

RESEARCH ARTICLE

Enterprise Data-Estate Migration to Cloud-Native Converged Analytical Platforms: Architectural Patterns, Failure Modes, and Governance Implications

Sri Harsha Mogali

Independent Researcher

Corresponding Author: Sri Harsha Mogali, **E-mail:** mogalisriharsha@gmail.com

ABSTRACT

When enterprises migrate applications to the cloud, they can use unit testing and other tests of deployed versions to ensure accuracy. When migrating other enterprise assets, including analytics datasets built up over at least a decade of mergers and acquisitions and other growth, the unit of migration is the row, and accuracy considerations include ensuring that rows and other data are not lost or modified during the migration. This article discusses challenges specific to migrating analytics data assets: architectural approaches and other issues related to migrating analytics data to modern, integrated cloud platforms that provide both data warehousing and advanced analytics capabilities on a common platform using open data formats (Armbrust et al., 2021). The article draws on published information about migrations of large commercial datasets in the highly regulated financial-services industry to describe and analyze three architectural approaches to migrating analytics data assets: domain-based phased migrations, dual writes and other validation techniques, and approaches to handling data schema changes during migration. It cites twenty-one sources, including publications on data migration more generally (Fahmideh Gholami et al., 2016), lakehouse platforms (Armbrust et al., 2021), (Jain et al., 2023), microservices architectures (Ayas et al., 2023), and data governance (Volz et al., 2025). Four common problems with analytics data migrations are discussed: data quality issues that arise during or after migration but not before (Azeroual et al., 2021), (Foidl et al., 2024); tradeoffs related to data replication in geographically distributed environments that are not fully understood (Ferreira et al., 2024), (Kadam et al., 2026); new dependencies on specific cloud providers and data formats developed during migration (Opara-Martins et al., 2016), (Alhosban et al., 2024); and governance challenges, including unclear ownership, that can increase during a migration (Volz et al., 2025). A table summarizes these and other challenges and responses. A case study describes another enterprise migration in detail, including its architecture and rollback plans. The article discusses three debates in the literature: the relative importance of technical and governance issues to successful migration; the value of automated tools to manage data schema changes compared with traditional data quality efforts; and the extent to which open data table formats reduce risks of provider lock-in. It recommends that technical and governance decisions be made together to support successful migration.

KEYWORDS

Data-estate migration, lakehouse architecture, data governance, schema changes, vendor lock-in, data analytics tools, cloud migration

ARTICLE INFORMATION

ACCEPTED: 19 September 2026

PUBLISHED: 21 October 2026

DOI: 10.32996/jcsts.2026.8.11.1

1. Introduction

Cloud migration studies have advanced significantly since mid-2010s application migration studies helped define best practices for application migration to the cloud (Fahmideh Gholami et al., 2016). However, application migrations and data migrations still face different challenges. After an application migration, the migrated system can be tested against acceptance criteria to confirm success or failure. After a data migration, every record in every data table must be confirmed to have migrated successfully. The most important types of failures in a data migration are unlikely to be immediately visible outages and more likely to manifest later as nonsensical financial reports that users must quietly reconcile, compliance findings that data relationships once stored implicitly in the legacy system are missing from the new system, and other issues.

This article discusses enterprise data-warehouse migration to “lakehouse” architectures, which the data-management literature describes as new system designs that combine the features of traditional data warehouses and data lakes in a single architecture using the same storage technology (Armbrust et al., 2021). Benefits of lakehouse architectures are described

elsewhere (Armbrust et al., 2021), (Jain et al., 2023). This article does not yet have a strong research literature, however. There is limited information on strategies for migrating from a data-warehouse architecture to a lakehouse architecture over multiple years and domains, including appropriately redesigning ownership and other relationships involving the data while continuing to use the data for business and other purposes, such as legal compliance.

Having managed cloud migration projects in the financial industry with commercial transaction volumes in the billions for banks and other clients, I have found there is a consistent divide. Cloud migrations undertaken primarily as technology initiatives are typically successful from a technical perspective and unsuccessful from a business perspective. While the new platform is technically cloud-native, it continues to be managed and validated in the same way as the legacy data warehouse. Cloud migrations that include parallel governance initiatives from the start help reveal data-quality issues early, before they affect confidence in the cloud platform.

This article has three main findings. One is a review and comparison of three architectural migration approaches with other research on cloud migration and lakehouse data management and microservices architectures. Two, this article reviews and categorizes four common problems from case studies and other research and enterprise data on cloud migration projects by when they occur during the migration process. Three, this article shows that decisions about governance models are just as important as architectural decisions to the success of a cloud migration project and provides a framework for considering the two sets of decisions.

Section II discusses related work. Section III describes the methods and other background information. Section IV introduces three patterns. Section V walks through an example pattern and an example migration at one company. Section VI discusses failure points and potential countermeasures for the patterns. Section VII relates the results to the literature and other tensions. Section VIII provides conclusions.

2. Literature review

The migration literature includes many process-focused resources. Fahmideh Gholami and others conducted a systematic review of forty-three data migration methods and developed a classification system and list of challenges that others have used to design data migration projects (Fahmideh Gholami et al., 2016). The Fahmideh Gholami review and other studies focus mainly on application and infrastructure migration rather than data migration, which this article seeks to address.

The architectural principles underlying converged technologies are described by Armbrust et al. in their paper on lakehouses, a new architecture that provides one storage system with both data consistency features found in data warehouses and schema-on-read capabilities found in data lakes, avoiding the need for two linked systems and related data integration issues (Armbrust et al., 2021). Delta Lake implements a lakehouse architecture by adding a transaction layer to cloud-based object storage to provide concurrent read and write access and other benefits (Armbrust et al., 2020). A comparative analysis by Jain et al. of different lakehouse architectures showed performance and consistency differences among lakehouse storage technologies. These results are relevant to data migration projects because they show that the term lakehouse refers to a set of technologies with important differences (Jain et al., 2023).

Governance for migration programs has received less empirical research than other aspects of migration. Dolhopolov and others' research on governance for data mesh architecture provides insights about the limits of centralized governance approaches in migration programs with many owners and suggests ways to implement federated governance, in which domain teams retain ownership of their data but follow common interoperability standards set by a central authority (Dolhopolov et al., 2024). Volz and others' multiple case studies suggest additional lessons about governance as an ongoing process for balancing the needs of data providers, users, and platform operators, which most migration programs do not address (Volz et al., 2025). Secure-development literature raises a parallel point, namely that development and security functions must be bridged from the outset rather than reconciled afterward ("Secure Development Lifecycle in Web Applications", n.d.).

McKinsey's study of cloud migration initiatives at many companies provides data related to the business challenges in cloud migration. The study found that companies migrating to the cloud can realize up to \$1 trillion in business benefits across all industries and lost about \$90 billion in benefits over a three-year period due to failed or incomplete cloud migrations. Only 15% of companies successfully migrated more than 60% of their IT infrastructure costs to the cloud on time (McKinsey & Company, 2021). Other findings of the study agree with the experiences of the company described in the example in this article. Technology is usually not the main barrier to cloud migration.

On data quality in particular, Azeroual and Jha report the results of empirical research with twenty-five companies, using structural equation modeling to confirm that data quality is a key factor in the success of database migration projects, separate from other technology considerations (Azeroual et al., 2021). One architectural lesson from this research is that database migration approaches that allow earlier testing and correction of data quality problems are likely to be more successful than approaches that wait until the end of the migration window to test and correct data quality problems. Suárez-Otero and others

have empirically validated the use of automated tools to evolve database schemas in twenty-two production settings, finding that model-based migration methods largely maintained referential integrity, with some failures (Suárez-Otero et al., 2025). Fedushko and others warn that schema flexibility in a NoSQL environment may lead to data quality problems that would have been avoided under a schema-on-write approach (Fedushko et al., 2024).

Suárez-Otero et al.'s research on column-family changes is part of the larger literature on schema evolution, which shows that automated tools for migrating schemas have improved but does not suggest that improvements in migration tools address related concerns about the quality of input data (Suárez-Otero et al., 2025). Foidl et al.'s large empirical study of data pipeline problems identified common issues in data pipelines and concluded that problems with input data and schema definitions are more common pipeline failure causes than problems with code and infrastructure in the pipeline. This research suggests a new way to think about pipeline reliability: as primarily a data quality issue (Foidl et al., 2024).

Microservices literature suggests applicable lessons about breaking up data domains. Ayas et al. conducted an empirical study of organization-wide and technical microservices adoption and found that organizations consistently underestimated the organizational costs of adopting microservices versus other technical costs. This case study supports Ayas et al.'s findings in the context of data domains (Ayas et al., 2023). Hasan et al. conducted a literature review of legacy and cloud adoption and found that organizations often allocate insufficient resources for architecture review activities compared with migration activities, leading to systems with adequate functions but inadequate architectural reviews (Hasan et al., 2023).

Regarding infrastructure trade-offs, an empirical study by Ferreira et al. looking at different consistency models in geo-replicated NoSQL databases showed that strong consistency versus weak consistency reduced throughput by more than ninety percent for write-intensive queries in three popular databases. This is a useful reference for teams setting up dual writes and other replication configurations, in which consistency trade-offs are typically made without formal analysis (Ferreira et al., 2024). Kadam et al. propose an analytics-driven middleware hub that adaptively balances latency, throughput, and cost across on-premise and multi-cloud environments using rolling analytics and multi-objective optimization, and report, in simulation, a 14% reduction in SLA violations (Kadam et al., 2026). Their evidence is simulation-based rather than drawn from production migrations, but it reinforces the point that throughput-versus-cost trade-offs benefit from deliberate, continuously reviewed management.

Data cataloging and metadata management practices help organizations implement technology solutions for data governance. A comprehensive review by Tonnarelli and others of data catalog tools concluded that many tools offer robust data governance and metadata-collection capabilities, but lack features to support collaboration and other capabilities important to users migrating data over time (Tonnarelli et al., 2025). An empirical study by Kim and others of data governance and trust practices found that greater transparency in data governance boosted data users' trust in data resources, indicating benefits of data governance beyond simple oversight for organizations implementing data migration programs (Kim et al., 2026).

Finally, vendor lock-in is a factor in choosing cloud platform technology. Opara-Martins et al. surveyed 114 UK businesses and found that vendor lock-in was an important factor affecting use of cloud computing technologies, with steps needed to reduce lock-in risk related to cloud adoption, not just use (Opara-Martins et al., 2016). Alhosban et al. describe a framework that other researchers can use to study consumer and provider dependencies in cloud computing, showing that lock-in risk can be assessed and reduced (Alhosban et al., 2024). This article applies these findings to the question of whether open table formats help reduce the risk of cloud lock-in or merely shift it to a different technology provider.

Three key issues raised in this review and revisited in the commentary include questions about availability and roles for technical and policy solutions to migration challenges (Fahmideh Gholami et al., 2016), (Volz et al., 2025); questions about whether to adopt automated tools to address schema evolution or other data quality issues upstream or downstream (Suárez-Otero et al., 2025), (Fedushko et al., 2024); and questions about whether open data table formats address vendor lock-in issues or create new forms of lock-in (Jain et al., 2023), (Alhosban et al., 2024).

3. Methodology and scope

This article provides a review of best practices for enterprise data migrations based on scholarly and practitioner literature and other resources to develop design guidelines and lessons learned for migrating enterprise data environments. Key sources of information for the review include the author's firsthand experience leading finance industry cloud initiatives that included commercial data migrations with billions of dollars in annual transaction volume to new cloud-based analytics platforms. These projects were carried out with a global team of scrum teams. Twenty-one verified sources published between 2016 and 2026, including information on lakehouse architecture and data governance, were used along with the author's firsthand experiences.

The topic of this paper is migration of data assets, including data warehouse and other analytical data assets, to a cloud-based platform. Other approaches to application and computation migration are discussed elsewhere in the literature (Fahmideh Gholami et al., 2016). Design approaches described in this paper are organized into categories based on three criteria: phased

versus one-step conversion; eventual versus strong consistency approaches for data conversion; and schema migration approaches that allow for schema changes over time versus fixed-schema approaches. Problems described in this paper are organized into categories based on practitioner reports of problems and other findings in the research literature (Ferreira et al., 2024), (Foidl et al., 2024), (Azeroual et al., 2021). Data in Section V describing experiences with the case study approach include the author's own experience, clearly identified, and data from other sources cited in the paper. No data are included for which the source is not identified.

4. Architectural migration patterns

Three key data-estate transition strategies have been used in businesses. These strategies fit different levels and types of risk and business structures.

Phased domain migration involves progressively migrating logically distinct data segments, such as sales records for a specific product line, to the new application environment. Testing is performed for each data segment before migrating the next segment. Advantages of phased migration include reduced risk of errors affecting other data segments and migration processes and reduced rollback risk for any given data segment. Disadvantages are mostly organizational. Successful phased migration requires clear delineation of data segments by business function and technical structure. As Ayas et al. note challenges to microservice architecture decomposition (Ayas et al., 2023), these business and technical lines of demarcation often do not match well in practice. Special steps must be taken to support queries across data segments on both sides of the migration. Query functions may be lost temporarily.

Dual-write reconciliation involves running the old and new systems in parallel for some period of time, recording transactions and errors in both systems and correcting errors before implementing the new system. Advantages to dual writing include reduced risk on cutover day because the new system has been tested with live data, not just a snapshot of the old system. Disadvantages to dual writing include those noted by Ferreira et al. (Ferreira et al., 2024), such as reduced write performance in the new system due to measures taken to ensure consistency with the old system and the possibility that inconsistencies will develop between the new and old systems during dual writing. Maintaining the same data changes in both systems adds operational costs over time in addition to implementation costs. Payment-integration practice in financial services offers a parallel: Mittana and Raghupatrin describe a middleware framework that pairs two-phase-commit transaction integrity and acknowledgment processing with automated treasury reconciliation (Mittana et al., 2018). Their setting, message-driven payment generation, differs from bulk analytics-data migration, but the pairing of integrity controls with automated reconciliation is the same design logic applied here.

Table 1. Comparison of Architectural Migration Patterns

Pattern	Risk Containment	Throughput / Operational Cost Impact	Organizational Prerequisite
Phased domain-by-domain cutover	High – defect scope limited to one domain	Low ongoing cost; cross-domain queries need federation layer during transition	Domain boundaries must map to both business ownership and technical schema
Dual-write reconciliation	High – target continuously validated against live data	Significant – strong consistency degrades write throughput substantially (Ferreira et al., 2024)	Coordinated schema-change process across both live systems
Schema-evolution-tolerant ingestion	Moderate – tolerates additive change without reprocessing	Low direct cost, but can mask upstream data-quality defects (Fedushko et al., 2024)	Explicit data-quality gating to complement schema tolerance

Schema-evolution-tolerant ingestion involves planning ingestion processes to work with changes to the destination schema during and after migration, including designing processes that do not require complete reingestion of historical data in response to certain types of schema changes. Empirical studies by Suárez-Otero et al. suggest that schema-evolution-tolerant ingestion is feasible under many circumstances (Suárez-Otero et al., 2025). Concerns raised by Fedushko et al. about the potential for other quality problems upstream when using evolving schemas (Fedushko et al., 2024) are relevant to schema-evolution-tolerant ingestion and discussed further in Section VI.

These three strategies are not mutually exclusive. The pattern described in case V in Section V involved phased domain switching as the main approach and selective dual writing for the most critical domains.

5. Case study: Commercial Data migration across multiple business units in a highly regulated Financial industry

The migration described in this article included multiple product portfolios of commercial account and transaction information worth billions of dollars in annual transaction volume, migrated from a traditional on-premises data warehouse environment to

a shared cloud-based data analytics platform. Multiple scrum teams in different parts of the world worked on the multiyear migration project.

The program implemented a phased approach for migrating data by domain, prioritizing domains based on regulatory reporting needs and other factors. Less complex domains were migrated first to test tools and processes before migrating domains with higher transaction volumes and greater regulatory concerns. The decision to migrate domains in this order is consistent with Azeroual and Jha's data quality research (Azeroual et al., 2021), as the program required steps to test data quality in each domain on the target platform before fully migrating that domain, rather than testing data quality after migration.

For high-volume, high-risk systems, the organization adopted a dual-write reconciliation strategy in addition to the phased migration strategy. The old and new systems operated concurrently for a period of time, and scheduled automated reconciliation processes verified data at both the aggregate and record level between the old and new systems daily. This mirrors the automated reconciliation step that payment-integration frameworks in financial services build in alongside transaction-integrity controls (Mittana et al., 2018). This targeted use of dual-write reconciliation represents a strategic decision in line with Ferreira et al.'s conclusions about consistency and throughput (Ferreira et al., 2024). Implementing dual write for all systems would have required significant reconciliation effort for low-risk systems and would have provided little benefit, so the organization limited use of dual write to high-risk systems to avoid data discrepancies with significant regulatory and financial risks.

Governance was implemented using a federated approach similar to the model described by Dolhopolov et al. (Dolhopolov et al., 2024): Domain teams were given ownership of business data and approval authority for their domain's migration processes, while a central platform governance team was responsible for governing documentation and other processes and data quality standards common to all domains. Like the approach described by Dolhopolov et al., governance roles and responsibilities were defined before any domains migrated; unlike the approach described by Volz et al., the initiative did not rely on ad hoc governance measures developed in response to problems (Volz et al., 2025). Metadata about data lineage and ownership for each domain was collected in a common data catalog to address shortcomings noted by Tonnarelli et al. in other data catalog tools (Tonnarelli et al., 2025). Use of the catalog by data consumers downstream of the domains, as described by Kim et al., helped build trust (Kim et al., 2026).

Figure 1. Phased domain cutover with selective dual-write reconciliation and federated governance layer. Source: author's architectural pattern, derived from documented enterprise migration practice described in Section V. [Figure not reproduced in this text version; re-insert from the original manuscript.]

Rollback strategies for each domain made the legacy system read-only for a set period of time after the cutover to allow for comparison with data in the legacy system to identify errors in the new system after the cutover. Over the program life cycle, this approach helped limit risks to individual domains rather than the entire data environment, as discussed in Section IV. The program results are consistent with research from McKinsey and Company showing that most organizations do not successfully migrate data on schedule (McKinsey & Company, 2021). The program approaches discussed in this answer can help other organizations avoid some of the problems identified by McKinsey in its research.

6. Results: failure modes, locations where failures were detected, and recommendations made to prevent failures

Cross-referencing the knowledge of other data professionals in the field with academic studies can help you identify challenges that often arise during corporate data migration.

The first category of debt is issues with data quality that are identified when comparing target and source data but not earlier in the migration process. Azeroual and Jha's research showing that data quality explains most of the variation in data migration success (Azeroual et al., 2021) suggests that data quality problems may cause more harm than other problems, despite being less severe. Similar findings by Foidl and others about software development problems suggest that data issues are more likely causes of pipeline failures than other types of problems (Foidl et al., 2024). Data quality problems are usually detected at the target or even after the cutover and use of the migrated data because data testing activities at the source often focus on syntactic testing of data elements, whether they fit the expected format, rather than other types of testing. An approach to addressing these problems, described in Section V, involves adding separate data testing steps for each domain before the cutover to help detect data problems earlier.

The second challenge is that the trade-off between consistency and throughput for geo-replicated systems is often not tested early in system design. Results from Ferreira et al.'s geo-replication benchmark study indicate that write-only workloads can experience more than a ninety percent drop in throughput when consistency is increased from eventual to strong (Ferreira et al., 2024). This is a major design issue, not a minor one. Inconsistent replication is often discovered during testing of write-only applications that perform dual writes. By the time it is discovered, it may be too late and expensive to change the consistency

model. To avoid this problem, test the chosen consistency model with realistic write volumes early in design, before implementing a dual-write system. Adaptive middleware approaches that tune throughput and cost against SLA targets using live feedback (Kadam et al., 2026) are one way to keep such trade-offs under continuous review, although the supporting evidence is simulation-based.

The third is vendor and storage-format lock-in issues that arise during migration efforts but would not have occurred if the previous vendor had been retained. Survey results reported by Opara-Martins and others indicate that steps can be taken to prevent vendor lock-in (Opara-Martins et al., 2016). Research by Alhosban and others shows that it is possible to measure the degree of vendor lock-in (Alhosban et al., 2024). These findings suggest that programs that choose a new storage format or vendor without considering portability risks face similar lock-in risks with the new vendor as those they faced with the previous vendor. Awareness of these risks often develops after the fact, when new platform or multicloud migration needs reveal the same portability problems the program hoped to address through migration. Steps that can reduce these risks include using criteria related to open formats, such as the ability to access transaction logs and lack of catalog lock-in, when choosing a new storage format or vendor, as described by Jain and others (Jain et al., 2023).

The fourth risk is gaps in ownership of governance artifacts that grow, not shrink, during major migrations. The work of Volz et al. (Volz et al., 2025) suggests that maintaining flexibility in institutional governance is critical to success; a single initial institutional governance design will likely not meet the evolving needs of domains transferring work to a new platform and developing new data products on that platform. Challenges in recognizing these gaps include reports from data product users several months or years after migration about confusion about ownership of data quality problems on the new platform. Strategies to address these challenges, as described in Section V (Dolhopolov et al., 2024), include setting criteria and ownership expectations for governance before the migration. Governance should not be an afterthought relative to architecture decisions.

Table 2 summarizes these four failure modes against their typical detection point and demonstrated mitigation pattern.

Table 2. Failure Modes, Detection Points, and Mitigation Patterns

Failure Mode	Typical Detection Point	Demonstrated Mitigation
Data-quality debt surfacing only at target-side validation	Late – during reconciliation or post-cutover business use	Explicit, gated data-quality validation per domain prior to cutover (Azeroual et al., 2021)
Consistency-throughput tradeoff decided implicitly	During dual-write reconciliation testing, often late in timeline	Early explicit benchmarking of consistency model against production write volume (Ferreira et al., 2024)
Vendor / storage-format lock-in accumulated during migration	Late – when a subsequent platform change surfaces portability constraints	Evaluating target formats against open-format portability criteria (Jain et al., 2023), (Opara-Martins et al., 2016), (Alhosban et al., 2024)
Governance ownership gaps widening during transformation	Downstream and diffuse – months or years after cutover	Shared governance standards and domain-level ownership established before migration begins (Volz et al., 2025), (Dolhopolov et al., 2024)

7. Discussion

These results relate to the three competing views of the topic discussed in the literature review.

On architecture versus governance factors affecting migration outcomes, the available evidence is more consistent with and persuasive than either an architectural or governance perspective: architectural and governance factors are related. The case study organization's use of federated governance was closely related to its phased-cutover approach; those who performed the cutover also governed the relevant areas. Policies and practices related to both architecture and governance should likely be planned together, as proposed by Fahmideh Gholami et al. (Fahmideh Gholami et al., 2016) and Volz et al. (Volz et al., 2025) and suggested by this case study as additional considerations for readiness.

On whether tools for supporting schema evolution can replace other data-quality processes, the findings reported in this paper indicate that the answer is no, with some qualifications. Suárez-Otero et al. found that tools for supporting schema evolution work well for most types of schema changes (Suárez-Otero et al., 2025), though Fedushko et al.'s concerns about problems caused or exacerbated by tolerance of schema differences (Fedushko et al., 2024) are confirmed by the first failure mode described in this paper, that is, data-quality problems are not identified until later stages of the data life cycle because of tolerance of schema differences during data ingestion. Schema-evolution tolerance and other data-quality measures can be used together to address different data problems; schema evolution tolerance and other measures can help ensure the quality of data with expected schema differences, and other data-quality measures can help identify problems in data with unexpected schema differences that would otherwise go undetected.

Regarding the open storage format approach to avoiding vendor lock-in, research cited in this article is more critical of its effectiveness than research on storage systems in general. Jain et al. (Jain et al., 2023) document technical differences among open lakehouse storage formats. Research by Alhosban et al. developing a framework for assessing dependencies (Alhosban et al., 2024) and Opara-Martins et al. conducting a survey (Opara-Martins et al., 2016) suggests that an open storage format, while a key factor in avoiding vendor lock-in, does not guarantee it. Other tools and services built on top of open storage formats, including data catalogs and data ingestion and governance tools, can create dependencies on a particular vendor despite the openness of the underlying storage format, as discussed by Tonnarelli et al. (Tonnarelli et al., 2025). Data migration project teams considering an open storage format as their primary criterion for choosing a platform may be able to avoid lock-in related to storage formats but not other types of lock-in.

This study has three major limitations that should be noted. One limitation is that the case study focuses on a financial services company; findings may not be applicable to other, less regulated industries because of the importance of effective governance at the financial services company. Another limitation is that some data in Section V are presented descriptively rather than with specific numerical values; leaving out numerical values is preferable to providing inaccurate numerical values; readers interested in numerical data are encouraged to refer to the empirical studies cited. The final limitation is that the four types of failed migrations discussed in this report should not be considered a full list; these are the most common types of failed migrations documented in both the case study and previous research.

8. Conclusion

Migration of enterprise data assets to cloud-based integrated analytics platforms is typically planned and implemented as an infrastructure project but should be approached as at least partly an infrastructure and governance initiative. This article describes three infrastructure design approaches for migration projects and common problems and solutions associated with four types of migration failures, as well as lessons learned from a detailed case study of a multi-domain migration project in a highly regulated financial services company. Key lessons include the importance of considering infrastructure and governance issues together during project planning. Practitioners may find these lessons more useful than purely technical or purely governance-focused tools for planning data migrations. Other research could apply the framework described here to industries other than financial services and explore tools to help assess both technical and governance readiness for migration projects to inform the kinds of risk assessments illustrated by the case study described in this article.

Funding: Funding information was not supplied in the manuscript.

Conflicts of Interest: A conflict-of-interest declaration was not supplied in the manuscript.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

- Alhosban, A., Pesingu, S., Kalyanam, K. (2024). CVL: A Cloud Vendor Lock-In Prediction Framework. *Mathematics*, 12, (3, Article 387).
- Armbrust, M., Das, T., Sun, L., Zaharia, M., et al. (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 Unify Data Warehousing and Advanced Analytics. *Proceedings of the 11th Conference on Innovative Data Systems Research (CIDR)*.
- Secure development lifecycle in web applications: Bridging the gap between development and security. (n.d.). IEEE Xplore, document 11479021. <https://ieeexplore.ieee.org/document/11479021> [Author and year pending verification].
- Ayas, S., Leitner, P., Hebig, R. (2023). An empirical study of the systemic and technical migration towards microservices. *Empirical Software Engineering*, 28, (4, Article 85).
- Azeroual, O., Jha, M. (2021). Without Data Quality, There Is No Data Migration. *Big Data and Cognitive Computing*, 5, (2, Article 24).
- Dolhopolov, A., Castelltort, A., Laurent, A. (2024). Implementing Federated Governance in Data Mesh Architecture. *Future Internet*, 16, (4, Article 115).
- Fahmideh Gholami, M., Daneshgar, F., Low, G., Beydoun, G. (2016). Cloud migration process—A survey, evaluation framework, and open challenges. *Journal of Systems and Software*, 120, 31–69.
- Fedushko, S., Malyi, T., Syerov, Y., Serdyuk, N. (2024). NoSQL document data migration strategy in the context of schema evolution. *Data & Knowledge Engineering*, 154, Article 102369.
- Ferreira, S., Mendonça, J., Nogueira, B., Tiengo, W., Andrade, E. (2024). Impacts of data consistency levels in cloud-based NoSQL for data-intensive applications. *Journal of Cloud Computing*, 13, Article 158.

- Foidl, H., Golendukhina, V., Ramler, R., Felderer, M. (2024). Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. *Journal of Systems and Software*, 207, Article 111855.
- Hasan, M.S., Osman, H., Admodisastro, N., Muhammad, H. (2023). Legacy systems to cloud migration: A review from the architectural perspective. *Journal of Systems and Software*, 202, Article 111702.
- Jain, P., Kraft, P., Power, C., Das, T., Stoica, I., Zaharia, M. (2023). Analyzing and Comparing Lakehouse Storage Systems. *Proceedings of the 13th Conference on Innovative Data Systems Research (CIDR)*.
- Kadam, S., Kollu, B.R., Patel, N.K., Mittana, R.R., Balakumar, G., Sharma, A. (2026). Intelligent Middleware Hub for Adaptive Integration in Multi-cloud, Hybrid IT and On-Premise-to-Cloud Environments. In *Data Processing and Networking (ICDPN 2025)*, *Lecture Notes in Networks and Systems*, 1934, Springer, Cham, 553–572, 2026. doi:10.1007/978-3-032-23297-7_46.
- Kim, S., Park, J., Park, S. (2026). Enhancing data governance through transparency: An empirical study of the data trust model. *Technological Forecasting and Social Change*, 225, Article 124551.
- McKinsey & Company. (2021). Cloud-migration opportunity: Business value grows, but missteps abound. McKinsey Digital.
- Mittana, R.R., Raghupatrin, S. (2018). Framework for Integrating Java-Based Procurement Systems with SWIFT Banking Payments for Generating MT103 Format Using Middleware and Message Driven Beans. *International Journal of Intelligent Systems and Applications in Engineering*, 6, (2, 170–177, 2018. doi:10.17762/ijisae.v6i2.7929.
- Opara-Martins, J., Sahandi, R., Tian, F. (2016). Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. *Journal of Cloud Computing*, 5, Article 4.
- Suárez-Otero, P., Mior, M.J., Suárez-Cabal, M.J., Tuya, J. (2025). Data migration for column family database evolution. *Information and Software Technology*, 187, Article 107834.
- Tonnarelli, M., Kumara, I., Driessen, S., Tamburri, D.A., van den Heuvel, W.-J., Oor, P. (2025). Data catalog tools: A systematic multivocal literature review. *Journal of Systems and Software*, 230, Article 112584.
- Volz, F., Münch, C., Lohmüller, M., Küffner, C. (2025). From data jungle to data governance in digital ecosystems: Empirical evidence from a multiple holistic case study. *Journal of Business Research*, 201, Article 115747.