International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Methodologies for Designing Enterprise-Grade Data Pipelines for AI Agents in Regulated Industries

Shanmuka Siva Varma Chekuri*

Data Engineer, American Software Group (ASG), United States, New Jersey

Abstract

The study examines engineering methodologies for building enterprise-grade data pipelines that support AI agents under stringent regulatory constraints in finance, insurance, healthcare, and payroll domains. The research addresses the fragmentation of multi-region data flows, the heterogeneity of legacy ERP environments, and the growing load from AI workloads that depend on reproducible, ACID-compliant lakehouse architectures and feature-store-centric design. The work generalizes recent advances in Delta Lake–based reliability patterns, AI/ML-optimized lakehouses, feature stores, and AI-driven compliance automation, integrating them into a unified blueprint for multi-tenant, audit-ready pipelines. The goal of the article is to synthesize a reliability framework for financial data, a multi-tenant AI lakehouse model for payroll, and a novel multi-file validation and reconciliation pattern for high-risk financial ETL. Comparative analysis, source criticism, and architectural synthesis are applied to a curated set of recent scientific and professional publications. The conclusions describe how these patterns reduce reconciliation effort, strengthen regulatory assurance, and create AI-ready data foundations. The article targets data engineers, architects, and technical leaders who design AI-enabled systems in regulated industries.

Keywords: enterprise data pipelines; lakehouse architecture; ACID transactions; feature store engineering;AI agents; regulated industries; compliance automation; multi-region data governance; financial data reliability; payroll systems.

Received: 11/16/2025
Accepted: 1/16/2026
Published: 1/26/2026

----

* *Corresponding author.*

1.Introduction

Modern, regulated enterprises operate across multiple jurisdictions, combining legacy ERP platforms with cloud-native services and increasingly relying on AI agents for anomaly detection, decision support, and automation. Under such conditions, fragmented data flows, batch-only processing, lack of ACID guarantees, and opaque lineage threaten both regulatory compliance and the reliability of AI models. Financial, insurance, and payroll organizations face strict obligations for traceability, reproducibility, ty, and auditability while simultaneously pursuing near real-time analytics, AI-assisted risk management, and global consolidation of financial positions.

The objective of the article is to develop a coherent methodological basis for designing enterprise-grade data pipelines that serve AI agents, grounded in lakehouse architecture, feature-store practices, and AI-driven compliance frameworks. The first research task involves describing a unified financial data reliability architecture that combines multi-layer lineage, ACID-governed storage, deterministic ingestion, and metadata-driven validation across multi-region deployments. The second task consists of formulating a generalizable blueprint for a multi-tenant AI lakehouse that supports payroll and compliance workloads, including feature-store integration, SCD-2 dimensional modeling, and deterministic computation graphs for AI agents. The third task involves justifying a novel multi-file validation and reconciliation pattern for high-risk financial pipelines, with a focus on handling late-arriving data, ensuring cross-file consistency, and implementing conflict-safe merge strategies.

Scientific novelty arises from the integration of heterogeneous strands of recent research—on lakehouses, feature stores, continuous AI pipelines, compliance automation, and scalable ETL—into a single, research-level methodology targeted at regulated environments. The article does not introduce proprietary schemas or client-specific logic; instead, it abstracts repeatable patterns from both academic sources and large-scale production practice in multi-region ERP automation and AI-driven payroll platforms.

2.Materials and Methods

The analysis is grounded in ten recent publications covering lakehouse architectures, ETL reliability, feature stores, AI development pipelines, and AI-driven compliance governance. A.R. Aileni [1] presents an AI/ML-optimized lakehouse architecture that unifies analytical and machine-learning workloads through a consolidated storage and compute fabric tailored to modern data science. C.C. Anichukwueze, V.C. Osuji, and E.E. Oguntegbe Reference [2] describe an enterprise-wide AI-driven compliance framework for mitigating real-time cross-border data transfer risk, emphasizing architectural controls for global regulatory regimes. M. Armbrust and co-authors Reference [3] introduce Delta Lake as an open-source ACID table storage layer over cloud object stores and document its adoption for large-scale, exabyte-level data processing. S. Boosa [4] examines AI-augmented continuous delivery in regulated industries, detailing how AI-enriched DevOps pipelines maintain auditability while accelerating deployment. J. de la Rúa Martínez and colleagues [5] present the Hopsworks feature store for machine learning as a highly available platform that unifies feature pipelines with training and inference while improving throughput and reuse. D. Kaul [6] investigates AI-powered autonomous compliance management for multi-region cloud deployments, outlining an architecture for continuous, AI-driven policy enforcement and audit readiness. D. Lee, T. Wentling, S. Haines, and P. Babu [7] systematize modern Delta Lake lakehouse architectures, covering physical layout, ACID transactions, governance, and optimization strategies for production-grade systems. S. Oye [8] proposes a scalable ETL architecture based on Apache Spark and Delta Lake for big data warehouses, highlighting transactional consistency and performance for hybrid batch–streaming pipelines. M. Steidl, M. Felderer, and R. Ramler [9] conduct a multi-vocal review of continuous AI development pipelines and derive a four-stage model (data handling, model learning, software development, and system operations) for end-to-end AI delivery. A.H. Swamy [10] analyzes innovations in data lake architectures for financial enterprises, including the adoption of lakehouses, real-time processing, AI/ML integration, and governance frameworks optimized for financial institutions.

Methods. For the present article, a problem-oriented synthesis method is employed, combining a comparative analysis of architectural patterns, structural-system modeling of data flows, and a critical interpretation of reliability and compliance guarantees described in the sources. Cross-source comparison supports the alignment of concepts such as ACID table storage, feature-store orchestration, and AI-driven compliance with the practical constraints of financial and payroll data. Elements of design-oriented research are applied: the proposed unified reliability framework, multi-tenant lakehouse blueprint, and multi-file validation pattern are formulated as generalized design constructs derived from the surveyed literature. Source analysis focuses on extracting reproducible design principles rather than implementation details, which allows the results to be applied under non-disclosure constraints typical for regulated industries.

3.Results

Synthesis of the selected literature confirms that reliable AI-ready pipelines in regulated industries depend on an architectural core that combines ACID-governed lakehouse storage with deterministic ingestion and rich metadata for validation, lineage, and compliance reporting. Delta Lake is consistently highlighted as a mechanism for enforcing ACID semantics, schema evolution, and time travel on top of low-cost object storage, which directly addresses data corruption and consistency issues that arise in traditional cloud storage strategies [3]. The lakehouse blueprint, described by D. Lee and co-authors, extends these primitives into a full-stack architecture, combining medallion-style layering, transactional tables, governance capabilities, and performance optimizations suitable for enterprise data platforms [7]. In parallel, work on AI/ML-optimized lakehouses emphasizes the need to co-design storage abstractions and computation paths for large-scale model training and inference, with a focus on explicit attention to resource efficiency and ML-centric data layouts [1]. For financial enterprises, recent surveys indicate a pronounced shift from raw data lakes to governed lakehouses that support both regulatory reporting and advanced analytics [10].

In the domain of ETL and data reliability, S. Oye proposes a scalable ETL architecture that tightly couples Apache Spark with Delta Lake to achieve efficient ingestion, transformation, and storage for big data warehouses that service both batch and streaming workloads [8]. This architecture leverages Delta Lake’s transaction log for idempotent writes, schema evolution handling, and rollback capabilities, thereby reducing operational overhead and enabling predictable recovery behavior. Empirical results reported in the Delta Lake paper show that such designs reduce the number of storage-related support incidents and deliver significant performance improvements, up to orders of magnitude in specific analytical scenarios [3]. Together with the engineering patterns collected in the Delta Lake monograph [7], these findings substantiate the first pillar of the present methodology: a unified financial data reliability framework that treats ACID-compliant tables as the single source of truth for both AI and ERP-facing workloads.

To make this reliability framework operational for AI agents, feature-store-centric pipelines emerge as a structural necessity. The Hopsworks feature store is described as a highly available platform that supports transactional, point-in-time analytical, and semantic queries over feature data, with explicit attention to feature reuse, skew avoidance between offline and online pipelines and high-throughput serving [5]. Experimental evaluations in that work demonstrate that optimized query paths yield throughput improvements over managed cloud feature stores, achieving up to several tens of times higher performance when reading large feature sets [5]. The integration of such a feature store into a Delta Lake–based lakehouse creates a dual-layer pattern, where raw and curated financial data reside in ACID tables. In contrast, derived features are surfaced through a governed, versioned feature catalog used by AI agents for inference and retraining. AI/ML-optimized lakehouse proposals reinforce this direction by advocating for the co-location of ML workloads and analytical storage to minimize data movement and simplify operational governance across data and models [1].

Multi-region data governance and regulatory diversity introduce another dimension to the design of enterprise pipelines. Deepak Kaul’s study on AI-powered autonomous compliance management for multi-region data governance highlights the importance of continuous regulatory monitoring, AI-based policy mapping, and automated evidence generation across jurisdictions [6]. The article presents AI-driven categorization of sensitive data, real-time audit readiness, and proactive risk identification as core features of such systems, with case studies reporting reductions in compliance-related expenditure and improvements in audit accuracy [6]. Anichukwueze and co-authors address similar concerns from the viewpoint of cross-border data transfer risk, proposing an AI-driven compliance framework that explicitly models regulatory constraints and technical safeguards for data flows crossing national boundaries [2]. Together, these works motivate a design in which the data platform exposes compliance metadata as first-class entities—linked to regions, regulations, retention policies, and consent—and feeds this metadata into AI agents responsible for classification, routing, and population of control tables.

Figure 1 integrates patterns from Delta Lake literature and AI-compliance research into a generalized reliability architecture for regulated financial pipelines. At the storage layer, all transactional and analytical data are stored in Delta Lake tables that support ACID transactions, schema evolution, and time travel [3; 7]. Above this layer, orchestrated ingestion jobs implement deterministic computation graphs that read from heterogeneous sources (banking systems, claim platforms, payroll inputs), apply harmonization logic, and write into layered tables (bronze, silver, gold). Control tables capture batch manifests, row counts, checksums, and reconciliation keys, while AI agents ingest this metadata and raw data snapshots through the feature store to drive anomaly detection, late-arrival handling, and policy enforcement [5; 6].

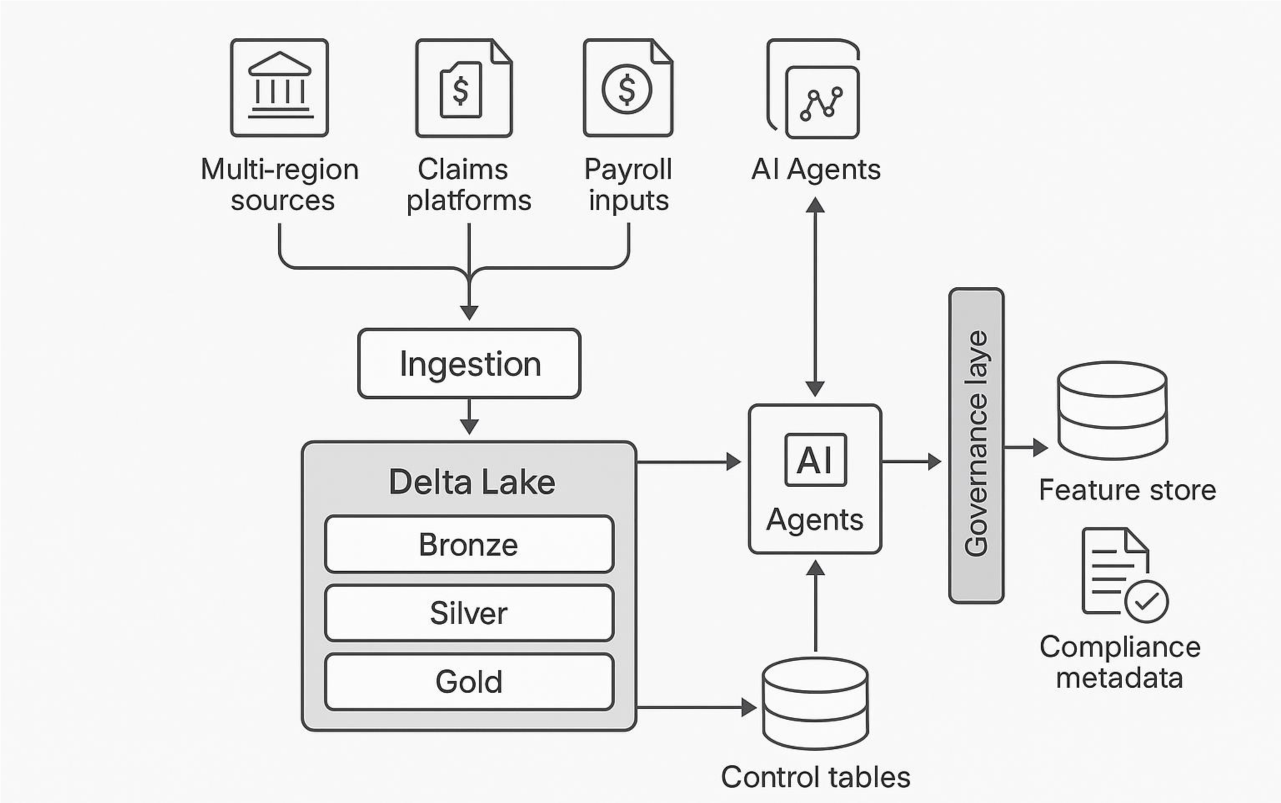

Figure 1: Generalized Delta Lake–based reliability architecture for multi-region financial data pipelines with AI-driven compliance and feature-store integration (compiled by author based on [3; 7])

The second significant result concerns the formulation of a multi-tenant AI lakehouse blueprint tailored to payroll and compliance automation. Insights from financial data-lake innovation studies suggest that financial institutions often separate operational and analytical stores, yet are increasingly experimenting with unified lakehouses that support near real-time reporting and machine learning [10]. In parallel, AI/ML-optimized lakehouse research proposes architectures where ML workloads share storage formats, metadata catalogs, and governance planes with analytical workloads, yielding lower latency and simplified management [1]. Synthesizing these lines of work with feature-store practice [5] and scalable ETL design [8], the proposed blueprint treats each tenant (for example, a payroll customer) as a logical partition over a shared physical lakehouse. Multi-tenant isolation is enforced through column- and row-level security, as well as region-aware storage policies. Shared Delta tables utilize SCD-2 dimensional modeling for entities such as employees, contracts, tax rules, and payment configurations.

Steidl and colleagues describe continuous AI development pipelines that traverse four stages—data handling, model learning, software development, and system operations—and argue that robust, observable data handling underpins every later stage [9]. Their taxonomy of pipeline tasks and triggers guides the structuring of computation graphs in the lakehouse blueprint: ingestion and curation jobs occupy the data-handling layer; feature materialization and labeling feed model learning; deployment workflows map to software development and system operations. AI agents for payroll anomaly detection or compliance checking integrate with this pipeline as consumers of feature groups and as producers of new signals (risk scores, exception tags), which are persisted back into the lakehouse for auditability. Such a design aligns with the continuous development model and offers deterministic, replayable paths from raw financial events to AI-driven actions [5; 8; 9].The third result area addresses validation and reconciliation in high-risk financial pipelines, where multiple files and feeds must be processed consistently despite late arrivals, partial failures, and regulatory deadlines. Classical ETL validation tends to focus on per-batch checks and simple referential integrity, leaving cross-file and multi-period consistency largely to manual procedures. The literature on scalable ETL with Spark and Delta Lake emphasizes the importance of transactional tables and schema evolution in supporting safe reprocessing and late data ingestion without violating invariants [3; 8]. Building on these properties, the article formulates a manifest-driven validation pattern. For each financial cycle, ingestion agents populate a manifest table that lists expected files, their structural signatures, and control totals. As files arrive, ingestion jobs record their actual properties and compute derived metrics. AI agents trained on historical manifest patterns learn to detect anomalies, such as missing jurisdictions, duplicated batches, or inconsistent control totals, across related files.

Compliance-focused studies reinforce the importance of such a pattern. Kaul’s work points to AI-based categorization and risk scoring as tools for prioritizing compliance cases and identifying high-risk data flows [6]. Anichukwueze and his colleagues stress the dangers of fragmented cross-border controls and advocate for the implementation of systematic, AI-supported monitoring of data transfer events [2]. Boosa, in the context of CI/CD pipelines for regulated industries, emphasizes the importance of embedding policy checks directly into deployment workflows to maintain an audit-ready posture [4]. When these insights are combined with the transactional features of Delta Lake [3; 7] and the multi-stage AI pipeline model [9], a coherent multi-file validation and reconciliation approach emerges: manifests and control tables represent the normative state; AI agents interpret deviations; deterministic merge operations, protected by ACID semantics, apply corrections in a conflict-safe way across multiple dependent tables.Finally, financial-enterprise studies on data-lake innovation indicate that lakehouse adoption already yields higher governance maturity, more effective support for AI use cases, and improved real-time analytics in banking and trading contexts [10]. AI-powered compliance frameworks and autonomous governance systems demonstrate reductions in manual audit effort, lower compliance costs, and improved resilience to regulatory change [2; 4; 6]. Feature-store research reports more efficient feature reuse, higher throughput, and reduced risk of offline–online skew [5]. When aligned with scalable ETL and Delta Lake architectures [3; 7; 8], these results support the central claim of the article: that a unified methodology—combining ACID lakehouse storage, feature-store-centered AI integration, continuous AI development pipelines, and AI-driven compliance automation—forms a robust foundation for designing enterprise-grade data pipelines serving AI agents in regulated industries.

4.Discussion

Comparison of the synthesized framework with existing lakehouse and data-lake designs reveals both convergence and gaps. Lakehouse-oriented works foreground ACID transactions, unified storage formats, and layered table structures as responses to scalability and reliability demands in big data analytics [1; 3; 7; 8]. Financial-sector analyses underscore the relevance of these patterns for institutions seeking to consolidate risk, regulatory, and business intelligence workloads under a single data platform [10]. At the same time, many published designs treat AI workloads as an extension of analytical processing rather than as first-class consumers with distinct constraints, including feature freshness, low-latency scoring, and strict governance for model behavior. The Hopsworks feature store addresses these missing elements directly by introducing a dedicated feature platform integrated with training and inference pipelines, enabling high throughput and strict control over feature definitions [5].The unified methodology presented in this article strengthens the connection between lakehouse storage, feature stores, and AI agents, particularly in the context of multi-tenant payroll and financial systems. Table 1 summarizes characteristic properties of several influential lakehouse-oriented contributions and shows how they align with the requirements of regulated environments.

Table 1: Comparative characteristics of recent lakehouse and related architectures relevant for regulated financial data platforms [1; 3; 5; 7; 8; 10]

Source	**Primary domain**	**Architectural focus**	**Reliability and governance features**	**AI/ML integration focus**
Aileni [1]	General AI/ML workloads	AI/ML-optimized lakehouse with unified storage and compute	Emphasis on scalable storage, schema unification, and performance for ML pipelines	Tight coupling of lakehouse with ML workflows and resource optimization
Armbrust and his colleagues [3]	General big data processing	Delta Lake ACID table layer over cloud object storage	Transaction log with ACID, schema evolution, time travel, and reduced data corruption	Foundation for reliable training and inference datasets
de la Rúa Martínez and his colleagues [5]	Machine-learning platforms	Distributed feature store tightly integrated with pipelines	Centralized, governed feature catalog and prevention of offline–online skew	Dedicated feature management, high-throughput serving, and reuse
Lee and his colleagues [7]	Enterprise data platforms	Complete lakehouse architecture on Delta Lake	Governance, performance tuning, medallion layering, and operational best practices	System-level view of AI workloads as part of lakehouse usage
Oye [8]	Big-data warehouses	Spark–Delta ETL architecture	Reliable batch and streaming ingestion with transactional consistency	Provision of consistent analytical and ML-ready tables
Swamy [10]	Financial enterprises	Modern data lakes and lakehouses in finance	Governance frameworks, security controls, and real-time processing capabilities	Support for AI/ML use cases in financial institutions

As indicated in Table 1, existing designs provide robust building blocks but rarely address, in a single scheme, multi-tenant isolation, multi-region harmonization, feature-store integration, and AI-driven compliance. The proposed methodology integrates these features into a lakehouse configuration that serves both ERP-style financial closing and AI agents for anomaly detection, payroll optimization, and compliance monitoring. Studies of financial data-lake innovation [10] confirm that institutions are increasingly pursuing such unification. Meanwhile, scalable ETL and Delta Lake research provides concrete techniques—such as transaction logs, schema evolution, and time travel—to maintain the reliability of these systems [3; 7; 8]. From a practical standpoint, this integration supports design goals familiar from large-scale enterprise work: a single, ACID-governed storage layer, deterministic ingestion graphs, feature reuse, and minimization of bespoke integration code.The second central axis of comparison concerns compliance automation and AI-assisted governance. AI-powered autonomous compliance research demonstrates that AI systems effectively handle multi-region regulatory complexity through continuous policy mapping, intelligent data categorization, and proactive risk detection, often resulting in reduced compliance costs and improved audit readiness [6]. Enterprise-wide AI-driven compliance frameworks for cross-border data transfer provide a structural view of control points, risk metrics, and architectural patterns for safe data movement [2]. AI-augmented continuous delivery work demonstrates how compliance checks can be deeply embedded into software delivery pipelines in heavily regulated environments [4]. When these approaches are confronted with the lakehouse-oriented literature, a gap emerges: while ACID lakehouses provide strong technical guarantees for data consistency, they do not, by themselves, encode regulatory semantics, jurisdiction-specific policies, or compliance evidence.

To clarify the interplay between these factors, Table 2 contrasts compliance-oriented capabilities described in AI governance sources with the mechanisms surfaced in lakehouse and feature-store research.

Table 2: Compliance-oriented capabilities in AI-driven governance frameworks versus lakehouse and feature-store platforms [2; 4–7; 10]

Capability	**AI-driven compliance and DevOps literature**	**Lakehouse / feature-store literature**	**Implications for enterprise AI pipelines in regulated industries**
Regulatory policy mapping and updates	Automatic mapping of legal texts to organizational policies and continuous monitoring of regulatory change	Governance focuses on schemas, access control, and data quality; legal norms remain external	Pipelines need metadata structures where AI agents can record regulatory interpretations and attach them to datasets and features
Real-time audit readiness and evidence	Continuous logging, audit trails, and the generation of machine-readable evidence for regulators	Transaction logs and time travel provide a technical history of data changes	Linking ACID logs and control tables with compliance evidence yields end-to-end auditability from ingestion to AI decisions
Risk scoring and anomaly detection	AI-based detection of high-risk data flows, policy violations, and suspicious behavior	Limited discussion, more focused on data quality and reliability than on regulatory risk	Feature stores and lakehouses must expose compliance-relevant signals as features consumed by AI agents to prioritize remediation
Integration into delivery and operational pipelines	Compliance checks embedded into CI/CD and operational workflows	ETL orchestration and data pipeline scheduling are described, but with limited explicit compliance linkage	Deterministic computation graphs should include compliance checkpoints and fail-safe branches informed by AI risk assessments

The comparison in Table 2 indicates that compliance-oriented AI research and lakehouse-oriented engineering research are complementary rather than overlapping. AI-driven governance work specifies what compliance automation should achieve in multi-region cloud deployments, including policy mapping, real-time evidence, risk scoring, and continuous improvement [2; 4; 6]. Lakehouse, Delta Lake, and feature-store literature, in turn, provide the technical substrate that enables immutable history, transactional integrity, and reproducible datasets for AI workloads [3; 5; 7; 8; 10]. The methodology proposed in this article connects these strands by treating compliance metadata as first-class citizens in the data model: tables for policies, obligations, data categories, consent artifacts, and region-scoped storage requirements become part of the lakehouse schema. AI agents trained on historical compliance data write risk scores and policy decisions back into these tables, which then influence ingestion rules, routing decisions, and reconciliation patterns.From the viewpoint of continuous AI development, the pipeline model derived by Steidl and co-authors supports this integration. The four stages identified in their multivocal review—data handling, model learning, software development, and system operations—frame the areas where reliability, AI, and compliance concerns intersect [9]. In the data-handling stage, ACID lakehouse design and scalable ETL patterns ensure that inputs for AI agents and ERP systems are consistent and fully traceable [3; 7; 8]. In the model-learning stage, feature stores and lakehouse integration provide repeatable, point-in-time datasets [1; 5]. During software development and system operations, AI-augmented continuous delivery and autonomous compliance frameworks inject regulatory intelligence and risk-aware controls into deployment and run-time behavior [2; 4; 6]. By aligning the unified methodology with this four-stage pipeline, the article extends current CI/CD and MLOps practice toward a discipline where reliability, AI, and compliance are treated as a single design problem rather than independent tracks.

5.Conclusion

The proposed methodology consolidates three engineering lines described in the reviewed literature—ACID lakehouse storage, feature-store-centered AI integration, and compliance automation—into a single design scheme for regulated enterprises. Delta Lake research motivates the selection of transactional table storage with schema evolution and time travel as the foundation for reproducible datasets and recoverable ETL execution. At the same time, enterprise lakehouse guidance clarifies how layered table design and governance primitives operationalize these guarantees at platform scale. On this basis, the article presents a reliability construct for financial data pipelines that formalizes deterministic ingestion, lineage capture across layers, and the use of control tables (manifests, checksums, row counts, and reconciliation keys) as audit-ready evidence artifacts, rather than auxiliary monitoring outputs.

A second achieved result is a multi-tenant AI lakehouse blueprint oriented to payroll and compliance workloads. The blueprint aligns AI/ML-optimized lakehouse proposals with feature-store engineering by separating raw/curated transactional data from governed feature materialization and serving, while preserving point-in-time correctness for training and inference. The design specifies tenant partitioning over shared physical storage, tenant isolation through fine-grained access control and region-scoped policies, and payroll-relevant dimensional modeling with SCD-2 for employee, contract, tax, and payment entities. The continuous AI pipeline model from prior studies provides a systematic mapping from ingestion and curation to feature generation, model learning, deployment, and system operations, enabling a replayable path from source events to AI-agent outputs and subsequent persistence of AI-produced signals for traceability.

A third notable achievement is the formulation of a manifest-driven, multi-file validation and reconciliation pattern for high-risk financial ETL. The pattern treats "expected vs. observed" file states as first-class records in a manifest table and binds them to transactional merge procedures in ACID tables. Cross-file consistency checks, late-arrival handling, and conflict-safe correction flows are expressed through deterministic computation graphs. At the same time, AI agents act as evaluators of deviations by learning the historical normality of manifests and control totals. In comparison with per-batch validation practices, the pattern shifts reconciliation from manual exception handling to a structured evidence workflow grounded in transaction logs, control tables, and versioned feature artifacts.The conclusion of the study is not limited to architectural synthesis; it clarifies what was accomplished at the methodological level: (i) a reliability framework that operationalizes ACID semantics, lineage, and controls as an integrated audit mechanism for financial pipelines; (ii) a multi-tenant lakehouse specification that connects payroll processing, feature reuse, and AI-agent execution under governance constraints; (iii) a reconciliation methodology that scales from single-feed validation to multi-feed, deadline-driven financial closing scenarios. These constructs jointly articulate how enterprise data platforms can be engineered to embed regulatory assurance into storage formats, metadata, and pipeline control structures, rather than relying solely on external procedures for enforcement.

Further improvement directions follow directly from the limits of the current work. The proposed methodology requires empirical validation under production conditions, with quantified reliability and compliance metrics, including reconciliation labor reduction, anomaly detection precision/recall over manifest deviations, mean time to recovery under partial failures, end-to-end lineage completeness, and audit evidence retrieval latency. A second enhancement concerns jurisdiction-specific policy encoding: the paper outlines compliance metadata tables; however, a more rigorous treatment demands a formal policy model (including obligations, retention, consent, residency, and transfer restrictions) coupled with testable invariants and automated evidence generation routines. A third development track concerns security and isolation proofs for multi-tenancy, including adversarial evaluation of row/column security, cross-tenant leakage risks in feature serving, and governance enforcement across regions. A fourth line targets AI-agent governance, including standardized logging of agent decisions, drift monitoring for compliance classifiers, the reproducibility of agent-triggered interventions, and human-in-the-loop escalation protocols for regulatory exceptions. Finally, a reference implementation and comparative benchmarking against alternative lakehouse stacks and managed feature platforms would strengthen the methodological claims by connecting design constructs to measurable cost, performance, and operational outcomes in representative regulated workloads.

References

[1] Aileni, A. R. (2025). AI/ML optimized lakehouse architecture: A comprehensive framework for modern data science. World Journal of Advanced Engineering Technology and Sciences, 15(2).

[2] Anichukwueze, C. C., Osuji, V. C., & Oguntegbe, E. E. (2025). Enterprise-wide AI-driven compliance framework for real-time cross-border data transfer risk mitigation. Computer Science & IT Research Journal, 6(1).

[3] Armbrust, M., Das, T., et al. (2020). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411–3424.

[4] Boosa, S. (2025). AI-augmented continuous delivery in regulated industries: A compliance-first strategy. International Journal of AI, BigData, Computational and Management Studies, 6(1), 106–115. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I1P111

[5] de la Rúa Martínez, J., Buso, F., Kouzoupis, A., Ormenisan, A. A., Niazi, S., et al. (2024). The Hopsworks feature store for machine learning. In Proceedings of the 2024 ACM SIGMOD/PODS Conference Companion (pp. 135–147).

[6] Kaul, D. (2024). AI-powered autonomous compliance management for multi-region data governance in cloud deployments. Journal of Current Science and Research Review, 2(3), 82–98.

[7] Lee, D., Wentling, T., Haines, S., & Babu, P. (2024). Delta Lake: The definitive guide: Modern data lakehouse architectures with data lakes. O’Reilly Media.

[8] Oye, S. (2023). Scalable ETL architecture with Apache Spark and Delta Lake for big data warehouses.

[9] Steidl, M., Felderer, M., & Ramler, R. (2023). The pipeline for the continuous development of artificial intelligence models—Current state of research and practice. Journal of Systems and Software, 203, 111615. https://doi.org/10.1016/j.jss.2023.111615

[10] Swamy, A. H. (2025). Innovations in data lake architectures for financial enterprises. World Journal of Advanced Research and Reviews, 26(1), 1975–1982. https://doi.org/10.30574/wjarr.2025.26.1.1252