International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 492-499
https://doi.org/10.70917/ijcisim-2026-5323
Article

Architecting Absolute Data Lineage: Automated Compliance Governance and Metadata Tracking in Legacy-to-Cloud Re-Platforming

Prashant Vikas Patil

Independent Researcher, USA. ORCID ID: https://orcid.org/0009-0007-8651-8386

Abstract: Enterprises retiring legacy mainframe platforms in favor of cloud-native infrastructure routinely encounter a governance blind spot: the ability to trace where a given piece of data originated, how it was transformed, and who touched it along the way degrades sharply once workloads move into distributed, multi-service cloud environments. This weakness matters most in regulated industries, where auditors and compliance teams depend on that traceability to demonstrate adherence to data protection and financial reporting requirements. Manual, retrospective audit processes, built around periodic spreadsheet reconciliation and log sampling, were designed for the comparatively static topology of mainframe systems and do not scale to the transient, horizontally distributed nature of cloud workloads. This paper proposes an architectural framework for automated data lineage capture and compliance verification, structured around a three-phase pipeline: non-intrusive metadata harvesting at the point of data movement, structural mapping of that metadata into a queryable graph representation, and continuous verification of data pathways against governance policy. The framework is informed by direct practitioner experience leading legacy-to-cloud re-platforming programs and the accompanying CI/CD governance, stage-gate review, and audit-readiness disciplines that such programs require. Rather than treating lineage tracking as a downstream compliance checkbox, the proposed design treats it as a first-class architectural concern, decoupled from transactional workloads so that audit visibility does not come at the cost of processing throughput. The paper situates this proposal against the existing literature on metadata lineage, data catalog systems, and blockchain-based audit trails, and discusses where a decoupled, graph-based capture architecture addresses gaps that batch-oriented and single-authority approaches leave open. The discussion also considers where this architecture would need empirical validation before enterprise adoption, and identifies directions for future work, including AI-assisted schema reconciliation across heterogeneous legacy sources.

Keywords: Data lineage; Cloud migration governance; Metadata management; Regulatory compliance; Legacy modernization

1. Introduction

Enterprise IT organizations have spent the past decade retiring mainframe and AS/400-class systems in favor of distributed cloud-native platforms, motivated by scalability, vendor ecosystem support, and the operational cost of maintaining aging hardware. This shift, however, exposes a structural weakness in how these organizations track their own data. On a mainframe, a data record's path through batch jobs and file transfers is comparatively easy to reconstruct after the fact, because the set of systems involved is small and well understood by the teams operating them. Once that same data begins moving through containerized services, managed databases, and third-party API integrations, the number of possible pathways multiplies, and the tooling that compliance teams historically relied on, manual log review and periodic spreadsheet reconciliation, no longer keeps pace [1], [3].

The cost of this gap shows up long after the migration itself is declared complete. In regulated industries such as insurance, financial services, and supply chain logistics, the inability to produce an accurate, current picture of data movement during or after a migration creates real exposure: audit findings, delayed sign-offs, and in some cases the need to freeze migration work entirely while manual tracing is performed. Existing metadata lineage tools, commercial

data catalogs, open-source lineage trackers, and blockchain-based audit-trail proposals, each address part of this problem, but the literature and available tooling reveal a consistent pattern: lineage capture is often bolted onto data platforms after the fact, rather than designed as a first-class architectural concern from the start of a migration program [5], [6].

1.1 Contribution of Paper

This paper makes three contributions. First, it proposes a three-phase architecture, metadata harvesting, graph-based structural mapping, and continuous compliance verification, designed specifically for the transitional period of a legacy-to-cloud re-platforming program, when data pathways are most volatile and least documented. Second, it positions this architecture against the existing literature on data lineage, metadata catalogs, and audit-trail mechanisms, identifying where a decoupled, non-intrusive capture layer addresses limitations of batch-oriented and single-authority approaches. Third, it grounds this architecture in the governance disciplines, stage-gate technical reviews, CI/CD promotion controls, separation-of-duties enforcement, that arise directly from enterprise re-platforming practice, rather than treating lineage tracking as a purely academic or vendor-tooling problem.

2. Related Work

Metadata lineage has a substantial research history predating the current wave of cloud migration. Early foundational work characterized data provenance in terms of why, how, and where a data element originated [3], [4], and subsequent surveys catalogued the range of provenance models used across scientific and enterprise computing [1]. These foundational models remain conceptually relevant, but the tooling built on top of them has evolved considerably.

Contemporary metadata management research falls into three broad categories relevant to this paper. The first concerns metadata catalog and knowledge-graph systems designed to make heterogeneous enterprise data discoverable and governable. Cherradi et al. [5] examined data lake governance built around IBM's Watson Knowledge Catalog, finding that catalog-centric governance improves discoverability but depends heavily on how completely metadata is captured at ingestion. Sawadogo et al. [6] similarly addressed metadata management for unstructured textual documents within data lakes, noting that generic metadata schemas struggle to capture the semantic context needed for compliance-grade traceability. Luo et al. [8] proposed a knowledge-graph-based approach to enterprise big data management, demonstrating that graph representations handle the many-to-many relationships in enterprise metadata more naturally than relational catalog schemas.

The second category concerns lineage tracking at genuine operational scale. Jacques-Silva et al. [7] described a unified lineage system built to track data provenance across a large distributed production environment, emphasizing that lineage systems built for scale must treat metadata capture as a background, asynchronous process rather than a synchronous check that could slow production workloads. This finding is consistent with the architectural separation this paper proposes between the transactional data plane and the metadata capture plane. Vieira et al. [12] took a more data-warehouse-specific approach, extending ETL tooling with provenance support to improve data quality auditing, while Johns et al. [11] applied a comparable provenance-tracking discipline to clinical data warehouses, underscoring that the underlying problem, reconstructing how a value in a downstream system traces back to its source, recurs across industries with very different regulatory regimes.

The third category concerns tamper resistance and audit-trail integrity, an area where blockchain-based proposals have gained attention. Nweje [17] and Rodrigues et al. [16] both examined blockchain mechanisms for securing audit trails and ETL data integrity, arguing that an immutable ledger strengthens the evidentiary value of a lineage record. These proposals address a real weakness in conventional lineage systems, namely that the lineage record itself can, in principle, be altered, but the added consensus and storage overhead of blockchain-based approaches has limited their adoption in the very-high-throughput environments most likely to need lineage tracking in the first place, a tension this paper's design chooses to route around by relying on append-only graph storage with cryptographic hashing rather than a full distributed ledger.

Alongside these categories, the broader migration and modernization literature offers relevant context. Karuppiah [14] proposed a risk-aware framework for multi-domain cloud migration, and Almeida et al. [15] conducted a tertiary study on legacy-to-microservice modernization, both observing that governance and data-quality considerations are frequently under-specified relative to the technical migration steps themselves. Kass [21] examined AI-assisted approaches to legacy code comprehension during modernization, a complementary problem to the one addressed here: understanding legacy code logic is a prerequisite to understanding what a legacy system's data actually represents, which in turn is a prerequisite to lineage mapping across the old and new environments.

Table 1 summarizes how these approaches compare against the requirements of a re-platforming-specific lineage architecture.

Table 1. Comparison of Data Governance and Lineage Approaches

Approach	Real-Time Capability	Migration-Phase Suitability	Overhead
Manual / batch audit	No	Poor	Low (high labor)
Catalog-centric governance	Partial	Moderate	Moderate
Knowledge-graph metadata mgmt.	Yes, if continuous	Good	Moderate
Operational-scale lineage systems	Yes	Good	Low (async)
Blockchain-based audit trail	Yes, with latency	Limited	High
Proposed decoupled architecture	Yes, continuous	Designed for this case	Low

What the existing literature does not directly address is the specific transitional window of an active re-platforming program, when data simultaneously exists in legacy and cloud-native form, integration patterns are still being finalized, and compliance teams need visibility into a system that is, by definition, in flux. This is the gap the proposed framework targets.

3. Methodology

3.1 Design Rationale

Anyone who has sat through a stage-gate review for a code change, and then watched a data integration pattern go live with no equivalent review, has seen this gap firsthand. Technical governance, stage-gate design reviews, CI/CD promotion policies, separation-of-duties controls across development, test, and production environments, is well established for code, but rarely extended with equivalent rigor to data pathways during a migration. Enterprises that would never allow an unreviewed code change to reach production routinely allow data integration patterns to be stood up ad hoc during a migration, with lineage documentation deferred to "after go-live." The framework proposed here applies the same discipline to data pathways that CI/CD governance already applies to code: continuous, automated verification rather than periodic manual review.

3.2 Three-Phase Architecture

The framework operates across three phases, illustrated in Fig. 1.

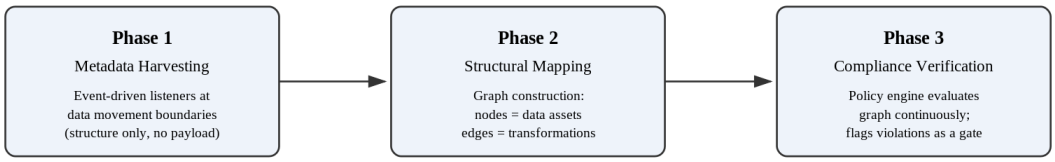

Fig. 1. Three-phase automated lineage capture and verification pipeline.

Fig. 1. Three-phase automated lineage capture and verification pipeline.

Phase 1 - Metadata Harvesting. Lightweight listeners are attached at the boundaries where data moves between systems, database change-data-capture streams, API gateways, message queue topics, and file transfer endpoints. These listeners record structural metadata only: source and destination identifiers, timestamps, schema or field-level descriptors, and transformation identifiers where applicable. They do not read or persist the underlying data payload, which keeps the capture layer outside the compliance boundary of the sensitive data itself and reduces the attack surface introduced by the lineage system. This design choice reflects a lesson familiar from CI/CD pipeline design: instrumentation that shares a failure domain or a performance budget with the system it monitors becomes an operational liability rather than a safeguard, a concern also raised in the context of high-throughput lineage capture [7].

Phase 2 - Structural Mapping. Harvested metadata is streamed asynchronously into a graph data store, where nodes represent data assets (tables, files, message topics) and edges represent observed transformations or movements between them, timestamped and versioned. A graph representation is used rather than a relational schema because the relationships involved, one field feeding multiple downstream consumers, or multiple upstream sources merging into one, are naturally many-to-many, a structural fit noted in prior enterprise knowledge-graph work [8]. This mapping runs continuously rather than in scheduled batches, so that the graph reflects the current, in-flight state of the migration rather than a stale snapshot.

Phase 3 - Compliance Verification. A policy engine continuously evaluates the graph against a defined set of governance rules, for example, that data classified as sensitive must not traverse into a storage location outside an approved jurisdiction, or that a given data element must retain an unbroken lineage chain back to its system of record. Where the CI/CD analogy in this paper's design rationale is most direct, this phase functions as a stage-gate check for data pathways: violations are flagged for review before they are treated as an approved integration pattern, rather than being discovered retrospectively during an audit cycle.

3.3 Decoupled Topology

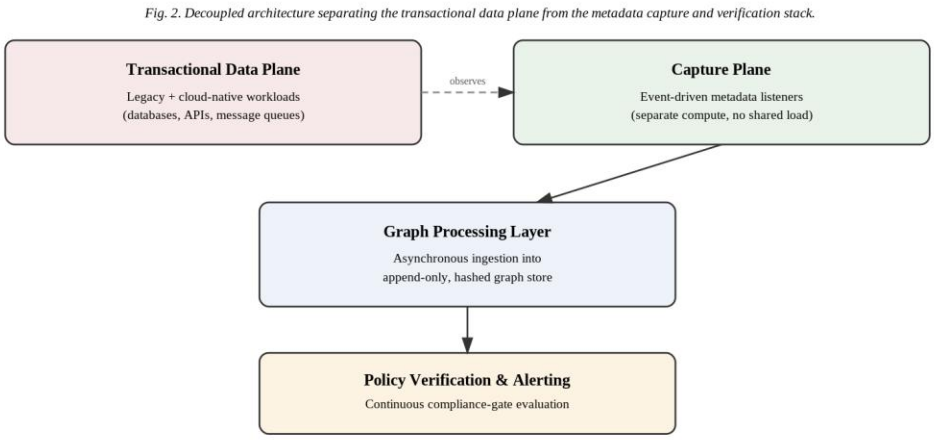

Fig. 2. Decoupled architecture separating the transactional data plane from the metadata capture and verification stack.

The three phases are deployed on infrastructure logically and physically separate from the transactional systems being monitored. This separation is the architectural analog of the environment segregation (development, test, staging, production) already standard in mature CI/CD practice [9]: just as a build pipeline should not compete with production traffic for the same compute capacity, the lineage capture plane should not compete with transactional workloads for database connections, network bandwidth, or compute. Where a re-platforming program already operates a CI/CD pipeline with defined promotion gates [10], the compliance verification phase can plausibly be integrated as an additional gate rather than a separate governance process, reducing the organizational overhead of introducing a new discipline.

4. Results and Discussion

This section works from the literature rather than from a completed deployment, since the paper is an architectural proposal rather than an empirical study. It discusses expected outcomes with reference to comparable systems reported in the literature, and situates those findings against the design choices made in Section 3.

Systems that separate metadata capture from transactional processing, an architectural choice shared by the framework proposed here, have been reported to sustain lineage tracking at substantial operational scale without materially affecting the throughput of the monitored systems [7]. This is consistent with the design rationale in Section 3.3: because the capture plane observes structural metadata only, and does so asynchronously, its resource demands do not scale in step with the volume or sensitivity of the underlying data payload.

Catalog-centric and graph-based governance approaches have been reported to improve the discoverability and traceability of enterprise metadata relative to manual documentation, though completeness depends heavily on how thoroughly metadata is captured at the point of ingestion rather than reconstructed after the fact [5], [8]. This finding directly supports the emphasis in Phase 1 of the proposed framework on harvesting metadata at the moment of data movement, rather than attempting to infer lineage retrospectively from logs or documentation, an approach whose limitations are well documented in provenance research going back to the earliest lineage surveys [1].

Where the literature is more cautionary is on the cost of stronger tamper-evidence guarantees. Blockchain-based audit-trail approaches offer a meaningfully stronger integrity guarantee than an ordinary append-only log, but the consensus and storage overhead they introduce has limited their use in high-throughput enterprise settings [16], [17]. Table 2 summarizes reported characteristics of these approaches alongside the design position taken in this paper.

Table 2. Literature-Reported Characteristics Summary

Approach	Real-Time Capability	Overhead Profile	Governance Fit
Catalog-centric governance	Partial	Moderate	Discoverability-focused
Knowledge-graph metadata mgmt.	Yes, if continuous	Moderate	Strong structural fit
Operational-scale lineage (async)	Yes	Low	Strong high-throughput fit
Blockchain-based audit trail	Yes, with latency	High	Strong tamper-evidence, weak throughput fit
Proposed framework	Yes	Low	CI/CD-gate-integrated governance fit

The proposed framework's use of cryptographic hashing over an append-only graph store, rather than a full blockchain, reflects an attempt to capture much of the tamper-evidence benefit reported in [16] and [17] without inheriting the throughput cost that has limited blockchain-based lineage adoption in production settings.

From a practitioner governance perspective, the framework's treatment of compliance verification as a stage-gate, analogous to a CI/CD promotion gate, addresses a pattern observed repeatedly in re-platforming programs: technical governance discipline is rarely the limiting factor for code changes, but is inconsistently applied to the data integration patterns that accompany a migration. Extending the same review discipline to data pathways does not require new organizational structures where a program already operates mature CI/CD governance; it requires treating lineage verification as one more gate in an existing process rather than a separate compliance initiative introduced after the fact.

5. Limitations and Future Work

The scope here is architectural, not empirical, and that shapes every limitation that follows. First, the framework has not been benchmarked against a production workload, and the performance characteristics discussed in Section 4 are drawn from comparable systems in the literature rather than from direct measurement of this specific architecture. Pilot deployment within a bounded, non-critical data domain would be a reasonable first validation step before broader adoption. Second, the policy engine in Phase 3 depends on governance rules being defined with sufficient precision to be machine-evaluable; translating existing manual audit criteria into this form is itself a non-trivial exercise that this paper does not address in detail. Third, heterogeneous legacy sources, particularly those with poorly documented or inconsistently labeled schemas, present a metadata-harvesting challenge that the framework's current design does not fully resolve; future work should examine machine-learning-assisted schema reconciliation, building on approaches already explored for automated legacy code comprehension [21], to reduce the manual effort of onboarding poorly documented legacy sources into the lineage graph. Finally, broader validation across multiple re-platforming programs, industries, and regulatory contexts would be needed before the framework's claims of general applicability could be considered established.

6. CRediT Author Contribution Statement

Prashant Vikas Patil: Conceptualization, Methodology, Writing - Original Draft, Writing - Review & Editing.

7. Acknowledgments

The author used Claude (Anthropic, claude-sonnet model) to assist with manuscript drafting and language refinement. The author takes full responsibility for the accuracy, originality, and integrity of all content presented in this manuscript, and confirms that all analysis and conclusions reflect independent work and verified sources.

8. Statements and Declarations

Ethical Approval: This study did not involve human participants, animal subjects, or identifiable personal data requiring ethics board review.

Consent to Participate: Not applicable.

Consent to Publish: Not applicable.

Funding: The author declares that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests: The author has no relevant financial or non-financial interests to disclose.

Availability of Data and Materials: This is an architectural/conceptual framework paper; no new empirical data were generated.

REFERENCES

1. Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," ACM SIGMOD Record, vol. 34, no. 3, pp. 31-36, 2005. https://doi.org/10.1145/1084805.1084812
2. L. Moreau et al., "The Open Provenance Model core specification (v1.1)," Future Generation Computer Systems, vol. 27, no. 6, pp. 743-756, 2011. https://doi.org/10.1016/j.future.2010.07.005
3. J. Cheney, L. Chiticariu, and W.-C. Tan, "Provenance in Databases: Why, How, and Where," Foundations and Trends in Databases, vol. 1, no. 4, pp. 379-474, 2009. https://doi.org/10.1561/1900000006
4. P. Buneman, S. Khanna, and W.-C. Tan, "Why and Where: A Characterization of Data Provenance," in Database Theory - ICDT 2001, Springer, pp. 316-330, 2001. https://doi.org/10.1007/3-540-44503-X_20
5. M. Cherradi, F. Bouhafer, and A. EL Haddadi, "Data lake governance using IBM-Watson knowledge catalog," Scientific African, vol. 21, e01854, 2023. https://doi.org/10.1016/j.sciaf.2023.e01854
6. P. Sawadogo, T. Kibata, and J. Darmont, "Metadata Management for Textual Documents in Data Lakes," in Proc. 21st Int. Conf. Enterprise Information Systems, 2019, pp. 72-83. https://doi.org/10.5220/0007706300720083
7. G. Jacques-Silva et al., "Unified Lineage System: Tracking Data Provenance at Scale," in Companion of the 2025 Int. Conf. on Management of Data (SIGMOD-Companion '25), 2025, pp. 457-470. https://doi.org/10.1145/3722212.3724458
8. P. Luo, J. Chen, and J. Li, "Enterprise big data management based on Knowledge Graph," in Proc. 2021 4th Int. Conf. Computing and Big Data, 2021, pp. 54-59. https://doi.org/10.1145/3507524.3507534
9. R. Bronzwaer, "The Role of Procedural Segregation of Duties in the Installation of System Software," EDPACS, vol. 28, no. 3, pp. 1-9, 2000. https://doi.org/10.1201/1079/43272.28.3.20000901/30626.1
10. H. K. Mupparapu, "Azure DevOps CI/CD Pipeline Design for Regulated Financial Software Deployment Environments," International Journal of AI, BigData, Computational and Management Studies, vol. 4, pp. 209-219, 2023. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I4P121
11. M. Johns, L. Baum, and F. Prasser, "Tracking provenance in clinical data warehouses for quality management," International Journal of Medical Informatics, vol. 193, 105690, 2025. https://doi.org/10.1016/j.ijmedinf.2024.105690
12. M. Vieira, T. de Oliveira, L. Cicco, D. de Oliveira, and M. Bedo, "From Tracking Lineage to Enhancing Data Quality and Auditing: Adding Provenance Support to Data Warehouses with ProvETL," in Proc. 26th Int. Conf. Enterprise Information Systems, 2024, pp. 313-320. https://doi.org/10.5220/0012634500003690
13. S. Chundru, "AI-Driven Data Provenance: Tracking and Verifying Data Lineage," FMDB Transactions on Sustainable Computing Systems, vol. 2, no. 3, pp. 107-118, 2024. https://doi.org/10.69888/FTSCS.2024.000258
14. S. Karuppiah, "Enterprise Cloud Migration Framework for Multi-Domain Risk-Aware Legacy Modernization," International Journal of Research and Analytical Reviews, vol. 12, no. 3, 2025.
15. N. Almeida, G. Campos, F. Moraes, and F. Affonso, "Modernization of Legacy Systems to Microservice Architecture: A Tertiary Study," in Proc. 26th Int. Conf. Enterprise Information Systems, 2024, pp. 581-592. https://doi.org/10.5220/0012633300003690
16. R. Rodrigues, B. Oliveira, and O. Belo, "Ensuring Data Integrity on ETL Systems Using Blockchain," in Proc. 28th Int. Conf. Enterprise Information Systems, 2026, pp. 378-385. https://doi.org/10.5220/0015006800004018
17. U. Nweje, "Blockchain Technology for Secure Data Integrity and Transparent Audit Trails in Cybersecurity," International Journal of Research Publication and Reviews, vol. 5, no. 12, pp. 4902-4916, 2024. https://doi.org/10.55248/gengpi.5.1224.0211
18. J. Sirisha Devi, P. V. Bhaskar Reddy, M. Purushotham Reddy, and A. Jayanthi, "SecureVault: Architecting a Privacy-First Cryptographic Layer for Multi-Cloud Data Governance," International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 9s, pp. 673-686, 2026.

19. J. Bhagwan, S. Rani, S. Kumar, and S. Godara, "DWCSO: A Hybrid Intelligent Algorithm for Multi-Type Workload Scheduling in Cloud Computing," International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 10s, pp. 227-244, 2026.
20. R. V. J. Fernandes, S. M. Tauseef, and N. A. Siddiqui, "An Integrated and Validated Ergonomic Risk Assessment Framework for the Construction Industry," International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 11s, pp. 1-19, 2026.
21. E. Kass, "Automating Legacy System Modernization for Cloud Readiness Using AI Techniques," Authorea preprint, 2025.