# State of the Art: The Evolution of Cloud Systems and Data Architectures

**Thesis Phase 1: Knowledge and Information Gathering**

# 1. Introduction

The last decade has witnessed a fundamental paradigm shift in how organizations manage, store, and utilize data. We have moved from an era of on-premise, monolithic systems, characterized by rigid schemas and capital-intensive hardware, to a cloud-native ecosystem defined by elasticity, distributed computing, and service-oriented architectures. This transition was not merely technological but represented a fundamental change in the operational and economic models of IT infrastructure.

The early 2010s were dominated by the challenge of "Volume," addressed by the Hadoop ecosystem and the initial concept of the Data Lake. However, as cloud computing matured, the focus shifted from simply storing vast amounts of data to extracting value from it efficiently. This gave rise to the "Modern Data Stack," decoupling compute from storage, and eventually led to the current socio-technical debates regarding centralization versus decentralization (Data Mesh).

This literature review analyzes the trajectory of this evolution. By synthesizing recent academic findings, specifically drawing from systematic reviews on Data Mesh, Lakehouse architectures, and Cloud Security, this chapter establishes the theoretical framework necessary to understand the modern data landscape. Furthermore, it highlights the persistent friction points, specifically security governance, financial accountability (FinOps), and vendor lock-in, that continue to challenge organizations despite technological advancements.

# 2. The Evolution of Data Architectures

The trajectory of data architecture can be categorized into three distinct eras, each emerging as a reaction to the limitations of its predecessor.

## 2.1 The Era of the Data Lake (2010–2015)

Before the dominance of the cloud, the Enterprise Data Warehouse (EDW) was the standard. EDWs relied on "Schema-on-Write," requiring data to be modeled before ingestion. While highly structured, this approach failed to keep pace with the explosion of unstructured data (logs, media, sensor data) driven by Industry 4.0.

The industry responded with the **Data Lake**. Built largely on the Hadoop Distributed File System (HDFS) and utilizing the MapReduce programming model, the Data Lake introduced the

concept of **"Schema-on-Read"**. This allowed organizations to ingest raw data in its native format and apply structure only when the data was queried.

However, the literature highlights a critical failure mode of this era: the **"Data Swamp"**. Without the rigid governance of the warehouse, Data Lakes often became dumping grounds for uncatalogued, low-quality data. While Data Lakes solved the problem of storage cost and volume, they struggled with data discovery, reliability, and the complexity of maintaining on-premise Hadoop clusters.

## 2.2 The Modern Data Stack and the Lakehouse (2015–2020)

The mid-2010s marked the transition to Cloud-Native architectures. The defining technological breakthrough of this period was the **separation of Compute and Storage**. Cloud providers (AWS, Azure, Google Cloud) allowed organizations to store infinite data cheaply (e.g., in Amazon S3) while spinning up transient compute clusters only when processing was required.

This era saw the rise of the **Data Lakehouse**. The Lakehouse architecture sought to unify the best features of the Data Warehouse (reliability, performance) with the flexibility of the Data Lake.

- **Transactional Capabilities:** Unlike early Data Lakes, Lakehouses introduced ACID (Atomicity, Consistency, Isolation, Durability) transactions to object storage.
- **Unified Workloads:** They enabled BI (Business Intelligence) and AI/ML workloads to operate on the same data copy, eliminating the need to maintain separate silos for reporting and data science.

Despite these technical leaps, the organizational bottleneck remained. The architecture was still fundamentally centralized: a single data engineering team was responsible for ingesting and cleaning data for the entire organization, leading to scaling issues as data variety increased.

## 2.3 Decentralization and Data Mesh (2020–Present)

The most recent evolutionary step represents a shift from a purely technical architecture to a **socio-technical** one. The centralization of data teams became a bottleneck that slowed down domain-specific innovation.

The solution proposed is the **Data Mesh**. Data Mesh is not a specific technology but a paradigm shift based on four foundational principles:

1. **Domain-Oriented Decentralization:** Ownership of data is pushed to the business domains (e.g., Marketing, Sales, Logistics) that understand the data best, rather than a central IT team.
2. **Data as a Product:** Domains must treat their data assets as products with consumers, ensuring high quality, documentation, and usability.
3. **Self-Serve Data Infrastructure:** To prevent duplication of effort, a central platform team

provides the tooling (infrastructure-as-code) that allows domains to spin up their own data products easily.
4. **Federated Computational Governance:** Global standards (security, interoperability) are applied automatically across the mesh to ensure that decentralization does not lead to chaos.

An extension of this decentralization is the concept of **Dataspaces**. A data space is a digital environment facilitating secure data sharing between organizations (not just within them). Unlike a Data Mesh which focuses on intra-organizational decentralization, Dataspaces provide the infrastructure and governance for *inter-organizational data ecosystems*. This allows for controlled access where data remains with the source but can be consumed by external partners under strict usage contracts.

While Data Mesh offers a path to organizational scalability, the literature suggests it introduces significant complexity regarding interoperability and governance, which brings us to the critical challenges facing modern cloud systems.

## 2.4 Beyond the Modern Data Stack (Future)

Recent innovations suggest a move beyond the standard Lakehouse model towards even more specialized and *AI-integrated architectures*.

- **AI-Native Warehouses and Vector Search:** Traditional warehouses treat data as static rows. However, the rise of Generative AI has necessitated "AI-Native" capabilities. Modern systems are embedding ML primitives (like k-means clustering or regression models) directly into the database engine. Furthermore, platforms like Google BigQuery are integrating **Vector Search** capabilities, enabling semantic search and RAG (Retrieval-Augmented Generation) directly on the data warehouse without moving data to a separate vector database.
- **Disaggregated Executions:** While compute and storage are already separated, the trend is moving towards "extreme disaggregation." New architectures describe systems where query execution is broken down into fine-grained tasks that access shared storage asynchronously and in parallel, allowing for even greater elasticity and resource utilization than current serverless models.
- **Specialized AI Compute Frameworks:** As AI workloads grow, the processing framework itself is evolving. Apache Spark has long been the standard for data processing. However, Ray is emerging as a preferred "AI Compute Engine". While Spark is optimized for synchronous and bulk-synchronous parallel (BSP) data tasks, Ray provides a more flexible, actor-based, asynchronous runtime better suited for the complex, heterogeneous patterns found in distributed AI training and serving.

# 3. Critical Challenges in Modern Cloud Systems

While the transition to cloud and decentralized architectures has solved issues of scalability and

hardware maintenance, it has introduced new classes of problems. The literature identifies three primary "friction points": Security, Cost, and Vendor Lock-in.

## 3.1 Security and Governance in a Fragmented Landscape

The shift from a centralized "fortress" (the Data Warehouse) to a decentralized Mesh creates a massive surface area for security vulnerabilities, with an inherent tension between the democratization of data and the enforcement of security protocols.

In a centralized model, a single gatekeeper controls access. In a decentralized model, access control policies must be propagated across hundreds of data products and microservices. The literature points to **Federated Learning** and policy-as-code as emerging solutions, where governance rules are embedded programmatically into the infrastructure. However, the challenge remains: how to ensure regulatory compliance (GDPR, HIPAA) when data ownership is distributed across independent teams.

## 3.2 Economic Pressures: The Rise of FinOps

One of the most disruptive aspects of cloud adoption is the shift in financial models. Organizations have moved from **CapEx** (Capital Expenditure, buying servers every 5 years) to **OpEx** (Operational Expenditure, paying monthly for usage).

While this offers agility, it often leads to "Cloud Bill Shock". The variable nature of cloud costs means that a poorly written query or an orphaned cluster can result in massive unexpected expenses. The literature argues that traditional procurement processes are ill-equipped for this dynamic. This has necessitated the rise of **FinOps (Financial Operations)**, a cultural practice where engineering teams take ownership of their cloud costs. The challenge is no longer just "can we build it?" but "what is the unit cost of this workload?".

## 3.3 Vendor Lock-in and Interoperability

As organizations rely more heavily on proprietary cloud services (e.g., AWS Redshift, Google BigQuery, Azure Synapse), they face the risk of Vendor Lock-in. While multi-cloud strategies are theoretically attractive for redundancy and negotiation power, they are technically difficult to implement due to data gravity and egress fees.

To mitigate this, the industry is increasingly turning toward **Open Table Formats** (such as Apache Iceberg, Delta Lake, or Apache Hudi). These formats allow the data to be stored in open standards, making it readable by different compute engines across different clouds. High-performance querying on formats like Iceberg was historically difficult due to metadata overhead, but modern query engines are now optimizing for these formats with intelligent *caching* and *metadata pruning*. This promises a future where data storage is truly commoditized and decoupled from the compute vendor, though the literature indicates that true interoperability remains a significant hurdle, particularly for smaller organizations that lack the resources to

manage complex multi-cloud abstractions.

# 4. Conclusion

The evolution of cloud systems and data platforms has been characterized by a pendulum swing: from the rigid centralization of the Data Warehouse to the chaotic freedom of the early Data Lake, and finally to the structured decentralization of the Data Mesh.

The state of the art, as defined by the reviewed literature, places us in the era of the **Lakehouse** (technologically) and the **Data Mesh** (organizationally). We have successfully solved the problems of storage volume and compute scaling.

However, significant gaps remain. The research indicates that while the architecture is mature, the operational disciplines are lagging. Specifically:

1. **Governance approaches** have not fully caught up to decentralized architectures, leaving security gaps.
2. **Financial frameworks (FinOps)** are often reactive rather than proactive.
3. **Interoperability** is hindered by proprietary cloud incentives, creating vendor lock-in risks.

These identified gaps represent the critical challenges that modern organizations face. The subsequent phase of this thesis will utilize a questionnaire to validate whether these academic observations correlate with the practical realities faced by industry professionals today.

# 5. References

**Data Mesh: A Systematic Gray Literature Review** (2023)
**From Data Warehouse to Lakehouse: A Comparative Review** (2022)
**Challenges and Opportunities in Big Data Analytics for Industry 4.0** (2025)
**Navigating the Data Architecture Landscape** (2023)
**Financial Cloud Cost Optimization: A FinOps Framework** (2024)
**Systematic Review on Cloud Computing Security** (2024)
**Evaluation of Multi-Cloud Strategies in Small Scale Organizations** (2025)
**Toward Data Lakes as Central Building Blocks** (2022)
**Visions: Everything you need to know about data spaces** (2024)
**Firebolt: Querying Apache Iceberg with Sub-Second Performance** (2025)
**US Patent US11954112B2: Systems and methods for data processing and enterprise AI applications** (2024)
**US Patent US12271375B2: Disaggregated query processing utilizing precise, parallel, asynchronous shared storage repository access** (2025)
**Ray: The AI Compute Engine**
**Google Cloud: Search embeddings with vector search** (2026)