Exploring Database Lakehouse Architecture Design Patterns: Best Practices and Considerations

Krishna Prisad Bajgai[1] (M.Phil. Scholar-ICT); Dr. Bhoj Raj Ghimire[2] (PhD)

[1;2]Faculty of Information and Communication Technology Nepal Open University, Lalitpur, Nepal

Publication Date: 2025/02/25

Abstract: Organizations face challenges in managing diverse, large-scale datasets while ensuring scalability, efficiency, and quality. Traditional data lakes and warehouses often fall short in modern big data environments. The Lakehouse architecture unit both, but cloud implementation faces issues like optimized ingestion, efficient storage, and integration of multiple data engines. Sectors like healthcare and agriculture struggle with real-time data and IoT, leading to inefficiencies. Current research highlights gaps in performance, scalability, and the integration of advanced analytics. Future work should focus on improving large dataset handling, real-time processing, and machine learning integration for better decision-making and performance.

Keywords: *Lakehouse Architecture, Big Data Integration, Cloud Computing, Real-Time Data Processing, Federated Governance, Machine Learning.*

How to Cite: Krishna Prisad Bajgai; Dr. Bhoj Raj Ghimire (2025). Exploring Database Lakehouse Architecture Design Patterns: Best Practices and Considerations. *International Journal of Innovative Science and Research Technology*, 10(2), 550-557. https://doi.org/10.5281/zenodo.14921215

I. INTRODUCTION

Organizations are increasingly facing challenges in managing and integrating diverse, large-scale datasets while ensuring scalability, efficiency, and data quality. Traditional data management architectures, such as data lakes and data warehouses, often fail to meet the demands of modern big data environments, leading to inefficiencies and limited support for data-driven decision-making [1] .The Lakehouse architecture has emerged as a promising solution, unifying the capabilities of both data warehouses and data lakes. However, its implementation in cloud-based environments presents complexities, including the need for optimized data ingestion, efficient storage mechanisms, and seamless integration of multiple data processing engines [8]. Sectors such as healthcare and agriculture struggle with managing real-time data, IoT devices, and diverse data sources, leading to inefficiencies in decision-making and operations [5][7]. Further, traditional storage formats in lakehouses do not support graph analytics, and federated governance in data mesh architectures requires further research [4][6]. Key challenges such as performance optimization and query execution also remain critical in the implementation of Lakehouse systems [13]. The limitations of current research highlight the need for further advancements in flexibility, scalability, and real-world deployment of lakehouses. Future work should focus on improving the handling of large datasets, real-time data processing, and the integration of machine learning to enhance decision-making [9]. Additionally, empirical studies are needed to validate the practical effectiveness of these systems across diverse industries, with a particular focus on cost optimization, scalability, and performance under large-scale deployments [3].

The limitations of current research highlight the need for further advancements in flexibility, scalability, and real-world deployment of lakehouses [1]. Future work should focus on improving the handling of large datasets, real-time data processing, and the integration of machine learning to enhance decision-making [8]. Additionally, empirical studies are needed to validate the practical effectiveness of these systems across diverse industries, with a particular focus on cost optimization, scalability, and performance under large-scale deployments [9].

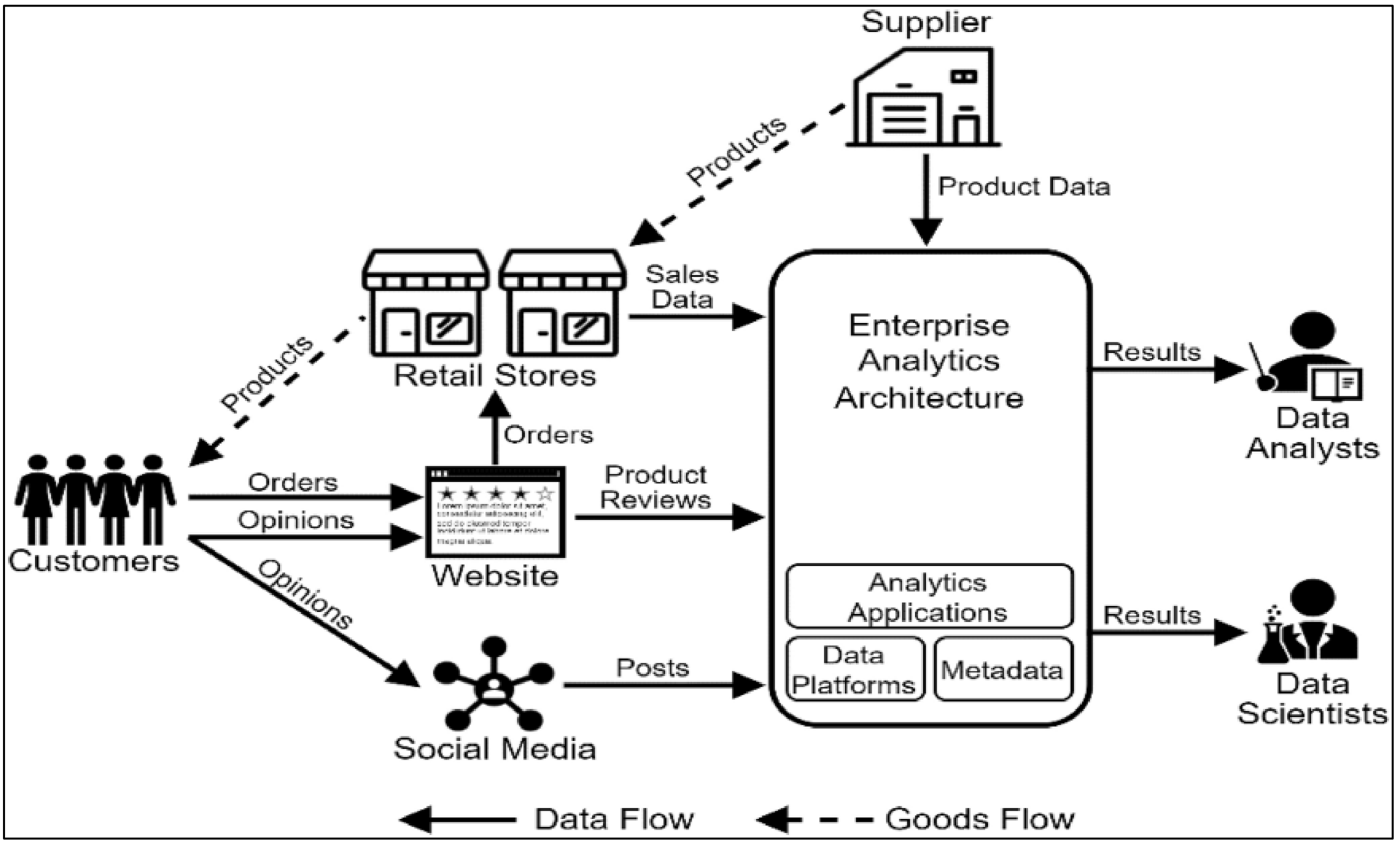

Fig 1 From: The Lakehouse: State of the Art on Concepts and Technologies

- *Problem Statement:*

Many of the Organizations face growing challenges in managing large-scale, diverse datasets as traditional architectures like data lakes and warehouses often fall short in scalability and efficiency [1]. The Lakehouse architecture offers a unified solution, but its cloud-based implementation poses complexities such as data ingestion, storage optimization, and processing integration [12]. Industries like healthcare and agriculture struggle with real-time data and IoT devices, highlighting the need for advancements in performance, scalability, and machine learning integration [5][7]. Future research must address these gaps to optimize cost, enhance decision-making, and validate Lakehouse systems in diverse, large-scale deployments [9].

Organizations struggle to manage large, diverse datasets due to the inefficiencies of traditional data architectures like warehouses and lakes [1]. While the Lakehouse architecture offers a unified solution, challenges persist in cloud-based implementations, including data ingestion, storage optimization, governance, and query performance [13]. Sectors like healthcare and agriculture face additional hurdles with real-time and graph data, necessitating innovative solutions to enhance scalability, integration, and decision-making [5][6][7].

II. LITERATURE REVIEW

Organizations today face significant challenges in managing and integrating diverse, large-scale datasets while ensuring scalability, efficiency, and data quality. Traditional centralized architectures, such as data warehouses and data lakes, often fail to meet the demands of modern big data landscapes. The lack of integration between these architectures results in inefficiencies, operational bottlenecks, and limited support for data-driven decision-making [1][2].

The emerging "Lakehouse" architecture offers a unified solution that combines the advanced analytics capabilities of data warehouses with the scalability and flexibility of data lakes. However, implementing lakehouses in cloud-based environments introduces complexities, including the need for optimized data ingestion, efficient storage mechanisms, and seamless sintegration of multiple data processing engines.[4][10].

In particular, the healthcare and agriculture sectors illustrate the challenges of managing diverse data sources, such as IoT devices, sensors, and real-time monitoring systems. Existing systems struggle to handle the velocity and variety of data, leading to inefficiencies in clinical decision-making and precision farming applications [5]

Additionally, managing graph data in lakehouse environments poses unique challenges, as traditional columnar storage formats like Parquet and ORC are not optimized for graph analytics. This limitation hinders performance for operations such as neighbor retrieval and label filtering, necessitating novel storage solutions tailored for graph data[6].

Organizations also encounter difficulties in implementing federated governance and ensuring data quality within distributed architectures like data meshes. Effective data governance and quality management are critical for supporting complex big data landscapes and enhancing decision-making processes [3]

Furthermore, optimizing query execution and performance remains a key challenge in Lakehouse systems. The concept of a unified Query Optimizer as a Service (QOaaS) has been proposed to address these issues, ensuring efficient data processing across diverse engines[11][13].

To overcome these limitations, the development of scalable, efficient, and unified architectures is crucial. By addressing the gaps in data integration, governance, and performance, the Lakehouse paradigm can unlock the potential for enhanced data strategies and innovation across industries [11][8].

➢ *Data Used :*

The following summarizes the types of data used in the referenced studies, highlighting their relevance to evaluating or validating proposed cloud data lakehouse architectures.

- *Publicly Available and Synthetic Datasets*

Studies frequently utilized synthetic datasets, publicly available datasets, or data from specific use cases to evaluate the performance and scalability of lakehouse systems. These datasets are often employed to test key architectural improvements in scalability, query optimization, and data processing performance[1][8].

- *Organizational Data (Structured, Semi-Structured, and Unstructured)*

Several studies focused on real-world organizational data, encompassing diverse formats and structures derived from various departments and systems. These studies evaluated lakehouse architectures for their capability to unify data management and enable decision-making[3][2].

- *Agricultural Data (Agriculture 4.0)*

Studies centered on data relevant to precision agriculture, including IoT sensor data, satellite imagery, and weather data. These datasets demonstrated the integration and processing capabilities of lakehouse systems in agriculture[5].

- *Graph Data*

Research into graph data focused on the Labeled Property Graph (LPG) model, evaluating how effectively lakehouse architectures could manage graph-specific operations, such as neighbor retrieval and label filtering[6].

- *Healthcare Data*

The studies explored structured and unstructured healthcare datasets, including Electronic Health Records (EHRs), imaging data, sensor data, and patient-generated data, to evaluate real-time processing capabilities in healthcare data lakes[7].

- *Open Data Formats and TPC-DS Benchmark*

Open data formats, such as Apache Parquet and ORC, were emphasized, along with the TPC-DS benchmark, a standard for decision support systems, to assess system performance[10][13].

- *Conceptual Focus on Architectures*

Certain studies concentrated on architectural discussions, without using specific datasets, highlighting integration, query optimization, and ingestion challenges in lakehouse systems[9][11][12].

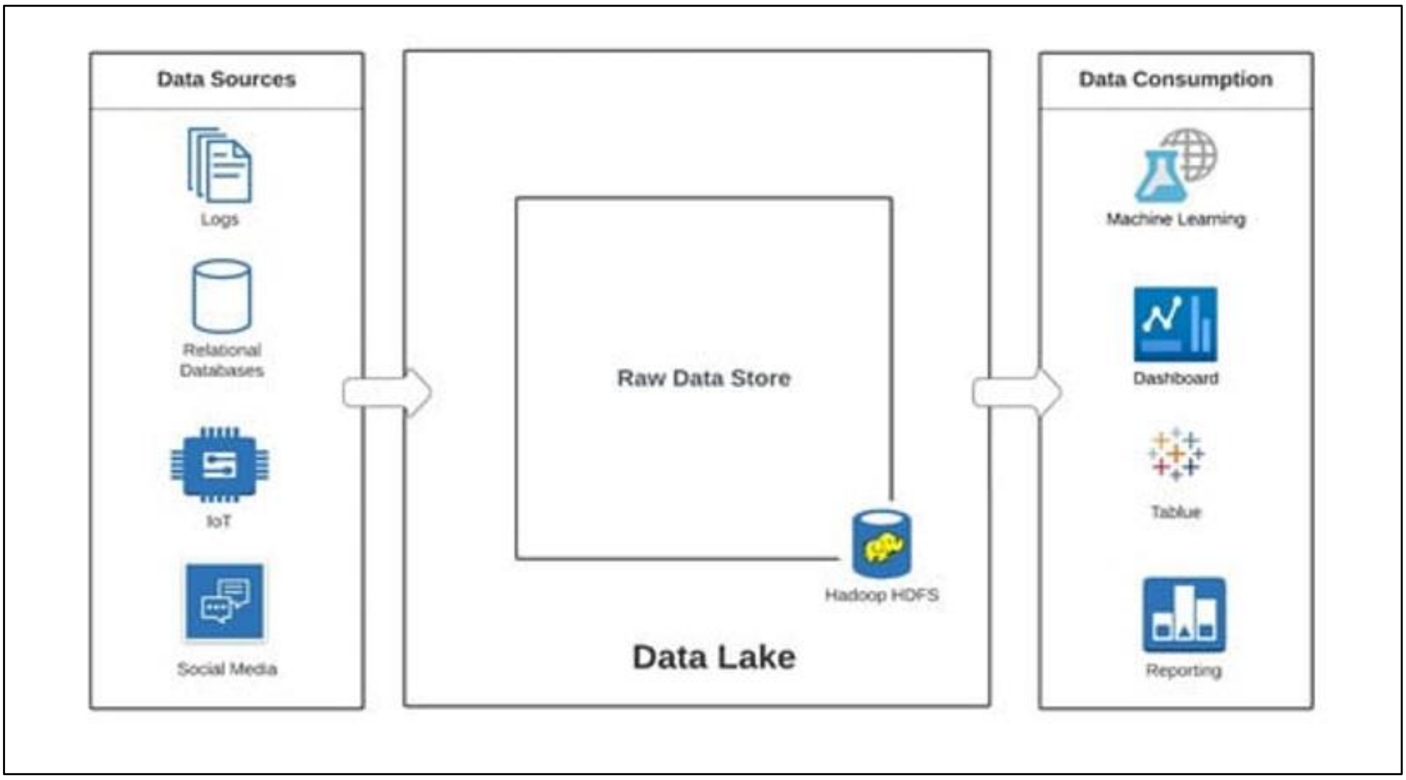

Fig 2 Mono-Zone Architecture.

- *Research Data in CRIS*

Studies focused on Current Research Information Systems (CRIS) data, covering projects, personnel, organizational units, funding programs, research outputs (publications, patents), facilities, and events[3][14].

A. Methods and Technologies :

➢ *Architectural Frameworks for Data Lakehouses*

Utilized cloud services like AWS, Azure, and Google Cloud for data lakehouses. Data processing frameworks (Apache Spark, Hadoop) and storage solutions (S3, Delta Lake) are discussed as key components for solving data integration challenges.[1][2][8][9].

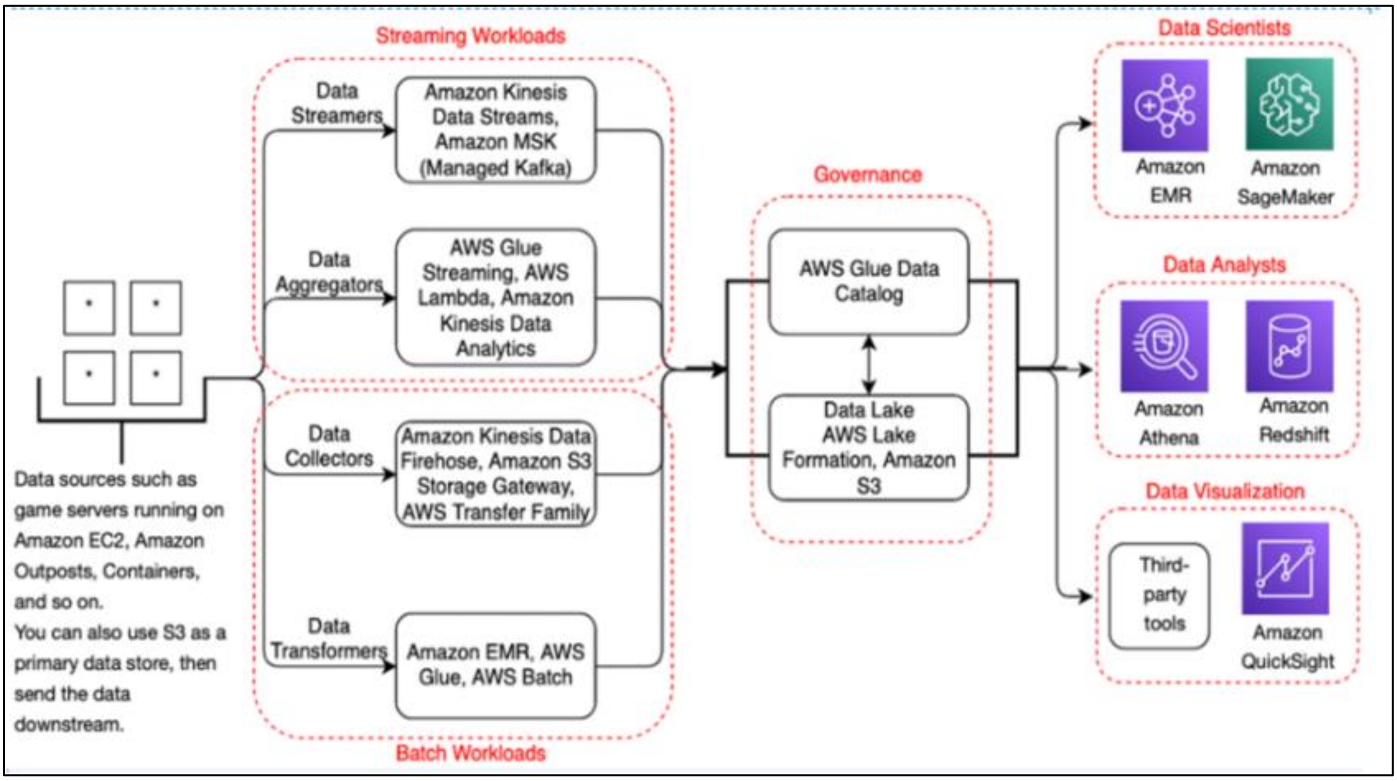
Streaming Workloads
Data Streamers
Amazon Kinesis Data Streams, Amazon MSK (Managed Kafka)
Data Aggregators
AWS Glue Streaming, AWS Lambda, Amazon Kinesis Data Analytics
Data Collectors
Amazon Kinesis Data Firehose, Amazon S3 Storage Gateway, AWS Transfer Family
Data Transformers
Amazon EMR, AWS Glue, AWS Batch
Batch Workloads
Data sources such as game servers running on Amazon EC2, Amazon Outposts, Containers, and so on.
You can also use S3 as a primary data store, then send the data downstream.
Governance
AWS Glue Data Catalog
Data Lake AWS Lake Formation, Amazon S3
Data Scientists
Amazon EMR
Amazon SageMaker
Data Analysts
Amazon Athena
Amazon Redshift
Data Visualization
Third-party tools
Amazon QuickSight

Fig 3 High-level Framework for Building a Data Lake on AWS.

➢ *Integration of OLAP and OLTP Systems*

Novel approaches to managing data consistency and schema enforcement by integrating OLAP and OLTP within lakehouse architectures were proposed.[8][9]*[10]*.

➢ *Data Mesh Architecture*

Emphasized a domain-oriented decentralized approach, treating data as a product, assigning ownership to domain teams, and implementing self-serve data platforms for enhanced accessibility and management.[3]

➢ *Federated Governance in Data Mesh*

Proposed federated computational governance, which ensures consistent policies across domains while granting local autonomy.[4]

➢ *Cloud and Distributed Computing for Agriculture*

Reviewed centralized and distributed cloud architectures for agriculture. These strategies optimize data storage, processing, and analysis for Agriculture 4.0.[5]

➢ *GraphAr for Graph Data in Data Lakes*

Introduced GraphAr as a specialized storage scheme leveraging Parquet for graph data management in data lakes. It focuses on labeled property graphs (LPG) and employs innovative encoding/decoding techniques.[6]

➢ *Healthcare Data Lakes*

Explored technologies for real-time data processing in healthcare data lakes, including:

- **Data Ingestion**: Platforms like Apache Kafka and Apache Flink.
- **Data Storage**: Scalable solutions such as HDFS and cloud storage.
- **Data Processing**: Real-time analytics frameworks.
- **Data Mining**: Machine learning for predictive analytics and personalized care.[7]

➢ *Lakehouse Architecture Innovations*

Built on open, direct-access data formats and incorporates features like ACID transactions, data versioning, and indexing. Supports machine learning workloads effectively.[8]

➢ *Comparative Reviews*

Analyzed strengths and weaknesses of existing DW and DL technologies, highlighting desired features for Lakehouse systems.[9]

➤ *Query Optimizer as a Service (QOaaS)*

Proposed QOaaS architecture using Substrait to standardize plan specifications, integrated with Microsoft's Fabric ecosystem.[10]

➤ *Data Ingestion Patterns*

Suggested a design pattern tailored for cloud-based architectures to improve big data ingestion processes.[11]

- *Photon: A Fast Query Engine*

Introduced Photon, a C++ vectorized query engine by Databricks, optimized for SQL and Apache Spark's DataFrame API in Lakehouse environments.[13]

- *Combining Data Lakes and Data Wrangling*

Presented a combined approach to use data lakes for central storage and data wrangling techniques to ensure data quality.[14]

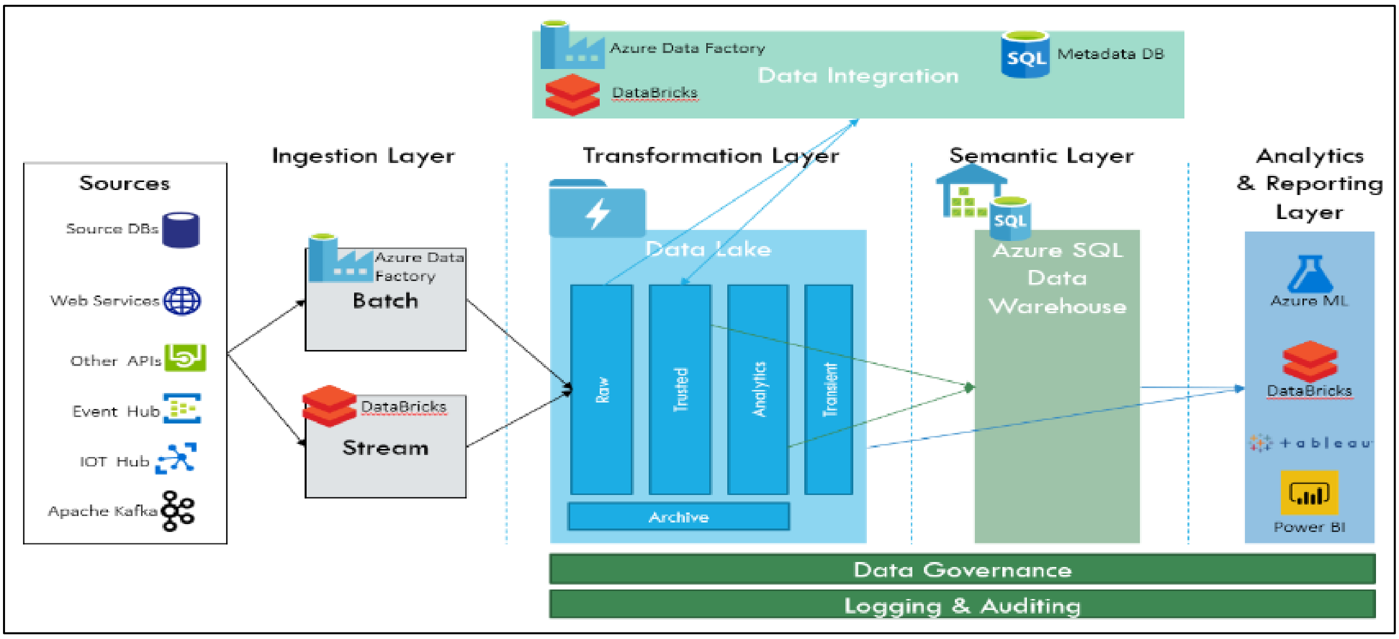

Fig 4 Modern Data Warehouse Architecture

B. Accuracy Evaluation Methods :

Aravind Nuthalapati (2024), This paper primarily focuses on best practices and future directions for data lake-houses but does not specify a formal method for accuracy evaluation.[1]

Jan Schneider et al. (2024) Evaluates the performance of the lakehouse model using the **TPC-DS benchmark**, comparing query execution times, data ingestion rates, and resource utilization.

The results show that the Lakehouse system built on Parquet is competitive with popular cloud data warehouses.[2]

Otmane Azeroual and Radka Nacheva (2023), Conceptual discussion on data mesh and its architectural benefits; however, no formal accuracy evaluation or performance benchmarks are included.[3]

Anton Dolhopolov et al. (2024), Discusses federated governance in data mesh architecture but does not provide empirical accuracy evaluations.[4]

Olivier Debauche et al. (2021), Reviews cloud and distributed architectures for agriculture data management but lacks empirical accuracy benchmarks.[5]

Xue Li et al. (2024), Evaluates **GraphAr**'s performance by benchmarking against conventional **Parquet** and **Acero-based methods**. Key metrics include speedup in neighbor retrieval, label filtering, and end-to-end workload efficiency[6].

Mitul Tilala et al. (2022), Explores healthcare data lakes but does not provide formal accuracy evaluation methods.[7]

Michael Armbrust et al. (2021), Performance of the Lakehouse system is benchmarked using **TPC-DS**, demonstrating advanced query performance comparable to cloud data warehouses.[8]

Dipankar Mazumdar et al. (2023), Provides conceptual discussions on the benefits of lakehouses without presenting formal accuracy evaluations.[9]

Ahmed Harby and Farhana Zulkernine (2022), A comparative review of data warehouse and lakehouse technologies, but no empirical evaluations are reported.[10]

Rana Alotaibi et al. (2024), Discusses the potential performance optimizations of **Query Optimizer as a Service (QOaaS)** but lacks empirical accuracy benchmarks.[11]

Chiara Rucco et al. (2024), Proposes a cloud-based design pattern for optimizing data ingestion but does not specify accuracy evaluation methods.[12]

Alexander Behm et al. (2022), Benchmarks **Photon**, a query engine for lakehouses, against cloud data warehouses and engines. Performance metrics include query execution times and resource efficiency.[13]

Otmane Azeroual et al. (2022), Combines data lakes with wrangling techniques but does not provide empirical performance or accuracy benchmarks.[14]

- *Validation and Verification of Proposed Models:*

Focuses on best practices and architectural principles for cloud-based lakehouses but does not provide experimental validation or verification[1]. Validates the proposed lakehouse architecture through a **comparative analysis with existing data warehouse (DW) and data lake (DL) systems**, demonstrating how the lakehouse addresses their limitations[2].

Discusses the effectiveness of the data mesh approach for enhancing scalability, data integration, and decision-making but does not include empirical validation or real-world testing[3]. Proposes integrating federated governance into data mesh architectures for improved data management but lacks empirical validation or verification[4]. Analyzes cloud and distributed architectures in agriculture data management but does not present validation or verification methodologies[5]

Validates **GraphAr's effectiveness** through **performance benchmarks**, achieving a 3,283× speedup for neighbor retrieval, 6.0× for label filtering, and 29.5× for end-to-end workloads compared to traditional methods[6]. Discusses the potential of real-time data processing in healthcare data lakes to enhance clinical decision-making but does not include empirical validation or testing[7].

Validates the lakehouse concept through **industry trends and logical reasoning**, comparing it to existing data management architectures but without real-world empirical testing[8]. Provides conceptual insights into lakehouse systems but does not include experimental validation or real-world testing[9]. Compares lakehouse architectures with data warehouses and data lakes but does not include formal validation methodologies[10]. Validates the **QOaaS concept** using prototypes and its integration within the Fabric ecosystem but does not detail specific validation techniques[11]. Proposes a cloud-based design pattern for optimizing data ingestion but does not provide validation or empirical testing details[12].Validates **Photon's performance** through benchmarks, demonstrating significant speed improvements over existing cloud data warehouses in SQL workloads [13]. Suggests that integrating data lakes and data wrangling processes enhances data quality but does not include empirical validation.[14].

III. RESULTS AND FINDINGS OF THE STUDIES

The studies highlight advancements in data management and architecture, focusing on scalable solutions like lakehouses and data meshes. Aravind Nuthalapati (2024) demonstrates the scalability and cost-efficiency of cloud-based lakehouse architectures. Jan Schneider et al. (2024) address challenges in traditional systems and enhance performance by unifying analytics and warehousing. Otmane Azeroual & Radka Nacheva (2023) advocate for decentralized data ownership through data mesh architecture. Anton Dolhopolov et al. (2024) emphasize federated governance to improve data management. In agriculture, Olivier Debauche et al. (2021) find cloud integration enhances centralized data management. Xue Li et al. (2024) improve graph data management in data lakes with GraphAr, while Mitul Tilala et al. (2022) explore real-time data processing in healthcare. Michael Armbrust et al. (2021) present lakehouses as solutions to data staleness and reliability. Dipankar Mazumdar et al. (2023) show lakehouses support structured and unstructured data workloads. Ahmed Harby & Farhana Zulkernine (2022) discuss lakehouse strengths in efficient data processing. Rana Alotaibi et al. (2024) propose QOaaS to optimize query execution. Chiara Rucco et al. (2024) explore a cloud-based design pattern for data ingestion. Alexander Behm et al. (2022) report up to 12x query performance improvements with Photon. Otmane Azeroualet al. (2022) combine data wrangling with data lakes to enhance data quality.

IV. LIMITATION, AND FUTURE WORK

- *Aravind Nuthalapati (2024)*

- **Limitations**: The paper discusses the general advantages of cloud-based lakehouse architectures but does not delve deeply into challenges such as **handling massive datasets** and **real-time data processing**.

- **Future Work**: Further research should focus on improving **system flexibility**, **handling more complex data types**, and integrating **AI or machine learning** to enhance data insights.[1]

- *Jan Schneider et al. (2024)*

- **Limitations**: While the lakehouse model addresses key challenges, the paper does not explicitly discuss **performance degradation under large-scale data** or the **integration of machine learning workflows**.

- **Future Work**: Research should focus on **scalability challenges**, real-time data handling, and **enhancing integration** with machine learning systems to improve decision-making.[2]

- *Otmane Azeroual and Radka Nacheva (2023)*

- **Limitations**: The paper does not explicitly identify specific limitations but suggests the **need for empirical validation** of the data mesh approach in real-world environments.

- **Future Work**: Further research could include **empirical studies** to evaluate the practical effectiveness of the proposed model in diverse organizational contexts.[3]

➢ *Anton Dolhopolov et al. (2024)*

- **Limitations**: The paper does not provide extensive real-world examples or data to support the federated governance model in practice.

- **Future Work**: Future research should focus on **scalability issues** and validating the **effectiveness of federated governance** across larger datasets in various domains.[4]

➢ *Olivier Debauche et al. (2021)*

- **Limitations**: The paper highlights the benefits of cloud and distributed architectures in agriculture but does not delve into the **scalability of these solutions** under large-scale deployments or **integration challenges**.

- **Future Work**: Future studies could include **empirical evaluations** of these architectures in real agricultural settings, focusing on **scalability** and **integration with other technologies**.[5]

➢ *Xue Li et al. (2024)*

- **Limitations**: **Graph data storage schemes** in data lakes need further refinement for larger datasets, and the approach does not discuss **performance issues** when scaling.

- **Future Work**: Future research should explore the **scalability of GraphAr**, especially with very large datasets, and enhance **integration with distributed systems**.[6]

➢ *Mitul Tilala et al. (2022)*

- **Limitations**: The paper focuses on real-time data processing in healthcare but does not address challenges in **scaling real-time systems** or **integration with legacy healthcare systems**.

- **Future Work**: Future research should examine **scalability** in large healthcare systems and explore **integration with AI-driven diagnostic tools**.[7]

➢ *Michael Armbrust et al. (2021)*

- **Limitations**: The paper presents the lakehouse as a solution but acknowledges that **real-world performance** and the **practicality of large-scale implementation** require further evaluation.

- **Future Work**: Future research should explore **additional features**, **optimize performance** for various data workloads, and address challenges faced during **implementation**.[8]

➢ *Dipankar Mazumdar et al. (2023)*

- **Limitations**: The article highlights benefits but does not explore the **specific challenges in real-world deployments** or the **cost implications** of lakehouses at scale.

- **Future Work**: Future work should involve **empirical validation** of the lakehouse model in diverse industries and focus on **cost optimization** and **real-world scalability**.[9]

➢ *Ahmed Harby and Farhana Zulkernine (2022)*

- **Limitations**: The paper does not provide detailed case studies or **empirical evidence** regarding the actual deployment of lakehouses in large-scale systems.

- **Future Work**: Future studies should **implement the architecture** in real-world scenarios to assess **scalability** and **performance across industries**.[10]

➢ *Rana Alotaibi et al. (2024)*

- **Limitations**: While QOaaS is promising, the paper acknowledges the challenge of **implementing flexible cardinality estimation** and **adapting it to different cost models**.

- **Future Work**: Research should focus on **prototyping QOaaS**, refining its approach, and **evaluating its real-world performance** in large systems.[11]

➢ *Chiara Rucco et al. (2024)*

- **Limitations**: The paper suggests using a **cloud-based design pattern** for data ingestion but does not explore the **limitations in processing speed** or **data variety** under high-load scenarios.

- **Future Work**: Future research should address the **scalability** of the ingestion pattern and integrate **AI-driven optimizations** for processing diverse data types.[12]

➢ *Alexander Behm et al. (2022)*

- **Limitations**: The study focuses on Photon's query engine performance but does not discuss its **scalability issues** or its **effectiveness across different data workloads**.

- **Future Work**: Future research could focus on **optimizing Photon** for a broader range of workloads and exploring integration with other **data processing frameworks**.[13]

➢ *Otmane Azeroual et al. (2022)*

- **Limitations**: The paper focuses on combining data lakes with wrangling but does not deeply analyze **real-time processing challenges** or **large-scale implementation constraints**.

- **Future Work**: Future work could involve **empirical validation** of the proposed model in **real-world CRIS implementations**.[14].

V. CONCLUSION

The studies collectively highlight advancements in data architecture, emphasizing scalability, integration, and performance improvements. Lakehouses unify data lakes and warehouses, addressing challenges like data staleness, cost, and diverse workloads. Innovations such as data mesh architecture, federated governance, GraphAr, and QOaaS enhance data management, decision-making, and query optimization. Applications in healthcare, agriculture, and big data scenarios demonstrate improvements in real-time processing, data quality, and ingestion efficiency. These findings underscore the transformative potential of modern data systems in addressing diverse industry needs.

REFERENCES

[1]. Nuthalapati, A. (2024). Architecting data lake-houses in the cloud: Best practices and future directions. 32 citations.

[2]. Schneider, J., Gröger, C., Lutsch, A., Schwarz, H., & Mitschang, B. (2024). The Lakehouse: State of the Art on Concepts and Technologies. 119 citations.

[3]. Azeroual, O., & Nacheva, R. (2023). Data Mesh for Managing Complex Big Data Landscapes and Enhancing Decision Making in Organizations. 31 citations.

[4]. Dolhopolov, A., Castelltort, A., & Laurent, A. (2024). Implementing Federated Governance in Data Mesh Architecture. 40 citations.

[5]. Debauche, O., Mahmoudi, S., Manneback, P., & Lebeau, F. (2021). Cloud and Distributed Architectures for Data Management in Agriculture 4.0: Review and Future Trends. 55 citations.

[6]. Li, X., Zeng, W., Wang, Z., Zhu, D., Xu, J., Yu, W., & Zhou, J. (2024). GraphAr: An Efficient Storage Scheme for Graph Data in Data Lakes. 77 citations.

[7]. Tilala, M., Pamulaparthyvenkata, S., Chawda, A. D., & Benke, A. P. (2022). Explore the Technologies and Architectures Enabling Real-Time Data Processing Within Healthcare Data Lakes. 30 citations.

[8]. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics. 53 citations.

[9]. Mazumdar, D., Hughes, J., & Onofré, J. B. (2023). The Data Lakehouse: Data Warehousing and More. 31 citations.

[10]. Harby, A., & Zulkernine, F. (2022). From Data Warehouse to Lakehouse: A Comparative Review. 31 citations.

[11]. Alotaibi, R., Tian, Y., Grafberger, S., Camacho-Rodríguez, J., Bruno, N., Kroth, B., et al. (2024). Towards Query Optimizer as a Service (QOaaS) in a Unified LakeHouse Ecosystem. 41 citations.

[12]. Rucco, C., Longo, A., & Saad, M. (2024). Optimizing Data Ingestion for Big Data: A Cloud-Based Design Pattern Approach. 32 citations.

[13]. Behm, A., Palkar, S., Agarwal, U., Armstrong, T., Cashman, D., Dave, A., et al. (2022). Photon: A Fast Query Engine for Lakehouse Systems. 59 citations.

[14]. Azeroual, O., Schöpfel, J., Ivanovic, D., & Nikiforova, A. (2022). Combining Data Lake and Data Wrangling for Ensuring Data Quality in CRIS. 35 citations.