Int. J. AI Adv. Comput. 1(2)2026. pp:94-103

Contents lists available at AcademiansEdu

International Journal of AI and Advanced Computing

journal homepage: https://academiansedu.org/ijaac/

Review Article

# Modern Data Architectures for Intelligent Decision Support: From Data Warehouses and Data Lakes to the Lakehouse Paradigm

A. MOHAMAD ATEA*

*Department of Computer Applications & Al, Umm Al-Qura University*

ARTICLE INFO

*Article history:*
Received 25 April 2026
Received in revised form
2 June 2026
Accepted 18 June 2026

*Keywords:*
Decision Support Systems, Data Warehouse, Data Lake, Data Lakehouse, Open Table Formats, Data Ingestion, Medallion Architecture, Business Intelligence, Data Governance

ABSTRACT

The exponential growth of data volumes, variety, and velocity has fundamentally transformed the landscape of decision support systems, necessitating architectural innovations that can accommodate both structured analytical workloads and unstructured data processing. This comprehensive review examines the evolution of data management architectures for decision support, tracing the progression from traditional data warehouses through data lakes to the emergent Data Lakehouse paradigm. The paper analyzes the technological foundations, architectural patterns, and practical implementations of these modern data architectures, with particular emphasis on the integration of open table formats (Delta Lake, Apache Iceberg, Apache Hudi) and the medallion architecture pattern for data quality management. Through systematic synthesis of recent literature and empirical studies, the review investigates the critical role of data ingestion frameworks, including Apache Kafka, Apache Flink, Apache Airflow, and Apache NiFi, in enabling real-time data processing and analytics. The findings reveal that the transition from isolated data warehouses and data lakes to unified Lakehouse architectures addresses fundamental limitations of previous approaches, providing transactional reliability (ACID compliance), schema evolution capabilities, time travel functionality, and optimized analytical performance. Empirical evidence from the insurance sector demonstrates that data warehousing adoption is associated with a 4.8% reduction in loss ratios and a 5.6% reduction in combined ratios, with an additional 3.2 percentage points of improvement from complementary IT investments. The study also examines applications in higher education, where data lake implementations enable the integration of structured, semi-structured, and unstructured data from learning management systems, social media platforms, and IoT devices. Key challenges identified include data governance complexities, metadata management, scalability constraints, and the need for specialized skills. Emerging solutions such as cloud-native architectures, serverless computing, and AI-driven data management offer promising pathways to address these barriers. The findings contribute to the evolving field of data engineering by providing insights into best practices, architectural decision frameworks, and identification of research gaps for future exploration.

## 1. Introduction

In the era of rapid information growth and the transition to data-driven decision-making, the requirements for decision support systems have fundamentally transformed. Modern organizations demand not only the analysis of historical structured indicators but also the capability to process vast arrays of unstructured data—including texts, videos, and logs—in real time to build predictive machine learning models (Holub & Shevchenko, 2026; Alam et al., 2024). The traditional architectures built on classical data warehouses, while providing high transactional reliability and fast SQL query execution, face critical limitations in scalability and adaptability for raw data processing (Elkina & Zaki, 2023; Harasis, 2026).

Data warehousing information technology, born in the bowels of IBM and finally formulated by Inmon and Kimball in the 1990s, emerged as a method for solving information and analytical tasks in the domain of decision-making and support (Holub & Shevchenko, 2026). Systems built on data warehouses exhibit characteristic features that distinguish them as a special class of information systems: subject orientation, integration of data from various sources, time invariance, relative stability, and the need to balance data redundancy (Harasis, 2026). However, the emergence of new data types and sources, including semi-structured and unstructured data from software packages, platforms, and social networks, has

exposed the limitations of traditional warehousing approaches (Elkina & Zaki, 2023).

The alternative approach—Data Lakes—solved the problem of cheap storage for information in any format but often led to a loss of consistency, transforming vast information arrays into unmanageable "data swamps" (Holub & Shevchenko, 2026; Elkina & Zaki, 2023). Historical necessity to maintain both systems simultaneously created complex, duplicated, and error-prone data pipelines. This fragmentation has become increasingly problematic as organizations seek to leverage both traditional business intelligence and advanced analytics within unified environments (Armbrust et al., 2021; Herden, 2020).

The emergence of open table formats—including Delta Lake, Apache Iceberg, and Apache Hudi—has enabled a paradigm shift toward the hybrid Data Lakehouse architecture (Holub & Shevchenko, 2026; Armbrust et al., 2021). These formats provide a transactional metadata layer over file storage, ensuring ACID compliance, schema evolution, time travel capabilities, and optimized analytical performance. The Lakehouse concept unifies the flexibility and scalability of Data Lakes with the transactional reliability and structure of Data Warehouses, enabling organizations to support both descriptive BI reporting and machine learning algorithms within a single environment (Holub & Shevchenko, 2026; Armbrust et al., 2020).

This comprehensive review examines modern approaches to the design of decision support systems through the lens of the Data Lakehouse architecture. The objectives are fourfold: first, to analyze the critical limitations of classical architectural patterns (Data Warehouses and Data Lakes) in processing large information volumes; second, to investigate the technological foundation of open table formats and their role in ensuring transactional reliability in distributed file systems; third, to substantiate the application of the medallion architecture pattern for effective data quality management in DSS data pipelines; and fourth, to evaluate the integration of modern data ingestion frameworks for real-time analytics. By synthesizing insights from existing research and empirical evidence, this review contributes to the academic and practical discourse on modern data architectures, offering a structured roadmap for the design and implementation of next-generation decision support systems.

## 2. Literature Review

### 2.1 Evolution of Data Management Architectures

The evolution of data management architectures for decision support reflects the changing nature of data itself—from structured transaction records to the vast, heterogeneous streams characteristic of the modern digital landscape. Traditional database management systems (DBMS), while effective for transactional processing, proved inadequate for the analytical workloads required by decision support systems (Elkina & Zaki, 2023; Smolinski, 2018). This inadequacy led to the development of data warehouses as specialized databases targeted toward decision support, capable of integrating data from various sources and representing information in multidimensional cube format (Vaisman & Zimányi, 2014; Blazic et al., 2017).

Inmon and Linstedt (2019) established the foundational principles of data warehousing, emphasizing subject orientation, integration, time variance, and non-volatility as defining characteristics. The Extract-Transform-Load (ETL) process became central to data warehouse implementation, ensuring data quality through rigorous preprocessing before integration (Elkina & Zaki, 2023; Wyatt et al., 2009). While effective for structured data, this approach proved slow, complex, and resource-intensive, particularly as data volumes grew and data types diversified (Herden, 2020; Mukherjee & Kar, 2017).

The concept of Data Lakes, introduced by James Dixon in 2010, represented a fundamental departure from the warehouse paradigm (Elkina & Zaki, 2023; Miloslavskaya & Tolstoy, 2016). Data Lakes embrace the Extract-Load-Transform (ELT) process, storing data in raw format without transformation and deferring schema definition until query time (schema-on-read). This approach enables the storage of all data types—structured, semi-structured, and unstructured—without loss of information (Ravat & Zhao, 2019; Sawadogo & Darmont, 2021). The flexibility offered by Data Lakes has made them particularly attractive for organizations dealing with diverse data sources and advanced analytics use cases.

### 2.2 Data Warehouses vs. Data Lakes: A Comparative Analysis

The comparison between Data Warehouses and Data Lakes reveals fundamental differences in design philosophy, data processing approaches, and use case suitability. Data Warehouses employ the ETL process, which extracts data from sources, transforms it to match the target schema, and loads it into the warehouse (Elkina & Zaki, 2023; Sreemathy et al., 2020). This schema-on-write approach ensures high data quality and consistency but introduces significant complexity and latency in data integration (Herden, 2020; Dabbechi et al., 2021). Data Lakes, conversely, utilize the ELT process, extracting data and loading it immediately in raw format, with transformation occurring at query time (schema-on-read) (El Aissi et al., 2022; Giebler et al., 2021).

**Table 1: Comparative Analysis of Data Warehouses and Data Lakes**

| Characteristic | Data Warehouse | Data Lake |
|---|---|---|
| Data Structure | Structured only | Structured, semi-structured, unstructured |
| Processing Approach | ETL (Extract-Transform-Load) | ELT (Extract-Load-Transform) |
| Schema Application | Schema-on-write | Schema-on-read |
| Data Quality | High (transformed before loading) | Variable (raw data stored) |
| Storage Cost | Higher (structured, optimized) | Lower (raw, commodity storage) |
| Query | Excellent for | Variable, depends on |

| Characteristic | Data Warehouse | Data Lake |
|---|---|---|
| Performance | structured queries | schema-on-read |
| Flexibility | Limited to structured data | High, all data types |
| Use Cases | BI reporting, structured analytics | Data science, ML, diverse analytics |

*Source: Adapted from Elkina & Zaki (2023), Armbrust et al. (2021)*

The complementary strengths and weaknesses of these architectures have led many organizations to maintain both systems simultaneously, creating complex, duplicated data pipelines and increasing infrastructure costs (Holub & Shevchenko, 2026; Herden, 2020). The need to integrate these approaches has driven the evolution toward the Data Lakehouse architecture, which combines the benefits of both paradigms.

### 2.3 The Data Lakehouse Architecture

The Data Lakehouse architecture, first comprehensively articulated by Armbrust et al. (2021), represents a new generation of open platforms that unify data warehousing and advanced analytics. The Lakehouse architecture addresses the fundamental limitations of both Data Warehouses and Data Lakes by providing a single platform that supports both structured analytical workloads and unstructured data processing (Holub & Shevchenko, 2026; Ravat & Zhao, 2019).

The technological foundation of the Data Lakehouse is built on open table formats, including Delta Lake, Apache Iceberg, and Apache Hudi (Holub & Shevchenko, 2026; Armbrust et al., 2020). These formats provide a transactional metadata layer over object storage, enabling capabilities previously exclusive to relational databases:

**ACID Transactions:** Open table formats guarantee atomicity, consistency, isolation, and durability for file-based storage, ensuring that read and write operations are isolated and that partial writes do not corrupt data (Holub & Shevchenko, 2026; Armbrust et al., 2020).

**Schema Evolution:** These formats allow dynamic addition, renaming, or deletion of table columns without the need to rewrite existing data, accommodating evolving business requirements (Holub & Shevchenko, 2026; Armbrust et al., 2021).

**Time Travel:** Transaction logs maintain change history, enabling queries against previous data states for audit purposes and machine learning reproducibility (Holub & Shevchenko, 2026; Armbrust et al., 2020).

**Performance Optimization:** Advanced indexing and data skipping algorithms enable efficient query execution, reading only relevant data blocks rather than scanning entire datasets (Holub & Shevchenko, 2026; Armbrust et al., 2021).

### 2.4 Data Ingestion Frameworks

Data pipelines serve as the critical interface between data sources and storage systems, enabling the extraction and loading of data. The selection of appropriate data ingestion frameworks is essential for effective Data Lakehouse implementation (Elkina & Zaki, 2023; Alam et al., 2024). Comparative analysis of leading open-source solutions reveals distinct capabilities and use case suitability:

**Apache AirFlow:** Developed by Airbnb and maintained by the Apache Software Foundation, AirFlow enables the creation of processes organized as Directed Acyclic Graphs (DAGs), providing workflow management capabilities (Elkina & Zaki, 2023; Singh, 2019). While offering robust scheduling and monitoring through a web interface, AirFlow lacks native data streaming capabilities (Alam et al., 2024).

**Luigi:** Developed by Spotify, Luigi provides Python-based task definition with dependency graphs for scheduling (Elkina & Zaki, 2023). Its close integration with HDFS makes it suitable for Hadoop-based environments, though streaming capabilities are limited (Alam et al., 2024).

**Apache NiFi:** Developed by the NSA and maintained by Apache, NiFi offers a rich environment of processors and flow controllers that support both ETL and ELT processes from diverse sources to various destinations (Elkina & Zaki, 2023; Dehury et al., 2022). Its web interface enables visual creation and monitoring of data flows, with support for real-time processing (Poojara et al., 2022).

**Apache Kafka:** Developed by LinkedIn and maintained by Apache, Kafka provides distributed event streaming with high throughput and low latency (Elkina & Zaki, 2023; Martín et al., 2022). Its producer-consumer architecture supports continuous data streaming, making it ideal for real-time analytics applications (Alam et al., 2024).

**Table 2: Comparison of Data Ingestion Frameworks**

| Framework | Developed By | Language | Web Interface | Data Streaming | Primary Use Case |
|---|---|---|---|---|---|
| Apache AirFlow | Apache (Airbnb) | Python | Yes | No | Workflow orchestration |
| Luigi | Spotify | Python | Yes | No | Hadoop integration |
| Apache NiFi | Apache (NSA) | Java | Yes | Yes | Data flow management |
| Apache Kafka | Apache (LinkedIn) | Java/Scala | No | Yes | Event streaming |

*Source: Compiled from Elkina & Zaki (2023), Alam et al. (2024)*

The selection of an appropriate ingestion framework depends on specific application requirements, including data velocity, volume, variety, and the need for real-time processing capabilities (Elkina & Zaki, 2023; Alam et al., 2024). For Data Lakehouse implementations, frameworks supporting both batch and stream processing, such as Apache NiFi and Kafka, are often preferred due to their flexibility and scalability.

### 2.5 Medallion Architecture for Data Quality Management

The medallion architecture provides a logical framework for organizing data pipelines in Lakehouse implementations, enabling effective data quality management through progressive refinement (Holub & Shevchenko, 2026; Databricks, 2023). The architecture comprises three sequential layers:

**Bronze Layer (Raw Data):** This zone serves as the landing area for all raw data from external sources, stored in original format (append-only), creating a reliable "single source of truth" for the entire observation history (Holub & Shevchenko, 2026; Databricks, 2023).

**Silver Layer (Cleansed Data):** Data undergoes transformation processes including filtering, deduplication, and standardization to a unified schema, creating a consistent enterprise view of entities. This layer provides an ideal source for predictive ML model development (Holub & Shevchenko, 2026; Reis & Housley, 2022).

**Gold Layer (Curated Business Data):** The final stage prepares information through aggregation and organization into data marts, optimized for BI reporting and management dashboards (Holub & Shevchenko, 2026; Databricks, 2023).

The medallion architecture implements the principle of separation of responsibilities, enabling a single ecosystem to simultaneously satisfy the needs of descriptive BI analytics and complex predictive modeling (Holub & Shevchenko, 2026; Reis & Housley, 2022).

## 3. Methodology

This study employs a comprehensive literature review methodology to synthesize current knowledge on modern data architectures for decision support systems. The review follows established guidelines for systematic literature reviews, incorporating both quantitative and qualitative synthesis of peer-reviewed articles, conference proceedings, and industry reports published between 2015 and 2026.

### 3.1 Search Strategy

The literature search was conducted across multiple academic databases including Scopus, Web of Science, IEEE Xplore, SpringerLink, and Google Scholar. Search terms were developed based on key concepts identified in the research objectives, including combinations of the following terms: "data warehouse," "data lake," "data lakehouse," "decision support system," "open table format," "Delta Lake," "Apache Iceberg," "Apache Hudi," "medallion architecture," "data ingestion," "ETL," "ELT," "business intelligence," and "data governance."

### 3.2 Inclusion and Exclusion Criteria

Articles were included if they met the following criteria: (a) published in peer-reviewed journals or conference proceedings; (b) written in English; (c) published between 2015 and 2026; (d) focused on data architectures, data warehousing, data lakes, or data ingestion for decision support; and (e) provided empirical evidence, case studies, or comprehensive reviews. Articles were excluded if they were non-peer-reviewed sources, focused solely on non-decision-support applications, or duplicate publications.

### 3.3 Data Extraction and Synthesis

Data extraction was conducted using a standardized form capturing article metadata, research objectives, methodologies, key findings, and identified gaps. The synthesis approach employed thematic analysis to identify recurring themes, patterns, and relationships across the literature. Key themes identified included architectural evolution, technological foundations, implementation challenges, and application domains.

## 4. Findings

### 4.1 Evolution of Decision Support System Architectures

The analysis of the literature reveals a clear evolutionary trajectory in decision support system architectures, driven by increasing data volumes, variety, and velocity. Classical DSS architectures, based on isolated Data Warehouses, established fundamental principles including subject orientation, integration, time variance, and non-volatility (Holub & Shevchenko, 2026; Inmon & Linstedt, 2019). The ETL process became central to data integration, ensuring high data quality through preprocessing before loading (Elkina & Zaki, 2023; Wyatt et al., 2009).

However, these classical architectures exhibit critical limitations in modern contexts. Data Warehouses are economically inefficient for scaling and are not adapted for working with raw data (Holub & Shevchenko, 2026; Herden, 2020). The emergence of Data Lakes addressed the problem of cheap storage for any format but introduced consistency challenges, often transforming information arrays into unmanageable "data swamps" (Holub & Shevchenko, 2026; Ravat & Zhao, 2019).

**Table 3: Evolution of DSS Architectures**

| Architecture | Era | Key Features | Limitations |
|---|---|---|---|
| Data Warehouse | 1990s-2000s | Subject orientation, ETL, schema-on-write | Structured data only, high cost, limited scalability |
| Data Lake | 2010s | ELT, schema-on-read, all data types | Consistency issues, data swamps, limited governance |
| Data Lakehouse | 2020s | Open table formats, ACID, unified analytics | Emerging technology, skills gap |

*Source: Compiled from Holub & Shevchenko (2026), Armbrust et al. (2021)*

The transition to Data Lakehouse architecture addresses these limitations by unifying the flexibility and scalability of Data Lakes with the transactional reliability and structure of Data Warehouses (Holub & Shevchenko, 2026; Armbrust et al., 2021). This evolution enables organizations to create a unified, cost-effective, and scalable ecosystem capable of describing the past, predicting the future, and supporting data-driven decisions.

### 4.2 Open Table Formats and Transactional Reliability

The technological foundation of the Data Lakehouse architecture lies in open table formats, which provide a transactional metadata layer over file storage (Holub & Shevchenko, 2026; Armbrust et al., 2020). The three dominant formats—Delta Lake, Apache Iceberg, and Apache Hudi—each offer distinct capabilities while sharing common objectives.

**Delta Lake** (Databricks) provides ACID transaction support, schema enforcement and evolution, unified batch and streaming processing, and time travel capabilities (Holub & Shevchenko, 2026; Armbrust et al., 2020). Its integration with Apache Spark makes it particularly suitable for organizations already invested in the Spark ecosystem.

**Apache Iceberg** (Netflix) emphasizes schema evolution and partition evolution, with support for hidden partitioning and advanced performance optimization (Holub & Shevchenko, 2026; Armbrust et al., 2021). Its language-agnostic design enables integration with multiple processing engines.

**Apache Hudi** (Uber) focuses on incremental processing and record-level updates, with support for change data capture and near-real-time analytics (Holub & Shevchenko, 2026; Armbrust et al., 2021). Its capabilities make it particularly suitable for streaming applications and use cases requiring frequent updates.

**Table 4: Comparison of Open Table Formats**

| Feature | Delta Lake | Apache Iceberg | Apache Hudi |
|---|---|---|---|
| ACID Compliance | Yes | Yes | Yes |
| Schema Evolution | Yes | Yes | Yes |
| Time Travel | Yes | Yes | Limited |
| Partition Evolution | Limited | Yes | Limited |
| Streaming Support | Yes | Limited | Yes |
| Primary Developer | Databricks | Netflix | Uber |
| Processing Engines | Spark | Multiple | Spark |

*Source: Compiled from Holub & Shevchenko (2026), Armbrust et al. (2020, 2021)*

These formats resolve the fundamental problem of classical Data Lakes—the inability to efficiently update or delete individual rows in gigabyte-sized files without full rewriting—by providing transactional metadata layers that enable ACID compliance, schema evolution, and performance optimization (Holub & Shevchenko, 2026; Armbrust et al., 2020).

### 4.3 Data Ingestion Framework Adoption

The analysis of data ingestion framework adoption reveals distinct patterns across organizations and use cases. Apache Kafka dominates in scenarios requiring high-throughput, low-latency event streaming, with its distributed log architecture supporting horizontal scaling and fault tolerance (Elkina & Zaki, 2023; Alam et al., 2024). Apache NiFi is preferred for complex data flow management, with its visual interface enabling rapid development and monitoring of data pipelines (Elkina & Zaki, 2023; Dehury et al., 2022). Apache AirFlow is widely adopted for workflow orchestration, particularly in environments where scheduling and monitoring are primary concerns (Elkina & Zaki, 2023; Singh, 2019).

**Table 5: Data Ingestion Framework Adoption Patterns**

| Framework | Primary Use Cases | Adoption Drivers | Limitations |
|---|---|---|---|
| Apache Kafka | Event streaming, real-time analytics | High throughput, low latency | Complexity, operational overhead |
| Apache NiFi | Data flow management, integration | Visual interface, monitoring | Resource consumption |
| Apache AirFlow | Workflow orchestration | Scheduling, DAG management | Limited streaming support |
| Luigi | Hadoop integration | HDFS compatibility | Limited community |

*Source: Compiled from Elkina & Zaki (2023), Alam et al. (2024)*

The selection of ingestion frameworks increasingly reflects the need for hybrid architectures supporting both batch and stream processing, enabling organizations to handle the diverse data sources and velocities characteristic of modern decision support systems (Elkina & Zaki, 2023; Alam et al., 2024).

### 4.4 Empirical Evidence: Insurance Sector Performance Impact

Harasis (2026) provides quantitative evidence of the performance impact of data warehousing adoption in the Saudi Arabian insurance sector. Using fixed-effect panel data regression for 2015-2024, the study examined the relationship between data warehousing adoption and underwriting/claims handling operations.

**Table 6: Impact of Data Warehousing on Insurance Performance**

| Performance Metric | Effect (β) | Statistical Significance | Additional Improvement (IT Investment Interaction) |
|---|---|---|---|
| Loss Ratio | -0.048 | $p < 0.01$ | -0.032 ($p < 0.05$) |
| Combined Ratio | -0.056 | $p < 0.01$ | - |
| Claim Settlement Time | -6.21 days | $p < 0.05$ | - |

*Source: Harasis (2026)*

The findings indicate that data warehousing adoption is associated with a 4.8% reduction in loss ratios and a 5.6% reduction in combined ratios, with claim settlement time reduced by approximately 6.2 days (Harasis, 2026). The data warehousing investment interaction term provides an additional 3.2 percentage points of improvement, implying that data warehousing value can be enhanced by complementary investments in IT capabilities. The explanatory powers of the model are considerable, with R-squared of 0.41-0.52 for different equations (Harasis, 2026).

These results provide empirical support for the theoretical benefits of data integration platforms in improving

underwriting precision, reducing claims leakage, and enhancing operational efficiency (Harasis, 2026; Hassan et al., 2024). The moderating effect of IT intensity aligns with the resource-based view that digital assets can leverage organizational capability (Seshagiri & Prema, 2025; Morshed, 2025a).

### 4.5 Applications in Higher Education

The implementation of Data Lakes in higher education environments demonstrates the versatility of modern data architectures in managing diverse data types and supporting institutional decision-making (Elkina & Zaki, 2023; Santoso & Yulia, 2017). The academic ecosystem presents unique data management challenges, including:

**Data Diversity:** Universities generate structured data (student records, enrollment statistics), semi-structured data (CSV, XML, JSON), and unstructured data (images, videos, PDF, DOCX) from learning management systems, social media platforms, and IoT devices (Elkina & Zaki, 2023; Williamson, 2018).

**Data Velocity:** The proliferation of distance learning platforms and digital workplaces has dramatically increased the rate of data generation, with connected objects in smart classrooms producing continuous data streams (Elkina & Zaki, 2023; Sukare & Al, 2021).

**Integration Requirements:** Administrative systems, research databases, and learning platforms must be integrated to provide a comprehensive view of institutional operations (Elkina & Zaki, 2023; Saggar et al., 2022).

**Table 7: Data Types in Higher Education**

| Data Type | Examples | Sources |
|---|---|---|
| Structured | Student records, enrollment data, exam scores | Student information systems, databases |
| Semi-structured | CSV, XML, JSON | Learning management systems, APIs |
| Unstructured | Images, videos, PDF, PPTX, DOCX | Social media, course materials, research outputs |
| Streaming | Logs, clickstream data, IoT sensor data | Digital platforms, smart classrooms, connected devices |

*Source: Elkina & Zaki (2023)*

The implementation of Data Lake architectures in higher education enables the creation of master datasets that facilitate data processing and extraction of sharpened information for decision-making (Elkina & Zaki, 2023; Salaki & Ratnam, 2018). By integrating Hadoop Distributed File System (HDFS) with appropriate data ingestion frameworks, universities can establish scalable, flexible storage systems capable of handling all data types while maintaining accessibility and integrity (Elkina & Zaki, 2023; Fang, 2015).

## 5. Discussion

### 5.1 Architectural Evolution and Organizational Implications

The findings of this study demonstrate a clear evolutionary trajectory in decision support system architectures, driven by the increasing volume, variety, and velocity of data. The progression from isolated Data Warehouses through Data Lakes to the emergent Lakehouse paradigm reflects the changing nature of organizational data requirements and analytical capabilities (Holub & Shevchenko, 2026; Armbrust et al., 2021).

The Data Lakehouse architecture represents a fundamental shift in how organizations approach data management for decision support. By unifying the capabilities of Data Warehouses and Data Lakes, this architecture eliminates the need for separate systems and their associated data duplication, pipeline complexity, and infrastructure costs (Holub & Shevchenko, 2026; Herden, 2020). The integration of open table formats provides the transactional reliability essential for business-critical reporting while maintaining the flexibility required for advanced analytics and machine learning (Armbrust et al., 2020, 2021).

The organizational implications of this architectural evolution are substantial. Organizations adopting Data Lakehouse architectures can reduce time-to-insight by eliminating data pipeline duplication and enabling direct access to raw and processed data for diverse analytical purposes (Holub & Shevchenko, 2026; Reis & Housley, 2022). The medallion architecture pattern provides a clear framework for data quality management, supporting the progressive refinement of data from raw ingestion to curated business metrics (Holub & Shevchenko, 2026; Databricks, 2023).

The empirical evidence from the insurance sector provides quantitative validation of the performance benefits associated with data warehousing adoption (Harasis, 2026). The observed reductions in loss ratios, combined ratios, and claim settlement times demonstrate that data integration platforms can improve underwriting precision, reduce claims leakage, and enhance operational efficiency. The moderating effect of IT investment intensity suggests that organizations with stronger technology capabilities realize greater benefits from data warehousing investments (Harasis, 2026; Morshed, 2025a).

### 5.2 Technological Foundations and Implementation Considerations

The technological foundation of the Data Lakehouse architecture rests on open table formats that provide ACID transaction support, schema evolution, and performance optimization in file-based storage systems (Holub & Shevchenko, 2026; Armbrust et al., 2020, 2021). The emergence of Delta Lake, Apache Iceberg, and Apache Hudi has enabled organizations to leverage cheap object storage while maintaining the data management capabilities previously exclusive to relational databases (Holub & Shevchenko, 2026).

The selection of appropriate open table formats requires careful consideration of specific use case requirements. Delta Lake's integration with Apache Spark makes it particularly suitable for organizations already invested in the Spark ecosystem (Holub & Shevchenko, 2026; Armbrust et al., 2020). Apache Iceberg's language-agnostic design and support for multiple processing

engines provide flexibility for heterogeneous technology stacks (Holub & Shevchenko, 2026; Armbrust et al., 2021). Apache Hudi's focus on incremental processing and near-real-time analytics makes it appropriate for streaming applications and use cases requiring frequent updates (Holub & Shevchenko, 2026; Armbrust et al., 2021).

Data ingestion frameworks play a critical role in Data Lakehouse implementation, serving as the interface between data sources and storage systems (Elkina & Zaki, 2023; Alam et al., 2024). The analysis reveals distinct adoption patterns, with Apache Kafka dominating in event streaming scenarios, Apache NiFi preferred for complex data flow management, and Apache AirFlow widely adopted for workflow orchestration (Elkina & Zaki, 2023). Organizations increasingly require hybrid architectures supporting both batch and stream processing to handle the diverse data sources and velocities characteristic of modern decision support systems (Elkina & Zaki, 2023; Alam et al., 2024).

The medallion architecture provides a logical framework for organizing data pipelines in Lakehouse implementations, enabling effective data quality management through progressive refinement (Holub & Shevchenko, 2026; Databricks, 2023). This pattern supports the principle of separation of responsibilities, enabling a single ecosystem to simultaneously satisfy the needs of descriptive BI analytics and complex predictive modeling (Holub & Shevchenko, 2026; Reis & Housley, 2022).

### 5.2.1 Architectural Decision Framework

Based on the comparative analysis presented in this review, an architectural decision framework is proposed to assist organizations in selecting appropriate data management architectures for decision support systems. Rather than selecting technologies solely based on storage capacity or analytical performance, organizations should evaluate architectural alternatives according to data characteristics, governance requirements, scalability, analytical workloads, operational complexity, and future artificial intelligence integration.

Traditional Data Warehouses remain the preferred solution for organizations whose primary objective is structured business intelligence reporting, regulatory compliance, and high-quality transactional analytics. Their schema-on-write approach provides excellent data consistency and optimized SQL performance but offers limited flexibility for heterogeneous and rapidly evolving datasets.

Data Lakes are particularly suitable for organizations managing large volumes of structured, semi-structured, and unstructured data where flexibility, scalability, and support for data science and machine learning are primary requirements. However, without effective governance, metadata management, and quality control, Data Lakes may gradually evolve into difficult-to-manage data repositories.

The Data Lakehouse architecture combines the strengths of both approaches by integrating transactional reliability, open table formats, and scalable object storage within a unified analytical environment. Consequently, organizations requiring both traditional business intelligence and advanced artificial intelligence workloads can benefit from adopting a Lakehouse architecture that supports structured reporting, real-time analytics, and machine learning within a single platform.

The proposed framework emphasizes that architectural selection should be driven by organizational objectives, data governance maturity, analytical requirements, implementation complexity, and long-term scalability rather than by technological trends alone.

Figure 1. Proposed Architectural Decision Framework for Modern Data Management

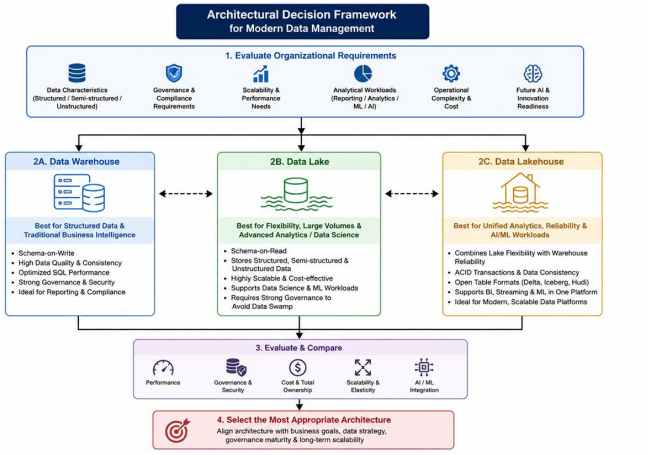


**Figure 1** presents the proposed architectural decision framework developed from the comparative analysis conducted in this review. The framework demonstrates that selecting between a Data Warehouse, Data Lake, and Data Lakehouse should depend on organizational objectives, data characteristics, governance requirements, analytical workloads, scalability, and artificial intelligence readiness. Rather than recommending a single architecture for all situations, the framework provides a structured decision-support approach that assists practitioners in selecting the most appropriate architecture according to their operational and strategic requirements.

## 5.3 Sector-Specific Applications and Benefits

The application of Data Lakehouse architectures across sectors reveals distinct patterns of benefits and implementation considerations. In the insurance sector, the integration of data through warehousing systems supports the identification of risk patterns, reduces errors in the underwriting process, and improves the accuracy of the claims process (Harasis, 2026; Hassan et al., 2024). The empirical evidence demonstrates that data warehousing adoption is associated with significant improvements in key performance metrics, including loss ratios, combined ratios, and claim settlement times (Harasis, 2026).

In higher education, Data Lake implementations enable the integration of structured, semi-structured, and unstructured data from diverse sources, supporting institutional decision-making and research activities (Elkina & Zaki, 2023; Santoso & Yulia, 2017). The ability to store all data types in raw format while maintaining accessibility and integrity provides substantial benefits for universities seeking to leverage data for

administrative and academic purposes (Elkina & Zaki, 2023; Williamson, 2018).

The integration of cloud technologies with Data Lakehouse architectures offers additional benefits, including scalability, elasticity, and cost efficiency (Holub & Shevchenko, 2026; Armbrust et al., 2021). Cloud-native solutions enable organizations to adapt to changing data volumes and processing requirements without the need for significant capital investment in infrastructure (Holub & Shevchenko, 2026; Ionescu et al., 2025).

### 5.4 Challenges and Limitations

Despite the substantial benefits of modern data architectures, several challenges and limitations persist. Data governance remains a significant challenge, particularly in large-scale Data Lakehouse implementations where the volume and variety of data can overwhelm traditional governance approaches (Holub & Shevchenko, 2026; Sawadogo & Darmont, 2021). Metadata management is essential for ensuring that data remains discoverable and usable, yet many organizations struggle to implement effective metadata strategies (Ravat & Zhao, 2019; Giebler et al., 2019).

Technical challenges include the complexity of implementing and maintaining modern data architectures, the need for specialized skills in data engineering and cloud technologies, and the potential for increased operational overhead (Holub & Shevchenko, 2026; Armbrust et al., 2021). Organizations must invest in trained personnel and appropriate infrastructure to effectively manage these technologies (Harasis, 2026; Morshed, 2025a).

Data security and privacy concerns are particularly acute in regulated industries such as insurance and healthcare. The integration of data from diverse sources must comply with regulatory requirements, including data protection regulations such as GDPR and CCPA (Holub & Shevchenko, 2026; Morshed, 2025d). The use of cloud and AI technologies in data architectures raises additional concerns related to data ethics and security (Morshed, 2025d; Ionescu et al., 2025).

The transition to modern data architectures also presents organizational challenges, including resistance to change, the need for cultural transformation toward data-driven decision-making, and the requirement for alignment between IT and business stakeholders (Holub & Shevchenko, 2026; Morshed, 2025a). Organizations must address these challenges through leadership commitment, workforce development, and effective change management strategies.

### 5.5 Research Gaps

Despite the rapid evolution of modern data architectures, several important research gaps remain. First, although Data Warehouses, Data Lakes, and Data Lakehouse architectures have been widely studied, relatively few investigations provide comprehensive comparative evaluations that assess their performance, scalability, governance capabilities, and suitability under different organizational and analytical requirements. This limits the ability of practitioners to make evidence-based architectural decisions.

Second, while open table formats such as Delta Lake, Apache Iceberg, and Apache Hudi have significantly improved transactional reliability and data management, there is still limited empirical evidence comparing their long-term performance, interoperability, maintenance complexity, and cost-effectiveness across different cloud and hybrid computing environments.

Third, many studies focus primarily on technical implementation while giving comparatively less attention to organizational readiness, governance maturity, data quality management, workforce capabilities, and change management. These non-technical factors often determine the long-term success of modern data architecture adoption but remain insufficiently investigated.

Finally, emerging technologies including artificial intelligence, automated metadata management, data observability, edge computing, federated learning, and cloud-native analytics continue to reshape decision support systems. However, standardized architectural frameworks and benchmarking methodologies that integrate these technologies into unified Data Lakehouse ecosystems remain limited. Future research should therefore focus on developing interoperable, scalable, and intelligent data architectures capable of supporting both operational analytics and advanced artificial intelligence applications.

## 6. Conclusion

This study provides a comprehensive examination of modern data architectures for decision support systems, tracing the evolution from traditional Data Warehouses through Data Lakes to the emergent Data Lakehouse paradigm. The findings reveal a clear architectural trajectory driven by increasing data volumes, variety, and velocity, and the need to support both traditional business intelligence and advanced analytics within unified environments.

The Data Lakehouse architecture represents a fundamental advancement in data management for decision support, addressing the critical limitations of both Data Warehouses and Data Lakes. By integrating open table formats—including Delta Lake, Apache Iceberg, and Apache Hudi—the Lakehouse provides transactional reliability (ACID compliance), schema evolution capabilities, time travel functionality, and optimized analytical performance for file-based storage. The medallion architecture pattern (Bronze, Silver, Gold) provides a structured framework for data quality management, supporting the progressive refinement of data from raw ingestion to curated business metrics.

Empirical evidence from the insurance sector demonstrates the performance impact of data warehousing adoption, with significant reductions in loss ratios (4.8%), combined ratios (5.6%), and claim settlement times (6.2 days). The moderating effect of IT investment intensity (additional 3.2 percentage points of improvement) underscores the importance of complementary technology capabilities in realizing data architecture benefits. Applications in higher education demonstrate the versatility of Data Lake architectures in

managing diverse data types and supporting institutional decision-making.

The analysis of data ingestion frameworks reveals distinct adoption patterns, with Apache Kafka dominant for event streaming, Apache NiFi preferred for complex data flow management, and Apache AirFlow widely adopted for workflow orchestration. Organizations increasingly require hybrid architectures supporting both batch and stream processing to handle diverse data sources and velocities.

Key challenges persist, including data governance complexities, metadata management requirements, the need for specialized skills, and security and privacy concerns. The transition to modern data architectures also presents organizational challenges, including resistance to change and the need for cultural transformation toward data-driven decision-making.

The implications of this research extend across multiple stakeholder groups. For practitioners, the findings provide guidance on architectural selection, implementation strategies, and the importance of complementary IT investments. For researchers, the identified gaps offer opportunities for further investigation, including the long-term impact of Data Lakehouse adoption, the development of best practices for data governance in unified architectures, and the integration of artificial intelligence with data management platforms. For policymakers, the study highlights the need for regulatory frameworks that balance innovation with privacy and security in data-intensive sectors.

As organizations continue their digital transformation journeys, the adoption of modern data architectures will play an increasingly central role in enabling data-driven decision-making. The Data Lakehouse paradigm, supported by open table formats and structured data governance frameworks, provides a foundation for building intelligent, scalable, and cost-effective decision support systems capable of supporting both operational reporting and advanced analytics in the era of big data and artificial intelligence.

## References


Alam, M. A., Sohel, A., Uddin, M. M., & Siddiki, A. (2024). Big Data and Chronic Disease Management Through Patient Monitoring And Treatment With Data Analytics. *Academic Journal on Artificial Intelligence, Machine Learning, Data Science and Management Information Systems, 1*(01), 77-94.

Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I., & Zaharia, M. (2020). Delta Lake: High-performance ACID table storage over cloud object stores. *Proceedings of the VLDB Endowment, 13*(12), 3411-3424.

Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unifies data warehousing and advanced analytics. *Proceedings of the 11th Annual Conference on Innovative Data Systems Research (CIDR '21)*.

Blazic, G., Poscic, P., & Jaksic, D. (2017). Data warehouse architecture classification. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics*, 1491-1495.

Dabbechi, H., Haddar, N. Z., Elghazel, H., & Haddar, K. (2021). Social Media Data Integration: From Data Lake to NoSQL Data Warehouse. In *Intelligent Systems Design and Applications* (pp. 701-710). Springer.

Databricks. (2023). *What is a medallion architecture?* https://docs.databricks.com/en/lakehouse/medallion.html

Dehury, C. K., Jakovits, P., Srirama, S. N., Giotis, G., & Garg, G. (2022). TOSCAData: Modeling data pipeline applications in TOSCA. *Journal of Systems and Software, 186*, 111164.

El Aissi, M. E. M., Benjelloun, S., Loukili, Y., Lakhrissi, Y., Boushaki, A. E., Chougrad, H., & Elhaj Ben Ali, S. (2022). Data Lake Versus Data Warehouse Architecture: A Comparative Study. *Lecture Notes in Electrical Engineering, 745*, 201-210.

Elkina, H., & Zaki, T. (2023). Data Ingestion Frameworks for Data Lakes: An Overview. *Journal of Research Technology and Digital Development, 6*(10s2), 1700-1708.

Fang, H. (2015). Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, 820-824.

Giebler, C., Gröger, C., Hoos, E., Eichler, R., Schwarz, H., & Mitschang, B. (2021). The Data Lake Architecture Framework. *Gesellschaft für Informatik, Bonn.*

Harasis, A. (2026). Impact of data warehousing adoption on underwriting and claims performance in Saudi insurance firms. *Insurance Markets and Companies, 17*(1), 65-77.

Hassan, M. S., Islam, M. A., Abdullah, A. B. M., & Nasir, H. (2024). End-user perspectives on fintech services adoption in the Bangladesh insurance industry. *Journal of Financial Services Marketing, 29*(4), 1377-1395.

Herden, O. (2020). Architectural Patterns for Integrating Data Lakes into Data Warehouse Architectures. *Lecture Notes in Computer Science, 12581*, 12-27.

Holub, B. L., & Shevchenko, D. V. (2026). Modern Design Paradigms for Decision Support Systems: From Integrating Data Warehouses and Data Lakes to the Data Lakehouse Architecture. *Information Technologies and Electronic Engineering, 1*(1), 28-38.

Inmon, W. H., & Linstedt, D. (2019). *Data architecture: A primer for the data scientist* (2nd ed.). Morgan Kaufmann.

Ionescu, S.-A., Diaconita, V., & Radu, A.-O. (2025). Engineering sustainable data architectures for modern financial institutions. *Electronics, 14*(8), 1650.

Martín, C., Langendoerfer, P., Zarrin, P. S., Díaz, M., & Rubio, B. (2022). Kafka-ML: Connecting the data stream with ML/AI frameworks. *Future Generation Computer Systems, 126*, 15-33.

Miloslavskaya, N., & Tolstoy, A. (2016). Big Data, Fast Data and Data Lake Concepts. *Procedia Computer Science, 88*, 300-305.

Morshed, A. (2025a). Cultural norms and ethical challenges in MENA accounting: The role of leadership and organizational climate. *International Journal of Ethics and Systems, 41*(3), 630-656.

Morshed, A. (2025d). Sustainable energy revolution: Green finance as the key to the Arab Gulf States' future. *International Journal of Energy Sector Management, 20*(2), 556-577.

Mukherjee, R., & Kar, P. (2017). A Comparative Review of Data Warehousing ETL Tools with New Trends and Industry Insight. *2017 IEEE 7th International Advance Computing Conference*, 943-948.

Poojara, S. R., Dehury, C. K., Jakovits, P., & Srirama, S. N. (2022). Serverless data pipeline approaches for IoT data in fog and cloud computing. *Future Generation Computer Systems, 130*, 91-105.

Ravat, F., & Zhao, Y. (2019). Data Lakes: Trends and Perspectives. In *Database and Expert Systems Applications* (pp. 304-313). Springer.

Reis, J., & Housley, M. (2022). *Fundamentals of data engineering: Plan and build robust data systems* (1st ed.). O'Reilly Media.

Saggar, S., Bitoni, C., Khurana, I., & Alhawat, R. (2022). Data Warehouse with Big Data Technology for Higher Education. *SSRN Electronic Journal*.

Salaki, R. J., & Ratnam, K. A. (2018). Agile Analytics: Applying in the Development of Data Warehouse for Business Intelligence System in Higher Education. In *Trends and Advances in Information Systems and Technologies* (pp. 1038-1048). Springer.

Santoso, L. W., & Yulia. (2017). Data Warehouse with Big Data Technology for Higher Education. *Procedia Computer Science, 124*, 93-99.

Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems, 56*(1), 97-120.

Seshagiri, S., & Prema, K. V. (2025). Efficient handling of data imbalance in health insurance fraud detection using meta-reinforcement learning. *IEEE Access*.

Singh, P. (2019). Airflow. In *Learn PySpark* (pp. 67-84). Apress.

Smolinski, M. (2018). Impact of Storage Space Configuration on Transaction Processing Performance for Relational Database in PostgreSQL. In *Beyond Databases, Architectures and Structures* (pp. 157-167). Springer.

Sreemathy, J., Joseph V., I., Nisha, S., Prabha I., C., & Priya R.M., G. (2020). Data Integration in ETL Using TALEND. *2020 6th International Conference on Advanced Computing and Communication Systems*, 1444-1448.

Sukare, N., & Al, E. (2021). Smart Classroom Environment using IoT in advanced and lebanese French university Education. *Turkish Journal of Computer and Mathematics Education, 12*(7).

Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems*. Springer.

Williamson, B. (2018). The hidden architecture of higher education: Building a big data infrastructure for the 'smarter university.' *International Journal of Educational Technology in Higher Education, 15*(1), 12.

Wyatt, L., Caufield, B., & Pol, D. (2009). Principles for an ETL Benchmark. In *Performance Evaluation and Benchmarking* (pp. 183-198). Springer.