First Experiences on the Application of Lakehouses in Industrial Practice

Jan Schneider[1], Arnold Lutsch[2], Christoph Gröger[2], Holger Schwarz[1] and Bernhard Mitschang[1]

[1]Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
[2]Robert Bosch GmbH, Borsigstraße 4, 70469 Stuttgart, Germany

Abstract

In recent years, so-called lakehouses have emerged as a new type of data platform that intends to combine characteristics of data warehouses and data lakes. Although companies started to employ the associated concepts and technologies as part of their analytics architectures, little is known about their practical medium- and long-term experiences as well as proven architectural decisions. Additionally, there is only limited knowledge about how lakehouses can be utilized effectively in an industrial context. Hence, it remains unclear under which circumstances lakehouses represent a viable alternative to conventional data platforms. To address this gap, we conducted a case study on a real-world industrial case, in which manufacturing data needs to be managed and analytically exploited. Within the scope of this case, a dedicated analytics department has been testing and leveraging a lakehouse approach for several months in a productive environment with high data volumes and various types of analytical workloads. The paper at hand presents the results of our within-case analyses and focuses on the industrial setting of the case as well as the architecture of the utilized lakehouse. This way, it provides preliminary insights on the application of lakehouses in industrial practice and refers to useful architectural decisions.

Keywords

Data Lakehouse, Data Platform, Platform Architecture, Data Analytics, Case Study, Industry Experience

1. Introduction

With the growing range of capabilities for data acquisition and the current advances in the field of analytics [1], data is becoming an increasingly important asset for enterprises of all business fields. For example, in the industrial sector, data from the shop floor can be exploited with data mining and machine learning techniques for efficiently orchestrating manufacturing processes, predicting the quality of products and scheduling the maintenance of machines [2]. Similarly, this also applies to organizations from other business fields, such as healthcare [3] and agriculture [4].

Data platforms constitute the technical foundation for all kinds of analytics applications within enterprises, as they are capable of storing and managing huge amounts of data for analytical purposes and thus support data collection, processing and analysis [5]. Traditional *data warehouses* [6] and the more modern *data lakes* [7] represent the two most popular types of data platforms. Originally, they were designed for different kinds of analytics applications and hence tend to show rather opposing characteristics [8]: While conventional data warehouses share many similarities with relational databases and are primarily utilized for reporting and Online Analytical Processing (OLAP) workloads, their proprietary data formats and rigid data models impede explorative data analyses and are typically less suited for many types of advanced analytics [8, 9, 10], such as data mining.

Data lakes attempt to close this gap by enabling the storage of raw data in their original formats, without requiring the data to be transformed into a pre-defined schema before it can be loaded onto the data platform. For this purpose, data lakes typically employ highly scalable and cost-effective storage systems like distributed file systems or object storages. However, in comparison to data warehouses, this increased flexibility comes at the cost of less comfortable data management and analysis capabilities. In summary, it can be concluded that data warehouses typically represent a reasonable choice for use cases in which the analysis questions are already known in advance, while exploratory workloads with unknown analysis questions are more appropriately supported by data lakes. Consequently, enterprises often need to operate both types of data platforms in parallel and either exchange or replicate the data between them in order to be able to serve all kinds of analytical workloads. This commonly results in complex architectures, high operational costs and slow analytical processes [9].

In order to address these issues, efforts have recently been made to develop so-called *lakehouses*, which represent another variant of data platform that intends to combine desirable characteristics of data warehouses and data lakes. This way, lakehouses are supposed to serve all kinds of analytical workloads by a single data platform. In literature, multiple different definitions for lakehouses exist [9, 11, 12] and apparently, there is currently no final consensus on how lakehouses can be charac-

35th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), May 22-24, 2024, Herdecke, Germany.
jan.schneider@ipvs.uni-stuttgart.de (J. Schneider)

terized. Nevertheless, most authors seem to agree that data lakes, which are based on highly scalable storage systems and have been enhanced for additional data warehousing capabilities with the help of certain specialized frameworks, can generally be referred to as lakehouses [13, 14, 15, 16, 17].

The most popular representatives of such frameworks are the open-source projects Delta Lake[1], Apache Hudi[2] and Apache Iceberg[3]. All of these frameworks provide libraries for various popular data query and processing engines, such as Apache Spark[4] and Apache Flink[5], and enhance them for additional data warehousing capabilities when working on data that resides on distributed file systems or object storages. This includes features for abstracting stored data as tables with relational characteristics, ensuring ACID properties and enabling efficient batch and stream processing [13, 14, 12]. Consequently, data can be flexibly stored in open formats and in a directly accessible manner, but still be conveniently processed and queried. This allows such lakehouses to cover large portions of the typical analytical workloads of data warehouses and data lakes.

In literature, several works can be found that propose and discuss different lakehouse implementations based on these frameworks, such as for the domains of healthcare [18, 19, 20], biomedical research [21], network management [22], IT security [23] and geospatial analytics [24]. However, these descriptions often lack details regarding the chosen architectural decisions and the extent to which these decisions have proven over time, especially in terms of aspects such as data organization, data modelling and data flow. Furthermore, to the best of our knowledge, there are yet no works available that present industrial real-world implementations of lakehouses for the manufacturing sector. As a consequence, it remains unclear under which circumstances and with which architectures lakehouses may represent a viable alternative to conventional data warehouses and data lakes for enterprises in this field. In an effort to address this gap, we conducted a case study on a real-world industrial case in which an analytics department developed, tested and leveraged a lakehouse approach for the management and analytical exploitation of manufacturing data from the shop floor with a volume in the magnitude of terabytes. This lakehouse has been in use for several months in a productive environment with various analytical workloads and hence represents a suitable candidate for investigating architectural decisions.

Section 2 explains the methodological approach of our study and introduces the industrial setting of the case. Section 3 then presents the first results of our within-case analyses by discussing the architecture of the data platform and pointing to interesting architectural decisions. Finally, Section 4 concludes our work.

[1] https://delta.io, accessed: 30.04.2024
[2] https://hudi.apache.org, accessed: 30.04.2024
[3] https://iceberg.apache.org, accessed: 30.04.2024
[4] https://spark.apache.org, accessed: 30.04.2024
[5] https://flink.apache.org, accessed: 30.04.2024

2. The Manufacturing Case

Case studies are generally recognized as an appropriate research method for complex topics and topics where the context needs to be taken into consideration [25]. Both applies to the research field of data platforms, as the construction of such platforms constitutes complex tasks that are subject to rapid developments and innovations, while architectural decisions and practical experiences of enterprises likely depend on contextual aspects, such as the domain and size of the enterprise, the available data volumes and the analytical use cases.

In the scope of our case study for the manufacturing case, we conducted interviews with one solution architect and one data engineer of the responsible analytics department, who both have been involved in the development of the corresponding lakehouse. The interviews lasted between 45 and 60 minutes and followed a semi-structured approach, for which questions had already been prepared in advance, but were spontaneously supplemented by follow-up questions during the interviews. The asked questions were related to the context of the manufacturing case, including the available data sources and the requirements for the data analyses that are supposed to be performed, as well as various architectural aspects of the developed lakehouse (cf. Section 3). Afterwards, the transcribed answers of the interview participants were structured and analyzed with the help of qualitative coding techniques.

Table 1 summarizes important characteristics of the investigated case, including details about the source data and the intended analytical use cases.

Table 1
Characteristics of the investigated manufacturing case.

Source Data:	Machine and Sensor Data
Source Systems:	Manufacturing Execution System
Data Types:	Structured, Unstructured
Data Volumes:	Terabytes
Analytical Workloads:	Reporting, OLAP, Machine Learning, Near-realtime Reporting
Analytics Types:	Descriptive, Diagnostic, Predictive
Users:	Business Users, Data Analysts, Data Scientists

This manufacturing case is situated at a large-scale, globally operating manufacturer, which develops and produces technical components of high volume. Along the shop floor, manufacturing machines and sensors col-

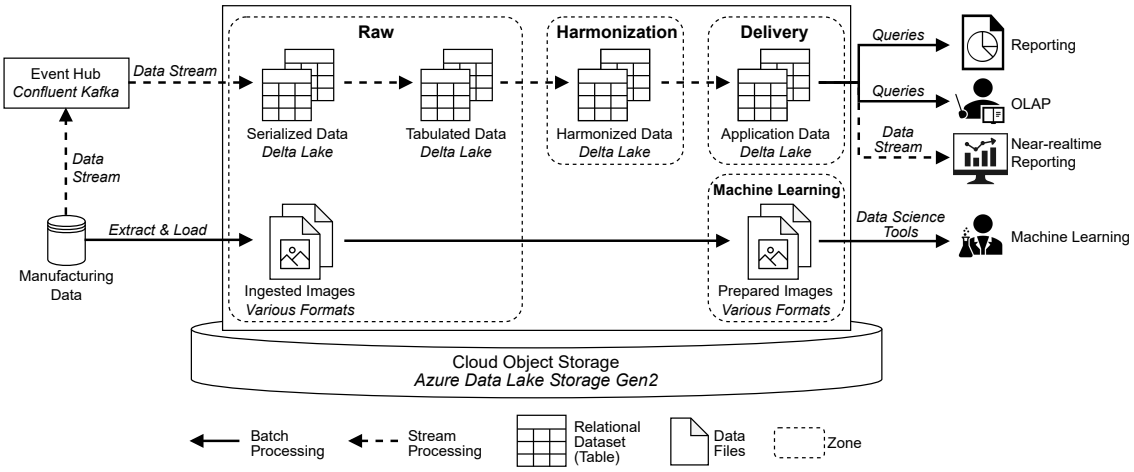

Figure 1: The lakehouse architecture as applied in the manufacturing case.

lect various types of data, reflecting the quality of the workpieces and the condition of the machines. This data is primarily made available in a Manufacturing Execution System (MES) and ingested as data stream into a lakehouse-like data platform. Besides structured measurement values of the machines and sensors, this data also includes graphical images of particularly error-prone parts of the produced technical components. A dedicated analytics department constisting out of multiple solution architects and data engineers is involved in this case and responsible for the development and operation of the data platform. The goal of this data platform is the collection, management, preparation and analysis of the generated manufacturing data in order to enable self-service analytics for business users and data analysts. This includes traditional reporting and OLAP workloads, but also near-realtime reporting, since some analysis results are supposed to be displayed on dashboards along the shop floor. In addition, data scientists pursue to train machine learning models from the available image data, which should enable the automatically detection of faulty workpieces with the help of image classification techniques.

3. The Lakehouse Architecture

As part of the case study, we examine the architecture of the lakehouse data platform that is employed in the manufacturing case and, in particular, focus on interesting architectural decisions that were made during its development, for example with respect to aspects such as data organization, data modeling and data flow.

During the interviews, both participants confirmed to us that after several months of operation, they are satisfied with both the lakehouse approach in general, as well as the architecture they have developed, since it allowed them to meet their requirements in terms of performance and the required range of analytical workloads. We therefore assume that their architectural decisions regarding the lakehouse have practically proven their suitability at least in the medium term.

In the context of this work, lakehouses can largely be regarded as data lakes that consist of a distributed file system or an object storage and have been enhanced for additional data warehousing capabilities with the help of specific frameworks, such as Delta Lake, Apache Hudi and Apache Iceberg [12]. Therefore, we utilized the *Data Lake Architecture Framework* by Giebler et al. [26] as a reference to guide and structure our within-case analysis for the architecture of the data platform. However, we limited ourselves to the aspects *Data Organization*, *Data Modeling*, *Data Flow*, *Data Storage* and *Infrastructure*, as these were most strongly covered during the interviews. Figure 1 illustrates the current architecture of the lakehouse that is applied in the investigated manufacturing case. Here, the lakehouse is depicted in the center, while the data sources and analytical workloads are shown on the left- and right-hand side. The lakehouse consists of multiple zones [27], which separate data of different granularity, quality and purpose. The individual aspects of this architecture according to the Data Lake Architecture Framework are discussed in the following sections.

3.1. Infrastructure and Data Storage

The lakehouse is built on top of Azure Data Lake Storage Gen2[6] as underlying storage system, which is a cloud object storage offered by Microsoft Azure[7]. For the batch and stream processing of data within the data platform, Databricks[8] is leveraged. In the investigated case, this processing engine heavily utilizes the Delta Lake framework, which persists the data on the object storage as data files in the column-oriented format Apache Parquet [9] and allows to represent collections of such data files as self-contained tables with relational properties on the logical level. In addition, the cloud-managed service Confluent Kafka[10] is used as event hub that buffers the streaming data before the ingestion into the lakehouse. For reporting, the data analysts primarily rely on Power BI[11] in connection with SQL endpoints provided by Databricks[12], while OLAP workloads are performed by executing SQL queries directly through Databricks. Near-realtime reporting and machine learning are also conducted via Databricks with the help of the corresponding libraries for streaming[13] and machine learning[14]. The decisions for the technologies that are utilized in this data platform were made on the basis of a prototype. According to the interview participants, an important factor for their selection was that they were already offered as part of a cloud solution and considered more mature than comparable open source projects at that time.

3.2. Data Flow

The measurement data that is generated by the machines and sensors on the shop floor is in the JSON format and, as illustrated in Figure 1, first forwarded to the event hub, which temporarily stores and buffers the data before it is loaded into the lakehouse. This way, the utilized stream processing engine can control the rate at which the data is ingested and processed. The stream processing engine then reads the newly arrived data from the event hub and stores it in tables of the Delta Lake framework within the *Raw Zone* of the lakehouse. It is worth noting that these first tables of the data flow store the serialized, unflattened data. This means that instead of de-serializing the incoming data records and mapping their embedded fields to corresponding columns of a table, the whole JSON string as such is stored in one text-based column of the table. This allows to store the raw data without needing perform potentially error-prone transformations during the ingestion, which could potentially lead to loss of information or failures. At the same time, the additional data management features of the Delta Lake framework, such as time travel capabilities and ACID guarantees, can be used on the raw data. From there, the ingested data is further processed in the scope of multiple stream processing steps, where the data is persisted on the object storage after each step. Within the scope of these steps, the raw JSON data is also de-serialized and transformed into a tabular representation, where each table consists of potentially multiple columns of suitable types that can be mapped to fields of the original source data.

The subsequent processing steps are responsible for different tasks such as flattening, cleansing and integrating the data, so that consolidated and unified data is made available in the *Harmonization Zone*. Starting from the Harmonization Zone, the data is then further preprocessed and prepared for the available analytical use cases, which includes the execution of filtering operations and the calculation of aggregations according to specific columns. As a result, the *Delivery* Zone finally contains data that is optimized for the respective analytical applications in terms of quality, granularity and performance. This data can then be consumed either via SQL queries for the reporting and OLAP workloads or as data streams, which allow to regularly update the dashboards on the shop floor. However, the stream processing jobs that are defined on the processing engine are not running continuously. Instead, they are only started at regular time intervals, for example once per day, and then process all data that has accumulated on the event hub in the meantime. After a certain runtime, the processing jobs are suspended again. As the analytics use cases of the manufacturing scenario are not time-critical according to the participants of the interviews, this approach allows to save resources and costs in comparison to continuously or long running stream processing jobs.

Besides the structured machine and sensor data, also the graphical images of the produced workpieces need to be managed on the data platform in order to enable machine learning tasks. In contrast to the measurement data, this unstructured, binary data is not transmitted to the event hub and instead regularly ingested into the Raw Zone of the object storage via batch processing. Furthermore, the image files are not embedded into tables of the Delta Lake framework, but persisted in their original image formats on the object storage. Within the data platform, the images are only processed and prepared in a rudimentary manner and almost directly transferred to

[6] https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction, accessed: 30.04.2024
[7] https://azure.microsoft.com, accessed: 30.04.2024
[8] https://azure.microsoft.com/en-us/products/databricks, accessed: 30.04.2024
[9] https://parquet.apache.org, accessed: 30.04.2024
[10] https://www.confluent.io, accessed: 30.04.2024
[11] https://www.microsoft.com/en-us/power-platform/products/power-bi, accessed: 30.04.2024
[12] https://docs.databricks.com/en/compute/sql-warehouse/create-sql-warehouse.html, accessed: 30.04.2024
[13] https://docs.databricks.com/en/structured-streaming/index.html, accessed: 30.04.2024
[14] https://docs.databricks.com/en/machine-learning/train-model/mllib.html, accessed: 30.04.2024

the *Machine Learning Zone*, where they can be used by data scientists for machine learning experiments.

3.3. Data Modeling

According to the participants of the interviews, data modeling is carried out only informally, meaning that data models are created ad-hoc depending on the structure of the data that is provided by the data sources and the requirements of the analytical use cases. Hence, widely researched modelling approaches, such as normal forms [28], multi-dimensional modeling [29] or the Data Vault concept [30] are not explicitly applied. Moreover, the data is intentionally de-normalized, as this enables a higher query performance at the cost of increased storage space, which represents a reasonable trade-off for cloud environments due to the lower costs for storage space in comparison to computational resources.

3.4. Data Organization

The lakehouse architecture of the manufacturing case applies a zone model for organizing the data of different granularity, quality and application-specificity within the lakehouse. In particular, this zone model defines four zones: A *Raw Zone*, which stores the raw and only slightly processed data, a *Harmonization Zone*, in which the relational, consolidated and unified data resides, a *Delivery Zone* for pre-aggregated, application-specific data and a *Machine Learning Zone* that holds data that is relevant for machine learning activities. These zones can be roughly mapped to the *Raw Zone*, *Harmonized Zone*, *Delivery Zone* and *Explorative Zone* of the Zone Reference Model (ZRM), which was originally proposed by Giebler at al. [27] for the data organization within data lakes. According to the participants of the interviews, both the use of a zone model in general, as well as the zones that were specifically selected for this case have proven their suitability. Therefore, it can be concluded that zone models for data lakes appear also to be relevant in the context of lakehouses and may be a suitable choice for organizing the data in these kind of data platforms.

4. Conclusion

This paper presented a real-world case in which a lakehouse has been developed and leveraged for the management and analysis of manufacturing data in industrial practice. In the scope of our study, we particularly focused on the architecture of the lakehouse, as well as the industrial setting and underlying goals. In this course, some interesting architectural decisions could be observed: Our study revealed that in the investigated case a) the periodic execution of stream processing jobs is preferred over continuously running stream processing jobs for economic reasons, b) that ingested raw data is stored and managed as serialized JSON strings in tables and not in raw text files, c) that data modelling is carried out only informally and that de-normalization techniques are applied in order to increase the query performance at the expense of higher costs for storage space and d) that zone models appear to be a suitable technique for data organization within lakehouses.

In future work, we plan to compare this case with several other real-world cases from different domains in terms of architectural similarities, the motivational factors for enterprises to utilize lakehouses, practical experiences and encountered challenges. This way, we want to further expand the findings of our work and become capable of generalizing them.

References

[1] J. Wang, C. Xu, J. Zhang, R. Zhong, Big Data Analytics for Intelligent Manufacturing Systems: A Review, Journal of Manufacturing Systems 62 (2022) 738–752. doi:10.1016/j.jmsy.2021.03.005.

[2] A. Dogan, D. Birant, Machine learning and data mining in manufacturing, Expert Systems with Applications 166 (2021) 114060. doi:10.1016/j.eswa.2020.114060.

[3] A. Rehman, S. Naz, I. Razzak, Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities, Multimedia Systems 28 (2022) 1339–1371. doi:10.1007/s00530-020-00736-8.

[4] M. Javaid, A. Haleem, I. H. Khan, R. Suman, Understanding the potential applications of Artificial Intelligence in Agriculture Sector, Advanced Agrochem 2 (2023) 15–30. doi:10.1016/j.aac.2022.10.001.

[5] C. Gröger, Industrial Analytics – An Overview, it - Information Technology 64 (2022) 55–65. doi:10.1515/itit-2021-0066.

[6] W. H. Inmon, Building the data warehouse, fourth edition, 4th ed. ed., Wiley, Indianapolis, Ind., 2005.

[7] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang, Leveraging the Data Lake: Current State and Challenges, in: Big Data Analytics and Knowledge Discovery, volume 11708 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 179–188. doi:10.1007/978-3-030-27520-4_13.

[8] A. Nambiar, D. Mundra, An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management, Big Data and Cognitive Computing 6 (2022) 132. doi:10.3390/bdcc6040132.

[9] M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia, Lake-

house: a new generation of open platforms that unify data warehousing and advanced analytics, in: Proceedings of CIDR, volume 8, 2021.

[10] R. Bose, Advanced analytics: opportunities and challenges, Industrial Management & Data Systems 109 (2009) 155–172. doi:10.1108/02635570910930073.

[11] D. Feinberg, P. Russom, N. Showell, Hype Cycle for Data Management 2022, 2022.

[12] J. Schneider, C. Gröger, A. Lutsch, H. Schwarz, B. Mitschang, The Lakehouse: State of the Art on Concepts and Technologies, SN Computer Science 5 (2024). doi:10.1007/s42979-024-02737-0.

[13] M. Armbrust, T. Das, L. Sun, B. Yavuz, S. e. a. Zhu, Delta Lake: High-performance ACID Table Storage over Cloud Object Stores, Proceedings of the VLDB Endowment 13 (2020) 3411–3424. doi:10.14778/3415478.3415560.

[14] P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, M. Zaharia, Analyzing and Comparing Lakehouse Storage Systems, in: Proceedings of the 13th Annual Conference on Innovative Data Systems Research, 2023.

[15] W. H. Inmon, M. Levins, R. Srivastava, Building the data lakehouse, first printing ed., Technics Publications, Basking Ridge, NJ, 2021.

[16] S. Ait Errami, H. Hajji, K. Ait El Kadi, H. Badir, Spatial big data architecture: From Data Warehouses and Data Lakes to the LakeHouse, Journal of Parallel and Distributed Computing 176 (2023) 70–79. doi:10.1016/j.jpdc.2023.02.007.

[17] D. Orescanin, T. Hlupic, Data Lakehouse - a Novel Step in Analytics Architecture, in: 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), IEEE, 2021, pp. 1242–1246. doi:10.23919/MIPRO52101.2021.9597091.

[18] P. Ren, S. Li, W. Hou, W. Zheng, Z. e. a. Li, MHDP: An Efficient Data Lake Platform for Medical Multi-source Heterogeneous Data, in: Web Information Systems and Applications, volume 12999 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2021, pp. 727–738. doi:10.1007/978-3-030-87571-8_63.

[19] Q. Xiao, W. Zheng, C. Mao, W. Hou, H. e. a. Lan, MHDML: Construction of a Medical Lakehouse for Multi-source Heterogeneous Data, in: Health Information Science, volume 13705 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, Cham, 2022, pp. 127–135. doi:10.1007/978-3-031-20627-6_12.

[20] T. Zhao, N. Hai, W. Li, W. Zheng, Y. e. a. Zhang, Multi-modal Medical Data Exploration Based on Data Lake, in: Health Information Science, volume 14305 of *Lecture Notes in Computer Science*, Springer Nature Singapore, Singapore, 2023, pp. 213–222. doi:10.1007/978-981-99-7108-4_18.

[21] E. Begoli, I. Goethert, K. Knight, A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Megabiobanks, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 4643–4651. doi:10.1109/BigData52589.2021.9671534.

[22] D. Tovarnak, M. Racek, P. Velan, Cloud Native Data Platform for Network Telemetry and Analytics, in: 2021 17th International Conference on Network and Service Management (CNSM), IEEE, 2021, pp. 394–396. doi:10.23919/CNSM52442.2021.9615568.

[23] F. Chen, Z. Yan, L. Gu, Towards Low-Latency Big Data Infrastructure at Sangfor, in: Emerging Information Security and Applications, volume 1641 of *Communications in Computer and Information Science*, Springer Nature Switzerland, Cham, 2022, pp. 37–54. doi:10.1007/978-3-031-23098-1_3.

[24] S. A. Errami, H. Hajji, K. A. E. Kadi, H. Badir, Managing Spatial Big Data on the Data LakeHouse, in: Emerging Trends in Intelligent Systems & Network Security, volume 147 of *Lecture Notes on Data Engineering and Communications Technologies*, Springer International Publishing, Cham, 2023, pp. 323–331. doi:10.1007/978-3-031-15191-0_31.

[25] J. Dul, T. Hak, Case study methodology in business research, Routledge, London and New York, 2008.

[26] C. Giebler, C. Gröger, E. Hoos, R. Eichler, H. e. a. Schwarz, The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture, in: Conference for Database Systems for Business, Technology and Web (BTW), volume 70469, 2021, pp. 351–370.

[27] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang, A Zone Reference Model for Enterprise-Grade Data Lake Management, in: 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), IEEE, 2020, pp. 57–66. doi:10.1109/EDOC49727.2020.00017.

[28] C. J. Date, An introduction to database systems: C.J. Date, 8. ed. ed., Pearson Addison-Wesley, Boston, 2004.

[29] M. Z. Iqbal, G. Mustafa, N. Sarwar, S. H. Wajid, J. e. a. Nasir, A Review of Star Schema and Snowflakes Schema, in: Intelligent Technologies and Applications, volume 1198 of *Communications in Computer and Information Science*, Springer Singapore, Singapore, 2020, pp. 129–140. doi:10.1007/978-981-15-5232-8_12.

[30] D. Linstedt, M. Olschimke, Building a Scalable Data Warehouse with Data Vault 2.0, Morgan Kaufmann, 2015.