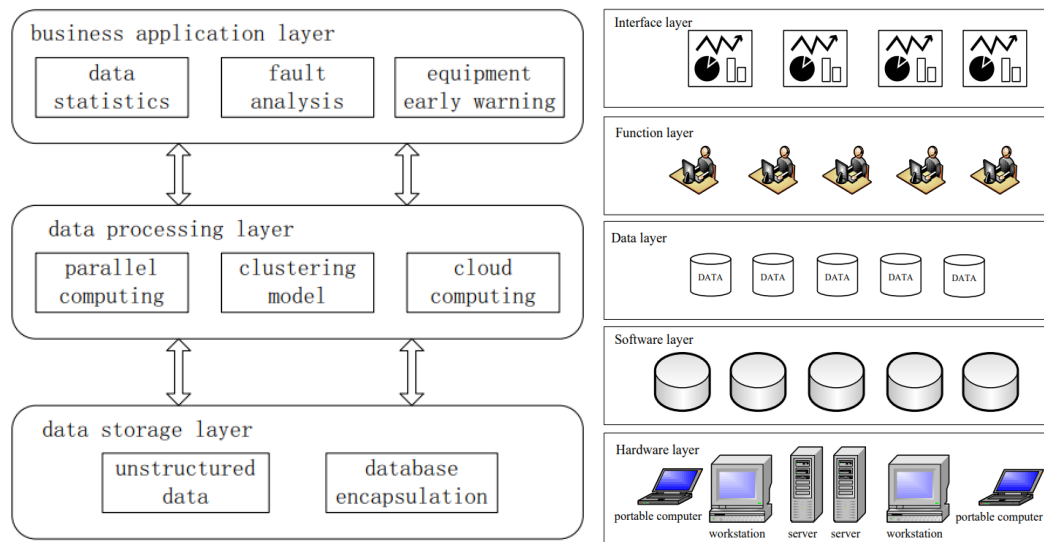


Figure 42: DMP Architecture Examples E (left) and F (right)

Choi & Shin (2018) developed a web-based healthcare data management platform to collect, store, and manage healthcare data coming from IoT devices. Its architecture is flexible to different types of data exchanges, and it consists of 3 layers: the data layer, the process layer, and the service layer (Figure 43G). The first layer enables acquiring health-related medical records and real-time lifelogging data. Through a medical record manager, the data will be translated and transformed into a standard format. This can then be used in the second layer. The process layer includes different modules: data management module, data aggregation module, semantic data integration module, and personalized risk analysis module. A data management module is built to manage health data repositories and device or user profiles. The data aggregation module collects the data that was processed into a standard format, and this data is being interpreted and analyses health conditions based on health ontologies in the semantic data integration module. Lastly, the personalized risk analysis module provides a function to perform risk analyses based on the health data that is being processed and visualizes the result dynamically to the patient and doctor in the service layer. The visualizations are adjusted and personalized according to whether a non-expert or an expert is trying to interpret the results. Finally, login, checking context conditions, and showing information by the role of user functions are implemented as data access control measures. This is to ensure sensitive data stays private and that only authorized users are allowed to access and view the data.

Fan et al., (2011) propose a data management platform to convert huge amounts of IoT data into lean data, and it is showcased in the industry of intelligent transportation by optimizing routing. The architecture of this platform consists of three layers: the data source layer, the data operation layer, and the application layer (Figure 43H). The first layer collects data from all the different available data sources and is then submitted to the upper layer. The upper layer would be the data operation layer which is responsible for storing and processing data. In total, this layer includes 7 modules: 1) data source management module, 2) data submission and storage module, 3) data acquisition module, 4) data selection and optimization module, 5) data integration module, 6) algorithm implementation module, and 7) extended function module. Altogether, the necessary data to make the expected function/service is picked and then processed by data selection, integration, and calculation to realize the expected function/service. This is then provided to the last layer which is the application layer where the results are shown in a user interface on the desired devices.

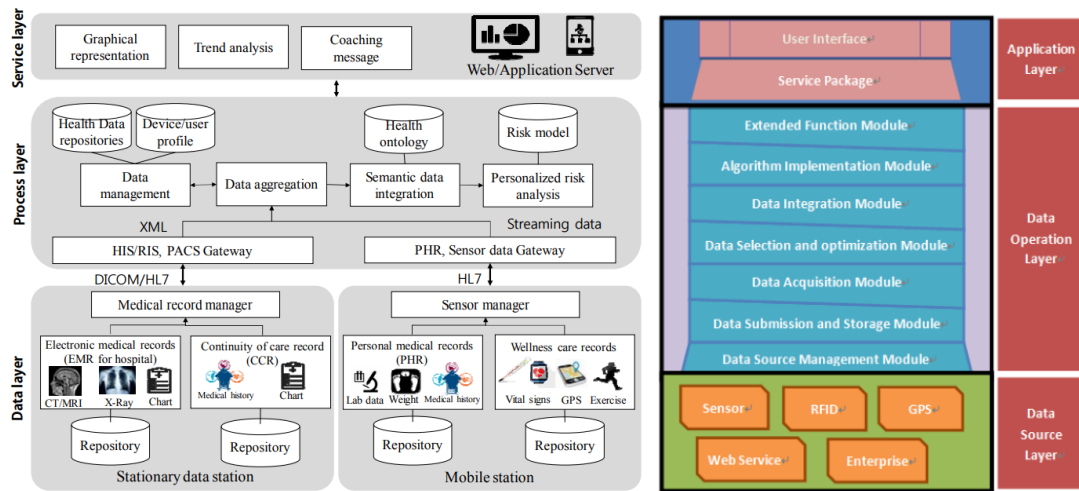


Figure 43: DMP Architecture Examples G (left) and H (right)

Delgado-Clavero et al., (2019) introduces a data management platform for monitoring Industrial IoT devices and visualizing features in a dashboard. The architecture consists of 4 horizontal layers and 1 vertical layer (Figure 44I). The first horizontal layer is the sensing layer, where information is obtained directly from the sensors placed on the work floor. This unprocessed data is interpreted and then grouped according to its purpose and data type. The relevant data is communicated to the next layer which is the network layer. This layer provides the connection between IoT devices and the service layer. Through a set of communication protocols, the IoT devices are integrated with the data management platform. When the data is transmitted from the network layer to the service layer, business logic processes are performed. On top of that, the information flow and data management tasks are also supported in this layer. Hence, this layer is the core of the platform given that most of the functionalities are supported here and it has a repository to store relevant data in an ordered way. Finally, there is an interface layer on top of the service layer where information is shown in a set of messages and a dashboard. Lastly, there is one vertical layer which is the security layer. Security should be taken into account across all layers, but it is mostly considered in the service layer where all the business logic is performed. Here the data is protected from outsiders through access control practices.

The last architecture that was examined is provided by Microsoft Azure. It consists of 6 layers of which the middle 4 layers form the core of the data management platform. The layers are as follows; 1) source layer, 2) ingest layer, 3) storage layer, 4) process layer, 5) serve layer, and 6) business layer (Figure 44J). The first layer indicates all the sources from which data is extracted. The provided examples include web servers, online traffic, OLTP, IoT, social networks, and mobile apps. Then the core is entered through the second layer which is the ingest layer. Here two elements are presented: an event hub and a data factory, which both trigger the process of ingesting data. The data factory is directly

connected with the next layer, which is the storage layer. In this layer, data is stored in a data lake. From here, data can be transmitted to the next layer which is the process layer. In this layer, data can be used for analytics through services like Databricks, Stream Analytics, and Cognitive Services. The results that are generated in these analytical tools can then be visualized in different applications that are supported by the serve layer. This fifth layer, the last layer of the core of the platform, allows data to be visualized in graphs, dashboards, or reports with for instance Power BI. Then finally, the sixth layer is the business layer which is not part of the core however, this is essential because this layer considers who the business users are of the data management platform, what the added value is of this platform, and with which devices this platform is compatible.

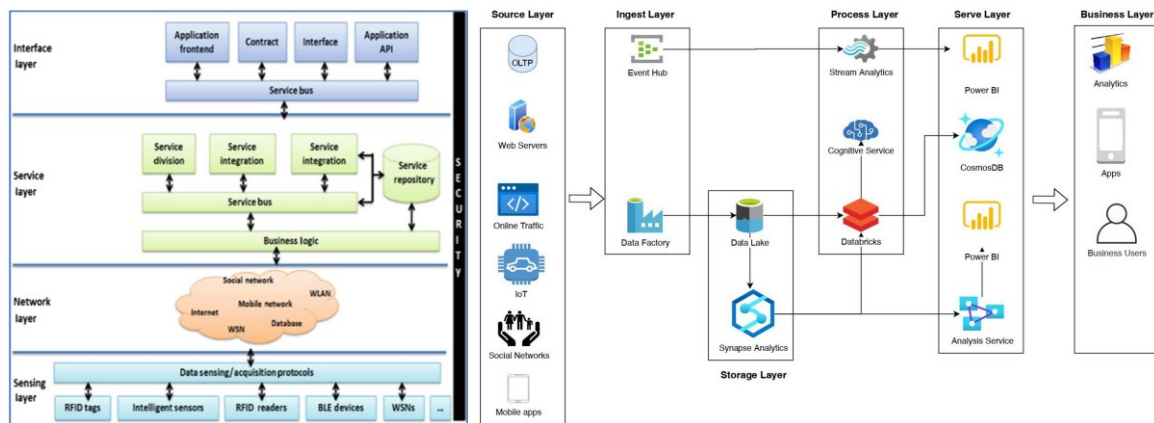


Figure 44: DMP Architecture Examples I (left) and J (right)

When taking the average of the number of layers in the architecture of the data management platforms that were examined, 4 horizontal layers and 1 vertical layer should be included. When comparing all the horizontal layers that were identified a lot of commonalities were found and a few differences. The differences were on one hand different terms for layers that had the same purpose, or some architectures included a layer that was not included in other studies. Two studies included a layer stating which data sources are included, three studies took into account which hardware is necessary and considered this to be a layer of the architecture, and finally, a clear distinguishment between architectures that consisted of 3 layers and architectures with more than 4 layers is the level of elaboration. For instance, two processes that take place in the second level of the 3-tier architecture, are split up and each has its layer in a 5-tier architecture. When looking at studies that implemented at least one vertical layer, the commonality was that this was always a security layer. Only one study added two other vertical layers: the data asset management layer and platform management layer. Given that only one study out of ten included these additional vertical layers, these were left out for the generic architecture of a data management platform and a security layer was included.

Concludingly, based on the observations and commonalities across all 10 different architectures, a generic architecture is presented in Figure 45 which captures the most essential layer while covering all aspects that were described in those architectures with more than four layers. The first horizontal layer is the Data Collection and Storage layer. This layer is the foundation of the platform given that here the ingestion of huge amounts of data is brought into the platform and stored in its raw form. Then the data will be processed in the second layer. In the Data Processing layer, the data will be prepared before it is pushed to the next layer. The third layer is the Data Analysis layer where analytical methods are applied according to the business requirements. Finally, the results of these analyses are presented in the top layer which is the Data Visualization layer. Next to these four horizontal layers, one vertical layer is implemented that considers all security measures that need to be implemented to protect the platform at all levels and make sure that data remains confidential, integer, and authentic.

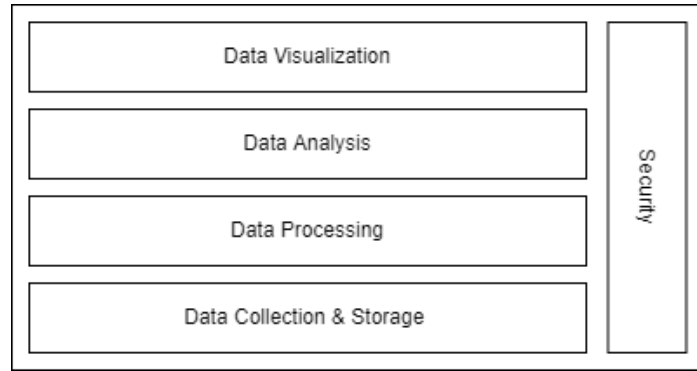


Figure 45: Generic Architecture of a Data Management Platform

F.2.3. Existing Data Management Platform Solutions

This section delves into the existing solutions in literature and practice. Due to a large number of existing solutions, they have been categorized according to the following domains: smart cities, healthcare, transportation, internet of things, and marketing (Table 30). Evidently, there exist multiple solutions in different domains as well. However, these were identified in the articles that were selected during the search strategy that was executed.

Domain	No. of Solutions	Solutions	References
Smart Cities	8	MEILI, Mobility Big Data Analytics, CloudTP, IRMA, SMIP, Future Mobility Sensing System, DataFog	(Prelicean et al., 2018), (Motta et al., 2015), (Ruan et al., n.d.), (Fukuda et al., 2018), (You et al., 2020), (Gupta et al., 2018)
Healthcare	7	Pdmdims, Data Platform for Diagnosing Epilepsy, IBM Watson Medical Platform, People's Hospital of Peking University DMP, healthcare data integration and management platform, RDMP, Qure Data Management platform	Peili et al., (2018), (J. Wang et al., 2017), Choi & Shin (2018), Nind et al., (2018), (Jäger et al., n.d.),
Transportation	3	Shuohuang Big Data Management Platform, Taxi Trajectory Data management Platform, Lean Data Management Platform	(Zhang & Li, 2019), (Krataithong et al. 2021), Fan et al., (2011),
Internet of Things	14	IoT Data Management Platform, IBM Mote Runner, MicroStrains SensorCloud Platform, Ostia Portus Platform, TempoDB Platform, FreshTemp Temperature Monitor, SensaTrack Monitor, Bluwired S-Cloud Platform, Xively Platform, Nimbits Platform, ThinkSpeak Platform Microsoft Azure Intelligent System Service, Paho Platform, INDIGO, OPTIFY	(Hasegaw & Yamamoto 2020), (Emeakaroha et al., 2016), (FreshTemp, 2022), (MicroStrain, 2022), (Ostia, 2022), (TempoDB, 2022), (Berbís et al., 2019), (Delgado-Clavero et al., 2019)
Marketing	3	Turn Data Management Platform, Microsoft DMP, Oracle DMP	(Elmeleegy et al., 2013), (Microsoft Corporation, 2022), (Oracle, 2022)

Table 30: Existing Solutions Identified in Literature per Domain

F.2.3.1. Smart Cities

In the field of smart cities, several solutions have been developed to deal with the data that comes from sensors and IT systems that are placed in cities. For instance, MEILI (the name is derived from the Norse god of travel), is a system that allows for “the collection and analysis of geo-referenced data to support the study of mobility patterns by integrating tracking technologies with Geographic Information System (GIS) and interactive web-based validation (Prelicean et al., 2018).” This platform is used to summarize how people travel to fulfil their daily schedule in travel diaries, by integrating multiple sources and making the travel diaries available for any type of research that uses trajectory data.

Two other solutions, Mobility Big Data Analytics and CloudTP show how beneficial homogeneous multi-source data fusion is for identifying popular regions and generating accurate travel trajectories. However, the downside of these two systems is that it is not suitable for studies that aim to analyse user behaviour more in-depth. (Motta et al., 2015; Ruan et al., n.d.) Two different solutions try to complement these limitations. Both the Integrated Real-time Mobility Assistant (IRMA) and Smart Mobility Information Infrastructure Platform (SMIIP) try to allow the collection and utilization of different data sources. This allows for more in-depth analysis; however, both these solutions are still not capable of tackling issues in processing and managing different data types in a unified and scalable manner (Fukuda et al., 2018; Motta et al., 2015). Based on these limitations, You et al., (2020) proposed a generic solution called the Future Mobility Sensing system, which can be customized for various stakeholders and different purposes. This is to ensure that heterogeneous multi-source data can now be processed and managed efficiently and accurately and also visualized in a suitable format depending on the type of data.

Lastly, one study found that traditional cloud-based time-series databases are not able to deal with the enormous amount of data coming from geo-distributed smart services. According to (Gupta et al., 2018) the existing solutions cannot meet the low-latency requirement for generating, ingesting, processing, and storing this data in the system. Consequently, Gupta et al. developed DataFog which is a “geo-distributed data management platform at the edge of the network to cater to the needs of smart services for the IoT age (Gupta et al., 2018).” This is the first system that enables data management at the network edge, according to the best of the authors’ knowledge, designed for IoT services for smart cities.

F.2.3.2. Healthcare

Another field in which multiple data management platform solutions are developed is healthcare. For instance, Peili et al., (2018) developed a data management system called “Patient Data & Deep Learning Model Data Integrated Management System (Pdmdims)”, in which a deep learning model was integrated to support and contribute to coronary heart disease (CHD) early warning research. With the integration of deep learning methods, the focus is not solely on establishing and optimizing early warning models, but also on the data lifecycle management through algorithms that conduct the lineage management of the deep learning model.

Another research proposes a data platform that specifically aids with the prognosis of epilepsy. By extracting quantitative data, a basis is provided for data analysis and data mining techniques which form the foundation for making predictions and intelligent diagnoses of epilepsy. While developing this data platform for diagnosing epilepsy, a variety of domestic and foreign data platforms were examined such as the IBM Watson Medical Platform and The People’s Hospital of Peking University Data Management Platform (J. Wang et al., 2017). The challenges and weaknesses faced here were used as critical focus points to improve the platform developed by Wang et al. This includes challenges like dealing with inconsistent medical data, continual change in medical data, and the transfer of data between different systems.

Furthermore, research was done into developing a data management platform for healthcare IoT devices. Given the rise of new technologies, there are numerous ways to measure health-related data with wearable devices. With this trend, Choi & Shin (2018) presented a web-based healthcare data

integration and management platform. By incorporating a flexible architecture, different types of data exchanges are allowed from electronic medical records that are produced in hospitals, to personal health records from wearables such as a Smart Watch or Fitbit that monitors activity, sleep quality, dietary habits, and more daily. Hence, this research concludes that the proposed platform is both suitable for hospitals for remote medical services and healthcare home services.

Lastly, two studies focused on setting up a data platform specifically for research that utilizes healthcare data. There was a significant need for an easy-to-use platform that allows the collection and processing of data in different forms and the exchange of this data with different parties. Moreover, this platform should incorporate strict governance to comply with privacy rules concerning personal medical data. The Research Data Management Platform (RDMP) proposed by Nind et al., (2018) handles privacy by having extraction rules in place that tells which columns are extractable, which columns require special governance approval, and which contain patient identifiers and are therefore not extractable. Next to that, they have proved that with the use of the RDMP the process of producing, extracting, requesting, and receiving data has significantly been improved through a reduction in time needed to execute one of these processes. The second solution we examined was presented by Jäger et al. (n.d.) who designed a flexible database platform for biomedical research called Qure Data Management platform. Just as Nind et al., Jäger et al., also concluded that there was a great need for systems "...that allow to design and launch data collection initiatives quickly, securely and reliably." As a result, they designed a database platform with a flexible architecture, that "...contains tools for dynamic designing of data models and electronic questionnaires, multiple user interfaces for data entry and a universal query engine for filtered output." Even though this research was focused on data produced and used in biomedical research, Jäger et al. argue that this platform could also be used for other healthcare research domains due to its flexible architecture.

F.2.3.3. Transportation

Also, for the transportation domain, multiple solutions have been presented by several studies for different reasons. One of the solutions is a Big Data Management Platform for a railway operator in China. The purpose of this platform is to gain maximum value from the available data and to provide intelligent analyses that support decision-making processes to guarantee safe transportation and on-time maintenance of the railways and locomotives. This is done through the implementation of 7 core functions "1) locomotive fault prediction and health management, 2) personnel health management, 3) transport environment monitoring, 4) intelligent locomotive maintenance, 5) locomotive electronic resume, 6) comprehensive analysis and release, and 7) comprehensive display for big data (Zhang & Li, 2019)." With these functions, this platform will cover 4 important elements: "human, loco, environment, and management" in one platform. Data that is then gathered such as massive traffic data, operational data, maintenance data, and production data will be exploited to the fullest through analyses and data mining, to gain maximum value to provide insights for decision-making.

A second study focused on a different type of transportation, namely the taxi industry. In 2021, Krataithong et al. proposed a platform that can handle a large volume of taxi trajectories and increase its value with more meaningful semantics. Adding meaningful semantics to the trajectory data of taxi GPS data is done with a knowledge base as one of the modules of the platform. With this knowledge base the following functionalities are enabled: 1) defining a semantic model of tourist movement behaviour, 2) formulate rules for the classification of tourist trajectories, and 3) semantic search and data integration. This was done to enhance the analysis of tourist behaviour so that meaningful insights can be gained into tourist movements and activities. Consequently, these insights can be used by governments and tourism businesses to improve their operations and they can also support making predictions of tourism trends in the future.

A third study that also focuses on bringing more semantic meaning to the data was done by Fan et al., (2011). The proposed data management platform incorporates lean principles concerning their data processing tasks, to eliminate waste from the customer's perspective. The basic lean principles are 1) add nothing but value, 2) centre on the people who add value, 3) flow value from demand, and 4)

optimize across organizations (Poppendieck, 2011). These principles are applied to the data pipeline to maximize the value of the data. The essence of applying lean principles to the data pipeline is to eliminate redundant data during different stages. The combination of applying lean principles and ontologies, a unified and efficient way “...for data flow description and semantic data storage” is achieved resulting in “...a unified understanding among different layers and modules (Fan et al., 2011).” The effectiveness of this platform is showcased with data concerning transportation and the purpose of the platform is to be able to optimize routes in real-time with more accuracy by “analysing and calculating all those relevant parameters, instead of considering the distance only (Fan et al., 2011).”

F.2.3.4. Internet of Things

Over the years, the Internet of Things has been a rapidly evolving field of technology in which various types of systems were developed to monitor, measure, and support human activities in numerous fields. For instance, by tracking location information through GPS devices of transport vehicles continuously, cargo can be tracked in real-time which helps to optimize routes and communication between all stakeholders. (He et al., 2009) The IoT systems generate real-world data hence there is a need for technologies that can deal with the data streams that are constantly incoming. However, there is even a greater need of having the assurance that the data is authentic and integer. Given that data from IoT systems assist in daily operations, it is of high importance that the technologies used for processing the data have sufficient functions for security measures in place. In 2020, Hasegaw & Yamamoto examined the prospects concerning building a data management platform using blockchain technology. They evaluated five studies that presented a data management system utilizing blockchain technology for multiple IoT systems. Based on the weaknesses of these platforms, Hasegaw & Yamamoto proposed a new IoT Data Management Platform that cannot only guarantee the integrity of the data but also the authenticity by preventing any attempt of tampering with the data. Their proposed platform has been used for a physical distribution system where the “verification of location information of transport vehicle, time is taken for registering the data on the blockchain and verifying the authenticity is evaluated (Hasegaw & Yamamoto, 2020).

Emeakaroha et al., (2016) also evaluated different existing solutions with a focus on cloud-based sensor data management platforms. They identified and compared the following existing solutions: IBM Mote Runner, MicroStrains SensorCloud Platform, Ostia Portus Platform, TempoDB Platform, FreshTemp Temperature Monitor, SensaTrack Monitor, Bluwired S-Cloud Platform, Xively Platform, Nimbits Platform, ThinkSpeak Platform Microsoft Azure Intelligent System Service, and Paho Platform. These platforms are each designed for different purposes. For instance, MicroStrains SensorCloud is used for structural health monitoring, whereas the Ostia Portus Platform is in use for mediating between multiple vendor sensors, networks, and platforms. Then there are two commercial tools, TempoDB Platform and FreshTemp Temperature monitor. The first is a tool for storing and analysing time series data from smart meters, sensors, and automotive telematics. The second solution collects data from temperature sensors during the production, transportation, and storage of perishable goods. Thus, this tool provides solutions for transportation, food services, industrial usage, and health care solely based on temperature sensors. Furthermore, multiple solutions are suitable for real-time monitoring, the analysis of sensor data and visualize it, monitoring machine-to-machine usage scenarios, and more (Emeakaroha et al., 2016; FreshTemp, 2022; MicroStrain, 2022; Ostia, 2022; TempoDB, 2022). After the examination of all these platforms, Emeakaroha et al. decided to develop a conceptual architecture for a generic cloud-based sensor monitoring and data gathering platform. Since all the existing solutions are used for specific use cases, Emeakaroha et al., found a gap in which no generic solutions exist. By addressing core challenges such as “communication bottle-neck, data interchange formats, security, and operability” an attempt was made to propose an open solution that can be used and adjusted for different use cases.

Another emerging stream is the application of Industry Internet of Things (IIoT), other terms that are used are Industry 4.0 and Cloud Manufacturing. Here, a lot of data is generated in manufacturing processes, enterprise management, and product transactions. Since this data is generated in different processes, the data is also maintained by different teams and departments. The classic result of this is

data isolation and consequently, data cannot be used for relational analyses. In order to tackle this problem in this domain, a cloud-based data management platform was created to “...maintain the data from isolated domains and distributed departments (Ren et al., 2020).” Two existing applications were combined into one platform: a Master Data Management system and a Graph Database. The first provides functionalities to manage related data in different systems, and the second provides an effective approach for storing and processing big data with complex relations.

Next to Ren et al., Berbís et al., (2019) also observed that the main challenge of Industry 4.0 is to prevent the existence of data silos. As a result, they present INDIGO which is an Industrial Internet of Things data management platform. This platform prevents the formation of data silos by building a semantics-based software platform to enable semantics and ontology-guided storage. Another study also presents an IIoT Data management platform that is built on semantics to monitor industrial IoT devices. This platform is called OPTIFY and was presented by Delgado-Clavero et al., (2019). Besides avoiding data to be isolated, this platform also tackles the challenge of measuring and optimizing the performance of industrial IoT devices that help to improve workflow control.

F.2.3.5. Marketing

For this domain, one study was found and official documentation of data management platform vendors. First, the study that was found presented the Turn Data Management Platform (Elmeleegy et al., 2013). Just as for the other domains, emerging technologies and devices have a great impact on the marketing field. It is a major challenge for marketers to formulate a fine-grained marketing strategy due to the existence of many different platforms and the use of multiple different devices. Because data is generated in different platforms and different forms, it is highly desired to have a data management platform for this field. The platform should be able to integrate data from different social media platforms with heterogeneous data. Moreover, it is desired that data on the same users across different platforms can be linked to gain a deeper understanding of the effectiveness of campaign activities. Another functionality that the platform should support is analytical capabilities. And lastly, it should not only be possible to ingest data in real-time but also generate and send data in real-time to make insights actionable. Hence, this platform is not only used for analysing customer behaviour across different channels, but it also generates actionable outputs in real-time to optimize the downstream media and also enhance the customer experience (Elmeleegy et al., 2013).

Next to this study, the documentation of a data management platform vendor was evaluated. Microsoft is one of the vendors of a data management platform and they have written documentation on how such a platform can be of value for the marketing domain. The four main functionalities that this platform provides are ad targeting, creating personalized experiences, communicating with other platforms, and data monetization. The first two functionalities are specifically focused on the audience that is targeted with campaign ads. Not only does this platform collect data on how the customer interacts with the ads, but it also creates customer profiles and specific content and ads that will be shown to the segmented audiences. The last two functionalities focus on the ability to communicate with other platforms to buy ad placements, second-, and third-party data, and to sell data that is directly collected with this platform as second-party data to others. Despite these valuable functionalities, there are also some limitations to this data management platform. One is that due to the level of analysis, the platform requires longer processing times for ingesting and analysing new data. Another limitation is the fact that the data is often relying on cookies which means that the data retention is typically only 90 days. Thus, there is no unlimited access to older data. Thirdly, even though you can view the insights that are generated by the data management platform, it is not always possible to view the data from which these insights were generated. This can be seen as a limitation given that there is a lack of transparency. Lastly, with a data management platform, it is not possible to target individual users so personalized marketing is not possible. The goal of this platform is to have a segmented audience based on attributes and not individual entities. If this is not something that is desired, it does not have to be a limitation. Nevertheless, a data management platform is very valuable for collecting, organizing, and activating data from different sources and putting it into actionable output. (Microsoft Corporation, 2022)

The final example of an existing solution is a data management platform offered by Oracle. This vendor also documented how and why this platform is valuable for digital marketing for the company to understand its customers better. They argue that being data-driven is not enough given all the emerging technologies and digitalization of our society. It is highly important to be quality data-driven to ensure that you can deliver “targeted, personalized messages that support and move customers along the purchasing process in a natural way. (Oracle, 2022)” The data management platform will help with excluding non-essential data to “safely analyse and refine your datasets so that only the most accurate data is used in your marketing efforts. (Oracle, 2022)” This way, marketers will reach both customers and look-alike profiles with targeted ad campaigns which not only leads to higher efficiency but can lead to more customer purchases.

F.2.4. Challenges of Data Management Platforms

Several studies examined which challenges they encountered while implementing a data management platform. In Table 31, an overview is provided of the challenges that were identified in three different studies.

You et al., (2020)	J. Wang et al., (2017)	(J. Wang et al., 2013)
<ul style="list-style-type: none"> • Incompleteness of data • Management of heterogeneous data • Data fusion process • Knowledge generation • Visualizing information 	<ul style="list-style-type: none"> • Different data structures • Continuous changes in data structures • Transferring data between 2 systems 	<ul style="list-style-type: none"> • Central storage for heterogeneous data • Management of frequently accessed data

Table 31: Overview of Challenges Identified with Data Management Platforms

You et al., (2020) identified five challenges in the collection and management of travel data for urban transport planning. The first challenge is how to deal with the incompleteness of data during the data collection phase. When a single source needs to be complemented by another source to retrieve meaningful information from this source, the collection function requires a component that is easy to deploy and allows for high scalability and flexibility. Moreover, it is of high importance to maintain the quality and integrity of the data that is collected. Another challenge was found in the management of data. Managing multiple sources that have heterogeneous data requires a flexible data management approach to store and link data semantically in a uniform way. This leads to another challenge which is how the heterogeneous multi-source data goes through a data fusion process, in which it is efficiently cleansed, fused, and analysed to extract useful and meaningful information from the data. The fourth challenge is the procedure of knowledge generation through data mining techniques. The knowledge needs to be mined from a multi-dimensional fused data set; hence an efficient data mining mechanism needs to be in place to extract meaningful information that does not have a too complex configuration. The last challenge is related to visualizing the information intuitively and concisely while keeping the usability and aesthetics in balance.

Three challenges were identified by J. Wang et al., (2017) during the development of a medical data management platform, one of which was the structural weakness of medical data. Medical data can take so many forms, from text description to image data, to audio data, to numerical data. In order to deal with this challenge, Wang et al. propose to manage structural and unstructured data separately to ensure that structural data can easily be used for data analysis. Another challenge specific to medical data is that there is continued change in data in the frontier medical field. Thus, the data platform should be customizable so that it can adapt accordingly to the configuration of the data. The last challenge is transferring data between two systems. The data systems that are in use are to some extent mature, but the data seems not to be compatible between different systems. Consequently, a method is required that allows for data migration between different databases.

Additionally, in 2013 J. Wang et al., poses two main challenges for multi-source data storage and management. The first is a centralized storage of data organization. The challenge here is when the same data is generated by different sources and they are all slightly differently formatted, to store this efficiently. Currently, there are many problems with data storage since storage is wasted on these kinds of duplicates. The solution that J. Wang et al. proposes is to store similar data in the same shared table and make this transparent to all stakeholders. However, the realization of this is very time-costly and stakeholders might need extra explanations on why they are seeing the same data in the same table but all just a little different. The second challenge is related to the management of frequently accessed data. The ability to access real-time data frequently will eventually lead to poor performance of the data management platform because it will be overloaded with many activities. This could be solved to some extent by incorporating more data storage space, however, the next challenge that arises is how to achieve and manage the distribution of data storage and maintain a transparent and easy-and-fast-to-access data management platform (J. Wang et al., 2013).