

RESEARCH ARTICLE

Federated lineage and metadata intelligence: A framework for scalable data governance in distributed enterprise architectures**Ratanchur Sarkar***Independent Researcher, USA [Senior Member—IEEE]***Corresponding Author:** Ratanchur Sarkar, **E-mail:** ratanchursarkar@gmail.com ORCID: 0009-0009-2287-7464

ABSTRACT

Large enterprises with heterogeneous distributed data platforms cannot achieve end-to-end verifiable data lineage through a centralized metadata architecture, including Data Mesh and Lakehouse architectures with domain-based data ownership. Traditional centralized data lineage solutions lack the scalability to meet the requirements of distributed data platforms with siloed domains, heterogeneous execution engines, and decentralized architecture. Presented here is the Federated Lineage and Metadata Intelligence Framework (FLMIF): a five-layer architectural model to facilitate balancing between domain autonomy and organizational governance in large regulated organizations. FLMIF enables automated lineage capture from heterogeneous execution engines using the OpenLineage standard, ML-based metadata inference and enrichment, multi-dimensional lineage quality scoring, policy-driven federated governance without central bottlenecks, and continuous evidence automation for regulatory compliance. The framework is developed using a Design Science Research method and assessed with a scenario-based architecture evaluation against regulatory and architectural requirements (BCBS 239, GDPR, SOX, DORA, EU AI Act). Relative to centralized and manually maintained approaches, FLMIF is projected to materially improve lineage coverage, substantially reduce the manual effort of compliance-evidence collection, accelerate root-cause analysis, and deliver near-real-time metadata freshness. A proof-of-concept implementation on synthetic data confirms that the framework's capture, scoring, and root-cause mechanisms are computationally sound and scale to graphs of thousands of datasets. Quantitative projections, with the underlying assumptions stated, are provided in Section 7; empirical validation in production deployments is identified as the primary future work.

KEYWORDS

Data Lineage; Federated Governance; Metadata Intelligence; Openlineage; Data Mesh; Bcbs 239; Enterprise Data Governance; Regulatory Compliance; Metadata Quality Scoring; Distributed Architecture

ARTICLE INFORMATION

ACCEPTED: 3 September 2026**PUBLISHED:** 3 October 2026**DOI:** 10.32996/jcsts.2026.8.10.2

1. Introduction

In modern business contexts, for an increasing number of industries—but especially financial services, healthcare, and telecommunications—complex data flows are necessary to support operational reporting and analytics, machine learning, and digital customer experiences. This requires high levels of trust in data with discoverable and trustable data lineage and strong metadata quality. Traditional centralized approaches to data and metadata management are inadequate in a world of distributed architectures and self-service analytics (Halevy et al., 2006; Sawadogo & Darmont, 2021).

The modern enterprise data architecture presents four key challenges in which centralized metadata management fails. First, high architectural heterogeneity: enterprises have hybrid architectures, with on-premise legacy systems, multi-cloud environments, real-time streaming engines (e.g., Apache Flink (Carbone et al., 2015)), microservices architectures, and modern Lakehouse architectures (Armbrust et al., 2020; Armbrust et al., 2021). Second, data decentralization: the Data Mesh model decentralizes data ownership to business domains, easing domain-oriented ownership and governance at the cost of reduced enterprise-level data lineage visibility (Dehghani, 2022; Machado et al., 2022; Wider et al., 2023). Third, the regulatory driver: BCBS 239 standards apply to all material risk data flows by requiring attribute-level lineage. Other regulations include tracking of data subjects under GDPR and auditability of transformations for SOX compliance (Basel Committee on Banking Supervision, 2013; European Parliament and Council, 2016). Fourth, operational velocity: streaming workloads, ML pipelines, and Lakehouse tables change continuously, and hundreds of domain teams change these pipelines daily, making lineage tracking efforts obsolete within days (Sculley et al., 2015).

In the early days, centralized metadata tooling had fundamental scalability and sustainability issues under such conditions: manual lineage tracking is fragile, central teams become bottlenecks, tooling is scattered around the organization, and risk builds up due to lack of lineage capture (Buneman et al., 2001; Sawadogo & Darmont, 2021). A 2019 global research report showed that while nine out of ten organizations recognize that data-driven capabilities are a business opportunity, for most organizations the governance infrastructure needed for trusted data at scale is missing (Ransbotham et al., 2019).

In order to resolve such problems, this paper introduces the Federated Lineage and Metadata Intelligence Framework (FLMIF), which includes: (1) a five-layer federated architecture enabling the integration of domain autonomy and governance by implementing federated lineage collection and federated policy enforcement; (2) a multi-dimensional scoring for lineage quality measurement and quantification in terms of completeness, accuracy, freshness, coverage, and governance; and (3) an automated compliance subsystem providing continuous monitoring, classification, and cryptographically secured audit trails. Section 2 reviews background and related work and positions FLMIF against existing tools. Section 3 states the regulatory and architectural requirements. Section 4 describes the research method. Section 5 presents the FLMIF architecture. Section 6 describes the implementation workflow and maps technologies to layers. Section 7 evaluates the framework and presents the projection model and worked case study. Section 8 sets out the implementation roadmap. Section 9 discusses findings, threats to validity, and limitations. Section 10 concludes.

2. Background and Related Work

2.1 Metadata and Lineage Management Evolution

The first metadata management tools were built during the data warehouse era, when data platforms remained relatively homogeneous. Commercial platforms like Informatica Metadata Manager and Collibra, as well as open-source tools like Apache Atlas, allowed for manual lineage mapping and structured metadata harvesting. While these solutions often excelled in extracting technical metadata from relational databases and their pipelines, their manual lineage maintenance, centralization to a dedicated team, and fragmentation across diverse execution engines made them less than ideal (Sawadogo & Darmont, 2021). The OpenLineage standard, released in 2021, represents a considerable step forward. It automates lineage collection through a standardized approach to metadata applied to three primary entities (datasets, jobs, and runs), and provides extensible facets for domain-level enrichment (OpenLineage Contributors, 2021). FLMIF adopts this open standard rather than a proprietary lineage schema because the framework's federated premise requires agents built independently by different domain teams, and in practice by different vendors' tooling, to emit lineage in a common, tool-agnostic format; a proprietary schema would recreate the central dependency the architecture is designed to avoid, and would tie every future capture agent to a single vendor's roadmap.

2.2 Data Mesh and Decentralized Governance

The Data Mesh, as conceived by Dehghani, is a radical approach to enterprise data architecture based upon four principles: domain-oriented decentralized data ownership, data-as-a-product, self-serve infrastructure, and federated computational governance (Dehghani, 2022). The research conducted by Machado et al. was one of the first to introduce data mesh concepts to an academic audience and to describe the impact of the data mesh model shift on the design of enterprise data architecture (Machado et al., 2022). Wider et al. summarize hands-on experience of a production data mesh platform and observe that decentralized governance requires a conceptual model as well as tooling that enables automated enforcement of governance rules and does not depend on the central Center of Excellence (Wider et al., 2023). Dolhopolov et al. describe a federated governance prototype using Kubernetes and blockchain to address 9 of the 11 governance requirements identified (Dolhopolov et al., 2024). More recently, Hutterer situates Data Mesh within a broader taxonomy of data platforms, data spaces, and data fabrics, arguing that these paradigms are complementary responses to the same decentralization pressure rather than competing architectures (Hutterer, 2025).

2.3 Data Lake Metadata Management

Sawadogo and Darmont reviewed data lake architecture and metadata management and identified four types of metadata: intra-object, inter-object, global, and evolution. They pointed out the difficulties of maintaining fully updated metadata on dynamically changing data lakes and the insufficiency of manual metadata collection for large-scale enterprise data lake use (Sawadogo & Darmont, 2021). This finding indicates the importance of FLMIF Layer 2's automated inference capabilities. The transition from warehouse to Lakehouse, surveyed by Harby and Zulkernine, further increases format and engine heterogeneity that lineage tooling must accommodate (Harby & Zulkernine, 2022).

2.4 Column-Level and ML Lineage

Column-level lineage (CLL) is a development from table-level lineage that traces the origin of values in individual fields. The purpose of CLL includes finding the root causes of data quality violations, analyzing the effects of schema modifications, and monitoring sensitive data flows (Basel Committee on Banking Supervision, 2013). Fragkoulis et al. have reviewed the history of stream processing platforms and highlighted problems related to pipeline lineage complexity and the gap between stream processing and governance capabilities (Fragkoulis et al., 2023). ML model governance adds further complexity in tracking feature lineage, versioning training data, tracking model parameter versions, and recording deployment metadata for AI Act compliance

for high-risk financial AI systems (European Parliament and Council, 2024; Zaharia et al., 2018). Production readiness for such ML systems has itself been formalized as a rubric of testable properties (Breck et al., 2017), which informs the quality dimensions adopted in Section 5.4.

2.5 Gap Analysis and Positioning

Recent systematic reviews substantiate that no existing architectural blueprint integrates automated lineage capture across heterogeneous execution engines, ML-based metadata inference, federated governance-policy enforcement, multi-dimensional quality scoring, and compliance-evidence automation for the regulated enterprise. The systematic multivocal review of catalog tools by Tonnarelli et al., which classifies 39 tools against a 24-feature framework, finds that few tools support the combination of automated lineage, governance, and productization features required for metadata-driven architectures (Tonnarelli et al., 2025). The systematic review of metadata management in Data Mesh by Suguimoto et al. likewise concludes that practical guidance for federated metadata management remains limited and identifies open challenges in scalable, interoperable governance across domains (Suguimoto et al., 2025), a finding echoed by the gray-literature survey of Goedegebuure et al. (Goedegebuure et al., 2024). The broader provenance (Herschel et al., 2017) and data governance (Abraham et al., 2019) literatures provide established conceptual frameworks but do not offer an integrated, regulation-aligned reference architecture. To make the gap concrete, Table 1 synthesizes a capability comparison of representative lineage and governance platforms across the dimensions FLMIF claims to advance. The comparison reflects the author's assessment, informed by the feature framework of the systematic review (Tonnarelli et al., 2025) and the platforms' current product documentation (Apache Software Foundation, n.d.; Collibra, n.d.; Databricks, n.d.; DataHub Project, n.d.; Microsoft, n.d.; OpenMetadata, n.d.); it positions FLMIF as an integrating reference architecture rather than a competing tool, designed to be realized using several of these tools as components.

Table 1. Capability comparison of representative lineage/governance platforms versus FLMIF (developed by the author, informed by the systematic catalog-tool review (Tonnarelli et al., 2025) and current product documentation (Apache Software Foundation, n.d.; Collibra, n.d.; Databricks, n.d.; DataHub Project, n.d.; Microsoft, n.d.; OpenMetadata, n.d.)).

Capability	Apache Atlas	DataHub	OpenMetadata	Collibra	MS Purview	Unity Catalog	FLMIF (this work)
Auto-capture across heterogeneous engines	Partial	Partial	Partial	Limited	Partial	Engine-bound	Yes (OpenLineage agents)
OpenLineage-native ingestion	Via bridge	Yes	Yes	Limited	Limited	Limited	Yes (core)
ML-based metadata inference	No	Limited	Limited	Add-on	Limited	Limited	Yes (Layer 2)
Federated policy-as-code enforcement	No	No	No	Central	Central	Central	Yes (Layer 3)
Multi-dimensional lineage-quality scoring	No	No	No	Partial	No	No	Yes (Layer 4)
Continuous compliance-evidence automation	No	No	No	Add-on	Partial	No	Yes (Layer 5)

A recurring pattern in Table 1 is that governance features cluster at two extremes: general-purpose catalogs such as DataHub and OpenMetadata support broad, engine-native lineage capture but leave policy enforcement centralized or absent, while governance-first platforms such as Collibra support enforcement but lag on automated, engine-native capture; no platform in the comparison set combines both without additional integration work.

3. Regulatory and Architectural Requirements

3.1 Regulatory Drivers

BCBS 239 comprises 14 principles, of which 11 apply to banks and 3 to supervisors. Its bank-facing principles require attribute-level data lineage for all material risk data flows and aggregate risk reporting, with reporting achievable within 24–48 hours in stressed scenarios (Basel Committee on Banking Supervision, 2013). Under Principle 3 (Data Architecture and IT Infrastructure), institutions are required to maintain a risk data dictionary and data lineage through to the originating source systems, the individual stages, and the final risk data aggregation reports. GDPR Article 30 mandates keeping a record of processing activities for all personal data and end-to-end data subject tracking across the enterprise data ecosystem (European Parliament and Council, 2016). SOX requires auditable, tamper-evident transformation records for financial data. DORA mandates that financial services firms be able to reconstruct data flows during incident response, specifying operational objectives on lineage completeness and freshness (European Parliament and Council, 2022).

3.2 Architectural Drivers

Combining Data Mesh domain ownership (Dehghani, 2022), Lakehouse format heterogeneity via Delta Lake (Armbrust et al., 2020), Iceberg and Hudi, MLflow (Zaharia et al., 2018) for tracking ML pipelines, and streaming pipeline throughput at scale via Flink (Carbone et al., 2015) creates a data lineage problem domain which is difficult to solve by a centralized tool. Domain teams scale through autonomy by generating lineage events too quickly and in too many formats to be manually captured. These considerations motivate the five-layer model for FLMIF. The requirements are summarized as R1 heterogeneous auto-capture, R2 cross-domain lineage stitching, R3 federated policy enforcement, R4 quality measurement, R5 compliance-evidence automation, and R6 near-real-time freshness; these requirements are used as the evaluation criteria in Section 7.

4. Materials and Methods

This work follows the Design Science Research (DSR) paradigm (Hevner et al., 2004), in which the principal research output is a designed artifact—here, the FLMIF reference architecture—together with an evaluation of its fitness against explicitly stated requirements. The method comprised four activities. (1) Problem identification and requirements elicitation: regulatory texts (BCBS 239, GDPR, SOX, DORA, EU AI Act) and the Data Mesh/Lakehouse literature were analyzed to derive the requirements R1–R6 in Section 3. (2) Artifact design: the five-layer architecture, the capture patterns (Table 2), the inference capabilities (Table 3), the quality-scoring model (Section 5.4), and the regulatory mapping (Table 5) were designed to satisfy R1–R6. (3) Evaluation: a scenario-based architecture evaluation in the style of the Architecture Tradeoff Analysis Method (ATAM) (Kazman et al., 2000) was performed, in which each requirement is treated as a quality scenario and the architecture is assessed for whether—and with what residual risk—it satisfies the scenario (Section 7.1). (4) Projection modelling: because no production deployment data is yet available, the expected operational benefits are expressed as a transparent, assumption-based projection model (Section 7.2) rather than as measured results.

Scope and validity: the evaluation in this paper is analytical and illustrative. It does not include production telemetry, a controlled experiment, or a user study; consequently, the quantitative figures are projections that depend on those assumptions and should be read as hypotheses to be tested, not as empirical findings. Threats to the validity of this approach are discussed in Section 9.2.

5. Federated Lineage and Metadata Intelligence Framework (FLMIF)

FLMIF operates on a five-layer architecture, founded on the principles of distributed capture, centralized intelligence, and federated governance. Domain teams inject lightweight agents into their pipelines to emit standardized lineage events. Events are captured by a central metadata system, improved and reconciled into enterprise lineage graphs, while global policies are turned into enforcement rules at the domain level and enforced without team involvement.

5.1 Layer 1—Domain-Centric Lineage Capture Agents

Lineage Capture Agents are lightweight, declarative modules embedded within domain pipelines that automatically capture full lineage metadata with minimal effort from the domain team, by listening into the internals of the execution engine: Spark QueryExecutionListeners, Flink interceptors, Trino query log parsers, and dbt manifest artifacts. All lineage data is serialized in OpenLineage JSON Schema and sent to the metadata ingestion service via event streaming (Kafka (Kreps et al., 2011), Kinesis) or REST API (OpenLineage Contributors, 2021). For ML pipelines, lineage agents can capture feature engineering, model training dataset versions, and deployment metadata for the training data provenance required by the EU AI Act Article 10 for high-risk financial AI systems (European Parliament and Council, 2024).

The following illustrates a simplified OpenLineage event for a financial reconciliation pipeline:

```
{ "eventType": "COMPLETE", "job": { "namespace": "domain-finance",
  "name": "reconciliation-pipeline" },
  "inputs": [ { "namespace": "db-prod", "name": "transactions" } ],
  "outputs": [ { "namespace": "warehouse-gold",
    "name": "reconciled_transactions",
    "facets": { "columnLineage": { "amount_reconciled": {
      "inputFields": [ { "field": "transaction_amount" } ],
      "transformationType": "AGGREGATION" } } } } ] }
```

Table 2. Lineage Capture Patterns by Execution Engine.

Execution Engine	Capture Mechanism	Lineage Granularity	OpenLineage Support	Financial Use Case
Apache Spark	Spark QueryExecutionListener; SparkListener	Table + Column level	Native via Marquez	Risk aggregation, batch ETL
Apache Flink	Flink interceptors; JobStatusListener	Table + Column level	Via custom facets	Streaming fraud detection, CEP

Execution Engine	Capture Mechanism	Lineage Granularity	OpenLineage Support	Financial Use Case
Trino / Presto	Query log interception; plugin API	Table level (column via parsing)	Community connector	Ad-hoc regulatory queries
dbt	dbt Cloud / Core artifacts; manifest.json	Column level (native)	dbt-ol integration	Data transformation lineage
Apache Airflow	OpenLineage Airflow plugin	Task + dataset level	Native OpenLineage plugin	Orchestration workflow lineage
SQL Engines (JDBC)	Query log parsing; bytecode analysis	Table + Column level	Custom adapter	Legacy core banking ETL

5.2 Layer 2—Global Metadata Intelligence Layer

The Global Metadata Intelligence Layer is the cloud-native metadata platform that collects and improves all metadata with lineage reconciliation across domain boundaries while enforcing domain ownership. The Metadata Unification Engine uses OpenLineage ingestion APIs to collect lineage metadata from the domain agents, stores them in a graph database (Neo4j (Webber, 2012) or equivalent), and allows efficient navigation upstream and downstream across domain boundaries (Sawadogo & Darmont, 2021). Conflict resolution applies source-of-truth hierarchies and timestamp ordering to eliminate conflicts when a dataset is described by multiple domain agents.

ML-based metadata inference is used for PII and sensitivity detection from NLP classifiers trained with column names and sample values, dataset similarity matching across domains from graph embeddings, and usage intelligence from query log predictions for business and consumer dependencies. The Lineage Stitching Engine combines lineage across domains to create enterprise lineage paths. It also identifies broken lineage links and makes suggestions to join upstream and downstream columns based on names, schema properties, and usage analytics (Halevy et al., 2006). The governance implications of automated mis-classification are discussed in Section 9.1.

Table 3. ML-Based Metadata Inference Capabilities.

Inference Capability	Technique	Output	Regulatory Value
PII / Sensitive Data Detection	Pre-trained NLP classifier on column names + sample values	Sensitivity classification tag	GDPR Art. 30 data mapping (European Parliament and Council, 2016); BCBS 239 data classification (Basel Committee on Banking Supervision, 2013)
Dataset Similarity Matching	Graph embeddings (Node2Vec) on schema + usage patterns	Related dataset recommendations	Cross-domain lineage stitching
Business Metadata Enrichment	NLP on technical metadata; business glossary mapping	Human-readable business terms	Data product discoverability
Freshness Risk Prediction	Time-series model on lineage event intervals	Freshness risk score per dataset	BCBS 239 timeliness principle (Basel Committee on Banking Supervision, 2013)
Lineage Gap Detection	Graph traversal with heuristic link suggestion	Missing lineage link candidates	Lineage completeness scoring
Usage Intelligence	Query log analysis; consumer dependency graph	Consumer dependency map	Impact assessment for schema changes

5.3 Layer 3—Federated Governance Control Plane

The Federated Governance Control Plane provides enterprise guardrails without becoming a central bottleneck. Policies written in policy-as-code languages (Open Policy Agent's Rego or Amazon Cedar) at the global level are compiled into domain-level rules that execute as lightweight sidecars within domain pipelines (Dolhopolov et al., 2024; Wider et al., 2023). This federation model corresponds to the Data Mesh principle of federated computational governance (global standards, domain execution) (Dehghani, 2022). Global glossaries extended with domain glossaries enable semantic definitions of business terms across multiple domains.

Schema evolution rules specify backward/forward compatibility and automatic schema change migration. Retention and masking policies are automatically applied to datasets along the lineage path: if personal data exists in one dataset, any downstream dataset must adhere to GDPR regulations.

5.4 Layer 4—Lineage Quality and Completeness Scoring Engine

To aid in measuring lineage quality, the Quality Scoring Engine uses a five-dimensional scoring model (Table 4). Let the dimensions be completeness (C), accuracy (A), freshness (F), coverage (V), and governance alignment (G), with non-negative weights w_c , w_a , w_f , w_v , and w_g . The Quality Score is a weighted average of the five dimensions:

$$\text{Quality Score} = (w_c \cdot C + w_a \cdot A + w_f \cdot F + w_v \cdot V + w_g \cdot G) / (w_c + w_a + w_f + w_v + w_g)$$

For financial institutions targeting BCBS 239, default weights of $w_c = 30\%$, $w_a = 25\%$, $w_f = 20\%$, $w_v = 15\%$, and $w_g = 10\%$ are proposed, reflecting the regulatory primacy of completeness and accuracy; these weights are configurable and their effect should be examined through sensitivity analysis for each institution; a sensitivity analysis on the at-scale results is provided in Section 7.4. When the score falls below a configurable threshold (e.g., 85%), the engine generates an alert and recommends remediation. Historical trending makes it possible to identify degradation patterns before they become findings. The dimension definitions draw on production-readiness rubrics for data and ML systems (Breck et al., 2017).

Table 4. Lineage Quality Scoring Dimensions, Metrics, and BCBS 239 Alignment.

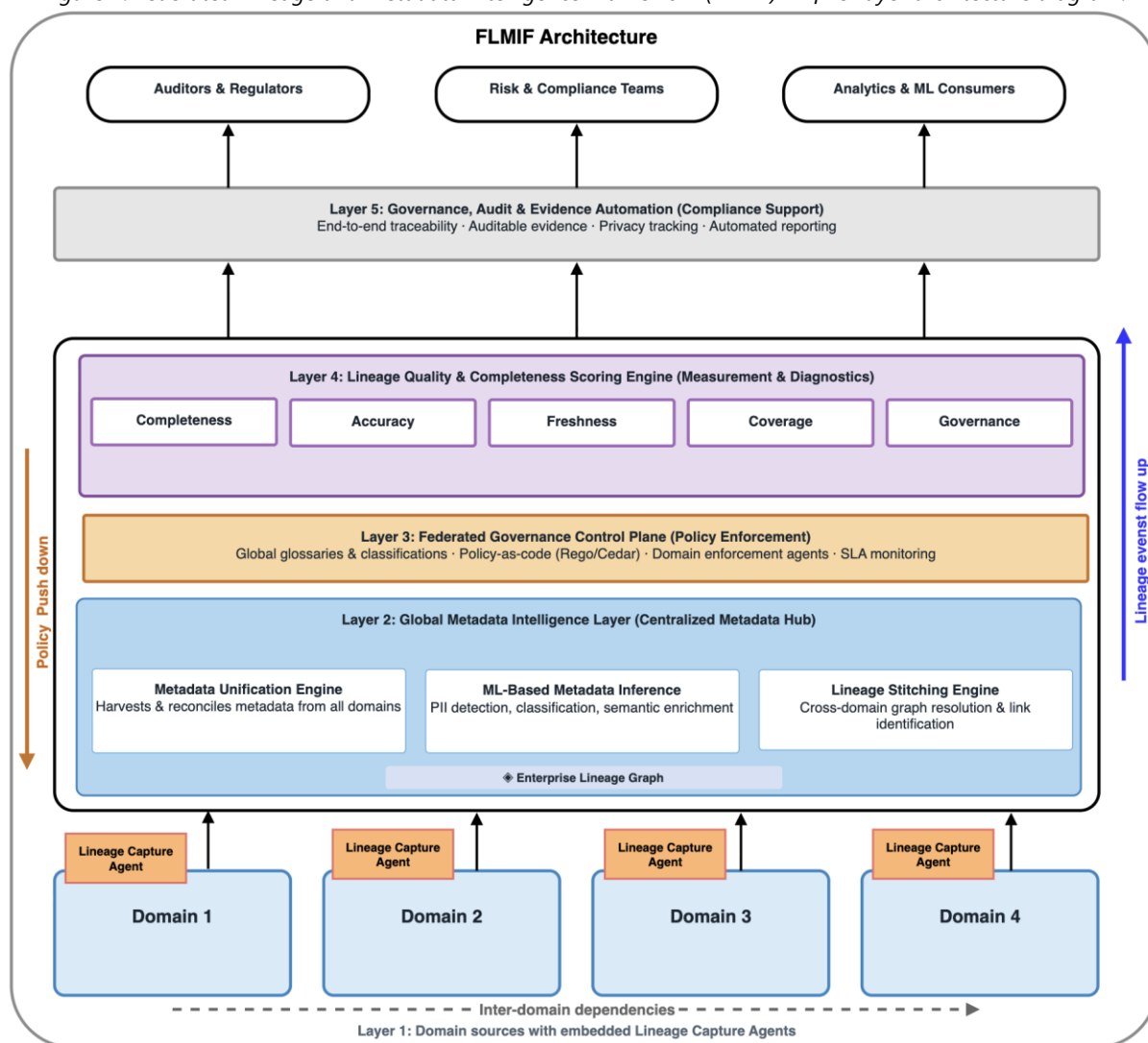
Dimension	Definition	Measurement Formula	BCBS 239 Principle	Weight (Illustrative)
Completeness	% of data flows with fully resolved lineage	$\text{resolved_flows} / \text{total_flows} \times 100$	Principle 3 — Data Architecture (Basel Committee on Banking Supervision, 2013)	30%
Accuracy	Correctness of transformation metadata	$\text{validated_transformations} / \text{total_transformations} \times 100$	Principle 6 — Accuracy (Basel Committee on Banking Supervision, 2013)	25%
Freshness	Time since last lineage update	$\max(100 - (\text{hours_stale} / \text{threshold}) \times 100, 0)$	Principle 5 — Timeliness (Basel Committee on Banking Supervision, 2013)	20%
Coverage	Depth (column-level) and breadth (cross-domain)	$\text{column_level_paths} / \text{total_paths} \times 100$	Principle 3 — Completeness (Basel Committee on Banking Supervision, 2013)	15%
Governance Alignment	Retention/masking policy propagation correctness	$\text{compliant_assets} / \text{total_assets} \times 100$	Principle 2 — Data Governance (Basel Committee on Banking Supervision, 2013)	10%

5.5 Layer 5—Governance, Audit, and Evidence Automation

Layer 5 implements lineage metadata to satisfy compliance and audit obligations. End-to-end tracing from source systems, through all transformations, to regulatory reports is implemented to satisfy regulatory requirements (e.g., BCBS 239) for demonstrable data provenance to the attribute level (Basel Committee on Banking Supervision, 2013). Transformations are versioned. Execution logs are append-only with cryptographic hash links, satisfying the SOX-required immutable audit chain. For GDPR data subject tracking, all systems storing personal data are identified, and automatic impact assessments for data subject requests are generated, such as those arising from the right to erasure (European Parliament and Council, 2016). Scheduled regulatory reporting produces evidence packages for BCBS 239 and SOX controls, and integrations with GRC tools provide continuous compliance dashboards that surface violations as they arise rather than at periodic review cycles.

Table 5. Regulatory Framework Mapping—FLMIF Layer Coverage and Automation.

Regulatory Framework	Key Lineage Requirement	FLMIF Layer	Automation Capability
BCBS 239 (Basel Committee on Banking Supervision, 2013)	Attribute-level lineage for all material risk data flows; 24–48 hr stress test SLA	Layer 1 (capture) + Layer 5 (evidence)	Automated evidence report generation; completeness scoring
GDPR (European Parliament and Council, 2016)	End-to-end data subject tracking; purpose limitation; right to erasure impact	Layer 3 (governance) + Layer 5 (audit)	Automated data subject impact assessment; sensitive field propagation tracking
SOX	Auditable financial data transformations; continuous control monitoring	Layer 5 (audit)	Tamper-evident transformation logs; cryptographically sealed audit chains
DORA (European Parliament and Council, 2022)	Operational resilience; ability to reconstruct data flows during incident response	Layer 2 (intelligence) + Layer 4 (scoring)	Near-real-time lineage for incident root cause analysis
EU AI Act (European Parliament and Council, 2024)	Training data provenance for high-risk AI systems; feature lineage	Layer 1 (ML pipeline capture) + Layer 5	ML model lineage; training dataset version tracking via MLflow (Zaharia et al., 2018)

Figure 1. Federated Lineage and Metadata Intelligence Framework (FLMIF) — five-layer architecture diagram.

An important pattern in Table 5 is that no single layer satisfies more than two regulatory drivers unassisted; sustained compliance depends on the layers acting together, which is the central architectural argument for treating governance as a cross-layer concern rather than a bolt-on control.

6. Implementation Workflow

FLMIF implements a five-step workflow, with representative technology choices per layer summarized in Table 6. Step 1—Domain Pipeline Instrumentation: lineage capture agents are provisioned with infrastructure-as-code tools, configured using declarative manifests stored in Git repositories, and lineage emission is triggered with execution engine hooks. Step 2—Event-Driven Metadata Ingestion: lineage events serialized with OpenLineage schemas are published to the ingestion service via message bus; idempotent ingestion with exactly-once semantics prevents double-insertions of lineage for re-executed pipelines (Carbone et al., 2015). Step 3—Global Intelligence Processing: lineage relationships are persisted in a graph database, and an ML inference pipeline classifies and improves lineage; conflicts are resolved by source-of-truth hierarchies. Step 4—Policy Enforcement and Validation: governance policies are compiled into executable rules, policies are evaluated during ingestion, non-compliant transformations are blocked automatically, and policy gates are added as part of CI/CD pipelines (Wider et al., 2023). Step 5—Consumption Interfaces: a GraphQL (Hartig & Pérez, 2018) API for flexible lineage queries, interactive web interface for visualizing lineage, and webhooks integration for CI/CD and incident management workflows.

Table 6. Technology Stack Recommendations by FLMIF Layer.

Layer	Component	Technology Options	Deployment Pattern
Layer 1 — Capture	Lineage capture agents	OpenLineage + Marquez; custom Spark/Flink listeners	Embedded in domain pipelines via IaC; declarative YAML config
Layer 2 — Intelligence	Metadata graph store	Neo4j; Amazon Neptune; JanusGraph	Centralized cloud-native; CDC-based incremental updates
Layer 2 — Intelligence	Search and discovery	Elasticsearch; Apache Atlas	Federated index with domain-level write access
Layer 2 — Intelligence	ML inference	scikit-learn; XGBoost; spaCy / BERT NLP	Batch inference pipeline; near-real-time for freshness scoring
Layer 3 — Governance	Policy-as-code engine	Open Policy Agent (Rego); Cedar	Sidecar deployment in domain pipelines; CI/CD policy gates
Layer 3 — Governance	Event distribution	Apache Kafka; Amazon Kinesis	Shared event bus for lineage events across domains
Layer 4 — Scoring	Quality metrics store	Prometheus + Grafana; custom dashboard	Time-series metrics; threshold-triggered alerting
Layer 5 — Audit	Evidence automation	Scheduled report generation; GRC API integration	Continuous compliance monitoring; quarterly regulatory package generation

7. Evaluation and Expected Benefits

7.1 Requirement-Coverage Analysis

Following the scenario-based method of Section 4, each requirement R1–R6 is treated as a quality scenario and assessed for the architecture's ability to satisfy it and the residual risk. Table 7 records the result. This assessment finds that FLMIF addresses all six requirements at the architectural level, with the principal residual risks concentrated in non-instrumentable legacy systems (R1) and in the accuracy of automated inference (R2, R4).

Table 7. Requirement-Coverage Analysis (Scenario-Based Architecture Evaluation).

Req.	Quality Scenario	Satisfying Layer(s)	Assessment	Residual Risk
R1	Auto-capture across heterogeneous engines	Layer 1	Satisfied for instrumentable engines	Legacy/opaque engines need manual annotation
R2	Cross-domain lineage stitching	Layer 2	Satisfied	Inference false joins; needs validation
R3	Federated policy enforcement	Layer 3	Satisfied	Policy-coverage gaps; sidecar overhead
R4	Quality measurement	Layer 4	Satisfied	Weight calibration; metric gaming

Req.	Quality Scenario	Satisfying Layer(s)	Assessment	Residual Risk
R5	Compliance-evidence automation	Layer 5	Satisfied	Auditor acceptance of automated evidence
R6	Near-real-time freshness	Layers 1–2	Satisfied for event-driven sources	Batch/legacy sources lag

7.2 Projected Benefits (Assumption-Based Model)

Because FLIMIF has not yet been deployed in production, the operational benefits below are projections derived from an explicit model, not measured results. Each projection states a baseline assumption, an FLIMIF assumption, and the resulting effect; the assumptions are deliberately conservative and should be tested empirically (Section 9.3). The baselines reflect commonly reported characteristics of centralized, manually maintained lineage programs and of compliance-oriented financial data platforms (Buneman et al., 2001; Ransbotham et al., 2019; Sawadogo & Darmont, 2021; Talati, 2025). Reference (Talati, 2025) is a single industry-oriented source and is used here only for directional context, not as evidence for any specific figure. The at-scale run reported in Section 7.4 is a buildability and soundness check: it demonstrates that the framework can be implemented and that its lineage stitching, quality scoring, and graph traversal are mathematically well-defined and stable on a graph of thousands of datasets. It is generated from synthetic data and is explicitly not a field trial; validation on production organizational telemetry remains the target of future work (Section 9.3).

Table 8. Projection Model for Expected Benefits (Illustrative Projections Based on the Stated Assumptions, Not Measured Results).

Metric	Baseline Assumption	FLIMIF Assumption	Projected Effect	Basis
Lineage coverage	~50% of flows documented under centralized manual upkeep	~70–75% via automated capture (legacy excluded)	≈40–50% relative increase	Automated capture across instrumentable engines (Layer 1)
Manual compliance-evidence effort	Periodic manual collation per reporting cycle	Continuous, auto-generated evidence for instrumented domains	≈60–70% reduction	Evidence automation (Layer 5)
Root-cause-analysis time	Manual upstream/downstream tracing across siloed catalogs	On-demand graph traversal of stitched lineage	Up to ≈75% reduction	Graph query vs. manual discovery (Layers 2, 4)
Metadata freshness	Weekly/monthly reconciliation cycles	Event-driven near-real-time updates	Near-real-time	Exactly-once event ingestion (Layers 1–2)
Policy-conformance rate	Manual, inconsistent enforcement	Automated policy-as-code gates	High conformance (target >95%)	Federated enforcement (Layer 3)

An important caveat visible in Table 8 is that the largest projected gains, in compliance-evidence effort and root-cause-analysis time, are also the ones whose baselines are least standardized across institutions; this is why the sensitivity ranges reported in Section 7.2 are widest for exactly those two metrics.

7.2.1 Basis of Estimate and Sensitivity

Each value in Table 8 is derived from the baseline and target assumptions stated above. The cited sources support the baseline conditions only. They are not offered as evidence for the percentages. For lineage coverage, let B be the fraction of the flows documented under current manual practice and T be the fraction reachable with automated capture. Taking B near 0.50, consistent with how manually maintained lineage tends to decay and how central teams become bottlenecks (Buneman et al., 2001; Ransbotham et al., 2019; Sawadogo & Darmont, 2021; Sculley et al., 2015), and T near 0.70 to 0.75 (since legacy and opaque engines still require manual annotation per residual risk R1), then the relative gain $(T - B)/B$ is approximately 40 to 50%. A starting point of 0.60 lowers it to roughly 17 to 25%, and a starting point of 0.40 raises it to roughly 75 to 88%, so each institution should recompute using its own measured baseline.

For compliance-evidence effort, multiplying the instrumented coverage (approximately 0.72) by the fraction of effort spent on manual collation and tracing (approximately 0.85 to 0.90) removes roughly 60 to 70% of the manual work, leaving the human judgment that automation does not replace. For root cause analysis, if manual discovery across separate catalogs accounts for 75

to 80% of total effort and FLIMIF replaces it with an on-demand graph query, then the reduction approaches 75%, consistent with how provenance has long been used for this kind of debugging (Herschel et al., 2017). The figure of greater than 95% policy conformance is an enforcement target for the Layer 3 policy gates, not a predicted outcome. All of these remain projections to be tested against production data.

7.3 Illustrative Worked Case Study: BCBS 239 Reconciliation Pipeline

Application of the framework is demonstrated through a representative financial reconciliation pipeline (the Layer 1 example in Section 5.1) feeding an aggregate risk report. Under the centralized baseline, an analyst preparing BCBS 239 evidence must manually trace the report's figures back through transformations to source systems across several catalogs—an effort measured in analyst-days per reporting cycle, with coverage limited to the flows that happen to be documented. Under FLIMIF, the reconciliation pipeline emits OpenLineage events capturing column-level lineage (e.g., `amount_reconciled ← transaction_amount, AGGREGATION`); Layer 2 stitches these into an enterprise path; Layer 4 scores the path's completeness, accuracy, freshness, coverage, and governance; and Layer 5 emits a ready-to-file evidence package. The same root-cause query that previously required manual discovery becomes a single graph traversal. The figures in Table 8 are the projected outcome of this mechanism; they are presented as hypotheses for empirical validation, not as observed results.

7.4 Proof-of-Concept Implementation and Feasibility Results

To show that the framework is buildable and produces the artifacts described above, a small proof-of-concept was implemented that runs the full capture-to-evidence path on synthetic data. The implementation is dependency-free and reproducible; a minimal in-memory directed-graph module (approximately 80 lines of standard-library Python) stands in for a production graph database such as Neo4j or Amazon Neptune, and the domain agents are synthetic emitters that produce real OpenLineage events. The proof-of-concept is a feasibility demonstrator, not an enterprise-scale validation: it confirms that the mechanism works and yields measurable lineage, scores, policy decisions, and root-cause traces, but it does not reproduce the enterprise-scale magnitudes projected in Section 7.2.

7.4.1 Setup

Four producing domains (finance, risk, retail, and core-banking) emit four OpenLineage events from synthetic agents that emulate the output of Spark, dbt, and Airflow integrations. The events are ingested into a lineage graph of seven datasets and four edges. One dataset, the reconciled ledger, is produced in the finance domain and consumed in the risk domain, providing a cross-domain handoff that Layer 2 stitches into a single enterprise path. One flow is left deliberately incomplete, a legacy ETL job whose upstream source was never instrumented so that the quality and coverage metrics have a real gap to detect.

7.4.2 Environment

All proof-of-concept results were produced in a Linux virtual machine (Ubuntu 22.04.5 LTS, Linux 6.8 kernel, ARM64) allocated 4 vCPUs and approximately 4 GB of RAM, hosted on Apple-Silicon hardware, under Python 3.10.12. The core implementation uses only the Python standard library (`json`, `time`, `random`, `statistics`, `datetime`, `os`); `matplotlib` 3.10.9 is used only to render Figure 2. Results are deterministic under a fixed random seed (42), and each latency measurement is the median of five repetitions. Absolute latencies are hardware-dependent, so the relevant result is the difference in how the two approaches scale with graph size rather than the absolute values in milliseconds.

7.4.3 Results

The policy-as-code rule for PII propagation flagged four governance violations, namely datasets that contain or inherit a personally identifiable field but carry no masking policy. The quality engine computed an overall enterprise quality score of 82.0% under the BCBS 239 default weights (completeness 100%, accuracy 75%, freshness 100%, coverage 75%, governance 20%), which falls below the 85% alert threshold and triggers a remediation alert, driven mainly by the governance gap. Layer 5 generated a complete source-to-report provenance path for the material risk report and returned its upstream contributors on demand. These outputs are summarized in Table 9.

Table 9. Measured Results from the Proof-of-Concept Run, Mapped to the FLIMIF Layer That Produces Each Result (Synthetic Data).

Aspect	FLIMIF Layer	Measured Result
Events captured	Layer 1	4 OpenLineage events across 4 domains (finance, risk, retail, core-banking)
Lineage graph	Layer 2	7 datasets, 4 edges
Cross-domain handoffs stitched	Layer 2	1 (reconciled ledger: finance → risk)
PII policy violations detected	Layer 3	4
Overall quality score (enterprise)	Layer 4	82.0% (below the 85% alert threshold)

Aspect	FLMIF Layer	Measured Result
Enterprise quality dimensions	Layer 4	Completeness 100, Accuracy 75, Freshness 100, Coverage 75, Governance 20
Quality score (per domain)	Layer 4	Finance 100.0%; Risk 90.0%; Retail 90.0%; Core-banking 60.0%
Evidence pack	Layer 5	1 end-to-end source→report provenance path
RCA upstream query	Layer 5 (via Layer 2)	2 upstream contributors returned for the material risk report
RCA latency at 10,000 datasets	Layer 2	≈0.015 ms (FLMIF graph traversal) vs ≈68 ms (naive event scan)

7.4.4 How the 82.0% Score Is Derived

Each of the five dimensions is a simple ratio measured directly on the graph, and the overall score is their weighted average using the BCBS 239 weights from Section 5.4. Completeness is the share of derived datasets that have at least one resolved upstream link; all four derived datasets qualify, so completeness is 100. Accuracy is the share of flows that carry a recorded transformation; three of the four edges do, so accuracy is 75. Freshness is 100 because the events form a single recent batch. Coverage is the share of derived datasets that have column-level lineage; three of four, so 75. Governance is the share of datasets carrying or inheriting personal data that satisfy the masking policy; only the reconciled ledger does, so governance is 1 of 5, or 20. Applying the formula: $[0.30(100) + 0.25(75) + 0.20(100) + 0.15(75) + 0.10(20)] / 1.00 = 82.0\%$. The result falls below the 85% threshold and raises an alert. The governance dimension is what pulls the score down—the intended signal that the graph still contains unmasked personal data.

7.4.5 Enterprise Versus Per-Domain Scores

The 82.0% figure is the enterprise roll-up, computed over the whole graph. The same engine also scores each domain on the data products it owns, enabling domain accountability through data-product scorecards as described in the Section 8 roadmap. Table 10 shows the per-domain result. Finance scores 100% because its reconciled ledger has full column lineage and a masking policy. Retail and risk both score 90%, each with governance at 0% because their personal-data products are unmasked. The core-banking domain scores 60%: its legacy report has no column lineage, so accuracy and coverage are 0%, although governance is 100% because that report carries no personal data. The enterprise score is not a simple average of these four: it also covers the raw ingested sources, which belong to no producing domain yet still carry unmasked personal data, so the enterprise governance dimension is lower than any single domain would suggest—a useful property that surfaces ungoverned ingestion a purely per-domain view would miss.

Table 10. Per-Domain Quality Scores (Each Domain Scored on the Data Products It Owns) and the Enterprise Roll-Up.

Domain	Quality Score (%)	Completeness	Accuracy	Freshness	Coverage	Governance
finance-domain	100.0%	100	100	100	100	100
risk-domain	90.0%	100	100	100	100	0
retail-domain	90.0%	100	100	100	100	0
core-banking-domain	60.0%	100	0	100	0	100
Enterprise (roll-up)	82.0%	100	75	100	75	20

7.4.6 At-Scale Feasibility

The same implementation was then run on a larger synthetic estate, generated by the identical code path, to confirm that the mechanism scales. Ten producing domains now emit 2,500 OpenLineage events, which Layer 2 ingests into a graph of 2,532 datasets and 4,937 edges in approximately 16 ms. Inputs are drawn mostly from a dataset's own domain and, with a fixed probability, from other domains, so the estate contains genuine cross-domain dependencies; personally identifiable fields, column-level lineage, masking policies, and update recency are assigned per dataset from per-domain instrumentation-maturity parameters configured as model inputs rather than measured findings, so the quality dimensions remain non-degenerate at scale. Concretely, masking, column-level lineage, and refresh are each assigned per dataset with probability equal to the domain's maturity (0.62 for legacy core-banking to 0.95 for the most mature, mean approximately 0.85). About 85% of personal-data products are therefore masked, giving enterprise governance approximately 84%; the illustrative example masks only one of its five such datasets, hence its 20%. Layer 2 stitched 752 cross-domain handoffs; Layer 3 flagged 396 datasets carrying or inheriting personal data without a masking policy; and Layer 4 computed an overall enterprise quality score of 88.5% (completeness 100%,

accuracy 84.9%, freshness 80.2%, coverage 85.0%, governance 84.3%). The per-domain scores range from about 75% for the legacy core-banking domain to about 95% for the most mature domain (Figure 2), so the engine localizes the enterprise's weakest lineage to a specific owner rather than reporting only a blended figure. Table 11 summarizes the results.

Table 11. At-scale proof-of-concept results (synthetic estate of 2,500 events across 10 domains; deterministic seed).

Aspect	FLMIF Layer	Measured Result
Producing domains	Layer 1	10
OpenLineage events	Layer 1	2,500
Lineage graph	Layer 2	2,532 datasets, 4,937 edges (built in ~16 ms)
Cross-domain handoffs stitched	Layer 2	752
PII policy violations detected	Layer 3	396
Enterprise quality score	Layer 4	88.5% (C 100, A 84.9, F 80.2, V 85.0, G 84.3)
Per-domain quality range	Layer 4	≈75% (legacy core-banking) to ≈95% (most mature domain)
RCA latency at 50,000 datasets	Layer 2	≈0.05 ms (FLMIF traversal) vs ≈1,696 ms (naive event scan)

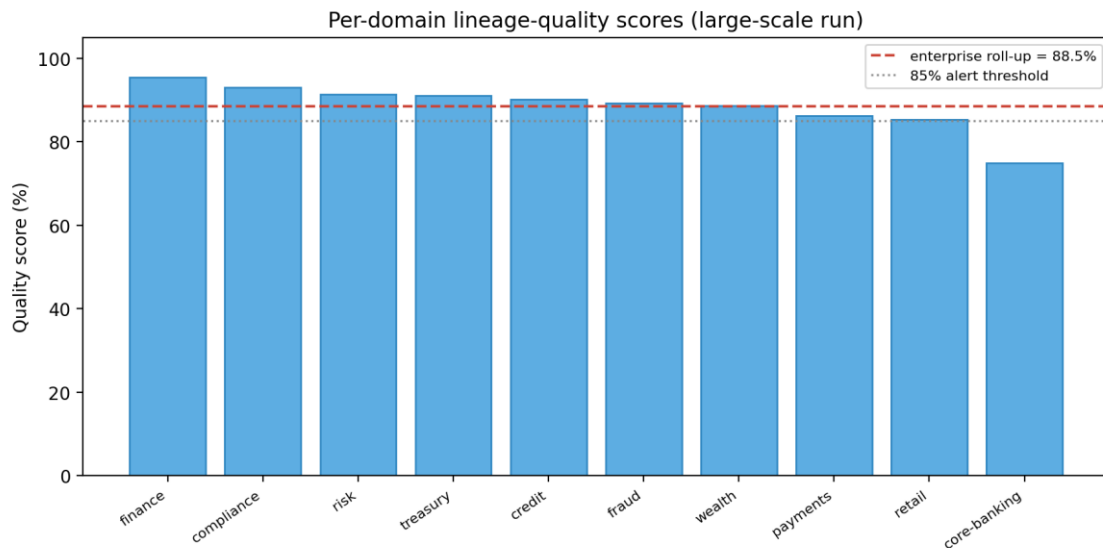


Figure 2. lineage-quality scores per domain for the at-scale run (10 domains), with the enterprise roll-up and the 85% alert threshold marked.

7.4.7 Root-Cause-Analysis Latency

The root-cause-analysis latency benchmark was extended to 50,000 datasets (Figure 3). A root-cause query returns every dataset upstream of a given one. The FLMIF graph traversal remained approximately constant (about 0.005 to 0.05 ms across the full range), while a naive baseline that rebuilds ancestry by scanning the event log on each step grew super-linearly, from about 0.9 ms at 500 datasets to about 1,696 ms at 50,000. The relative reduction exceeds 99% from 500 datasets onward. As before, the absolute timings are hardware-dependent; the result is the divergence in scaling, which is precisely the regime in which manual, non-indexed root-cause analysis becomes impractical and an indexed lineage graph does not.

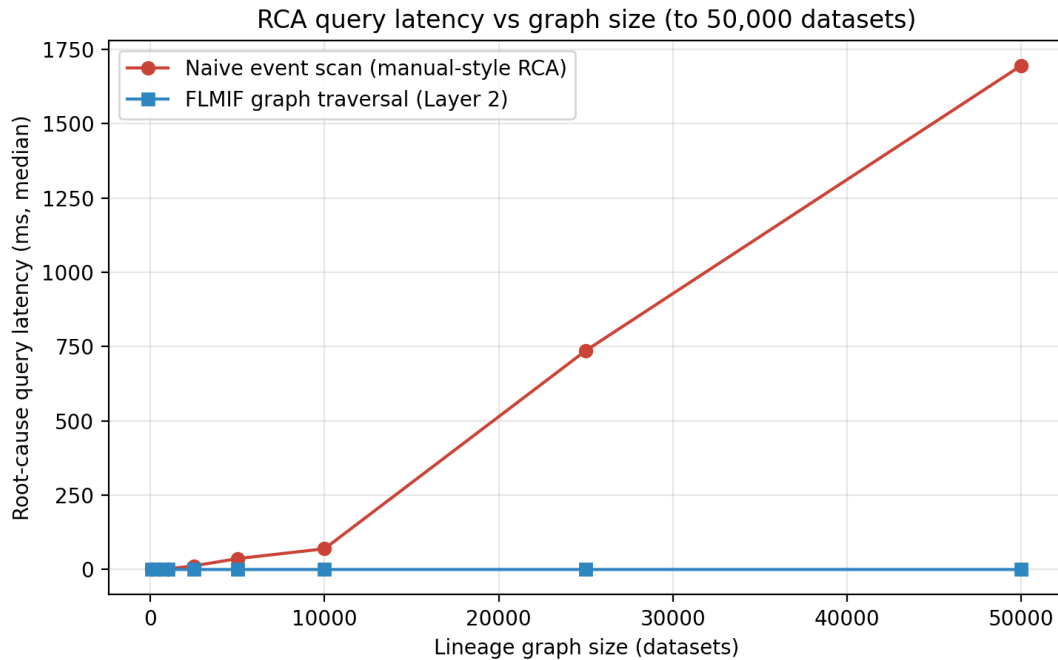


Figure 3. Root-cause-analysis query latency versus lineage-graph size (to 50,000 datasets): FLMIF graph traversal versus a naive event-scan baseline (median of repeated runs; proof-of-concept scale).

This at-scale run does not alter the framing of Section 7.2: the operational magnitudes for a real enterprise remain projections to be validated in production. What it establishes is that the mechanism is buildable and behaves correctly at a realistic graph size, with thousands of datasets, hundreds of cross-domain handoffs, per-domain quality diagnostics, and root-cause queries that stay near-constant as the estate grows.

7.4.8 Weight Sensitivity

The overall score is robust to the choice of weights. Holding the at-scale dimension values from Table 11 fixed (base score 88.5%), each weight was varied by plus or minus 10 percentage points and the remaining weights renormalized. Table 12 reports the resulting scores. Across all ten perturbations the overall score stays within 87.2% to 89.5%, a movement of at most about 1.3 points, so the conclusion does not depend on the exact weighting. Completeness has the largest influence because it is the highest scoring dimension.

Table 12. Sensitivity of the overall quality score to plus or minus 10 percentage-point changes in each dimension weight (at-scale dimension values; base score 88.5%; remaining weights renormalized; at the minus 10 pp point the governance weight becomes 0).

Dimension weight (base)	Score at – 10 pp (%)	Score at +10 pp (%)
Completeness (30%)	87.2	89.5
Accuracy (25%)	88.8	88.1
Freshness (20%)	89.4	87.7
Coverage (15%)	88.8	88.1
Governance (10%)	88.9	88.1

8. Implementation Roadmap

Phase 1—Foundation (Months 1–3): Build a core metadata platform (e.g., Neo4j or an equivalent graph store), governance taxonomies, policy templates, instrument 2–3 pilot domains with lineage capture agents, and form a cross-domain governance council. Phase 2—Federation (Months 4–6): Deploy domain-level capture agents for all priority domains, stitch domain boundaries and term maps, deploy first ML-based metadata inference systems, and publish data product scorecards for domain teams. Phase 3—Automation (Months 7–12): Operationalize ML-driven metadata enrichment, policy-as-code enforcement, lineage in CI/CD pipelines, and continuous data contract validation (Dolhopolov et al., 2024). Phase 4—Compliance Integration (Months 13–18): Automate BCBS 239 compliance evidence generation and reporting, end-to-end SOX audit workflows, ML model lineage and AI

Act governance. Achieve continuous compliance posture with real-time GRC dashboards (European Parliament and Council, 2024; Talati, 2025).

9. Discussion, Limitations, and Future Work

9.1 Limitations of Automated ML Inference

FLMIF relies on ML inference for PII detection, dataset matching, and gap suggestion. Such inference is imperfect: PII classifiers produce false positives (over-restricting access and propagating unnecessary masking) and false negatives (leaving sensitive fields untagged, a compliance risk), and models drift as schemas and naming conventions evolve. Automated classification can also encode bias—for example, systematically under-detecting sensitivity in certain domains—with downstream governance consequences; the fairness and bias literature for ML systems is directly relevant here (Mehrabi et al., 2021). ML outputs should therefore be treated as recommendations subject to human-in-the-loop validation, inference decisions should be logged for auditability, and periodic re-training and threshold calibration are recommended. Inference quality should itself be tracked as a production-readiness concern (Breck et al., 2017).

This risk is addressed architecturally rather than left to model quality alone. Layer 2's PII and sensitivity classifiers emit a confidence score alongside every classification. When that score falls below a configurable threshold, the classification is not treated as final: Layer 3's policy-as-code engine applies a conservative fallback rule and treats the field as sensitive by default, propagating the associated masking and retention policy along the lineage path regardless of the Layer 2 output. An uncertain inference therefore cannot pass through the system as "not sensitive"; the fallback biases toward over-masking rather than under-masking, which is the appropriate failure mode for a regulated institution weighing a false negative against a false positive, given that an undetected PII propagation event carries the larger regulatory exposure under GDPR Article 30 and BCBS 239 data classification obligations (Basel Committee on Banking Supervision, 2013; European Parliament and Council, 2016). Every low-confidence classification and every fallback invocation is logged with its triggering confidence score, giving data stewards a review queue rather than a black box. This places the compliance boundary in Layer 3's deterministic policy engine rather than in Layer 2's probabilistic inference, consistent with the architectural role each layer is designed to play; a recent systematic review of AI governance situates this kind of control within a broader structure spanning team, organizational, industry, and national governance layers, of which the fallback described here operates at the team and organizational levels (Batool et al., 2025).

9.2 Threats to Validity

Construct validity: the projection model relies upon assumptions of a baseline and target which differ from one institution to another; institutions must use their own baselines when deriving practical conclusions. A sensitivity analysis in Section 7.4 is only a first step toward calibration.

Internal validity: the requirement-coverage analysis is performed by the architecture's designer, which risks optimistic bias. To mitigate this, assessment criteria and residual risks were specified explicitly before evaluation — Table 7 records both the satisfaction assessment and the residual risk for each requirement, rather than reporting only positive outcomes. This approach is consistent with DSR evaluation practice as described by Hevner et al. (Hevner et al., 2004); independent replication remains the appropriate next step.

External validity: the framework is inspired by the regulated financial services industry and can be applicable poorly to any other industry. The mapping of the regulations provided in Table 5 is particular for BCBS 239, GDPR, SOX, DORA, and EU AI Act and firms that operate beyond this legal framework should map the compliance layer differently.

Conclusion validity: no statistical claims have been made since there have been no data gathered; the numbers presented are mere projections based on the underlying assumptions, and are thus to be taken as hypotheses for further verification.

9.3 Limitations and Future Work

In complex transformations such as legacy core-banking ETL, automated capture may need to be complemented or replaced by a structured manual-annotation interface. New tools and execution engines require new agents, as OpenLineage connectors do not yet cover all enterprise environments (OpenLineage Contributors, 2021). Cross-organization metadata sharing that respects privacy and security constraints remains open. The primary future work is empirical: deploying FLMIF in production at regulated financial institutions and measuring the realized effects against the projections in Table 8. Further work includes Apache Iceberg table-level metadata integration, ML-based lineage generation for non-instrumented pipelines, cross-organization lineage federation for regulated data-sharing partnerships, and alignment with evolving EU AI Act requirements for training-data provenance in high-risk financial AI systems (European Parliament and Council, 2024).

10. Conclusion

Federated Lineage and Metadata Intelligence enables large enterprises practicing Data Mesh or Lakehouse architecture within a governed environment to centralize metadata collection, intelligence, and governance via federated ownership with central guardrails. The Federated Lineage Metadata Intelligence Framework addresses a key challenge of centralized lineage methods at enterprise scale. With OpenLineage-standard automated capture (OpenLineage Contributors, 2021), ML metadata inference,

policy-as-code federated governance (Dolhopolov et al., 2024; Wider et al., 2023), multi-dimensional quality scoring aligned with BCBS 239 (Basel Committee on Banking Supervision, 2013) priorities, and continuous compliance evidence automation, FLMIF enables both local domain autonomy and global enterprise control.

Under the assumptions of the projection model in Section 7, FLMIF is expected to deliver materially higher lineage coverage, substantially lower manual compliance effort, faster root-cause analysis, and near-real-time freshness; these are projections to be validated, not measured outcomes. A synthetic proof-of-concept demonstrates that the underlying mechanisms are buildable and behave correctly at a realistic graph scale, which supports the feasibility of these projections without substituting for them. This framework provides a standards-based reference model to modernize data governance with lineage in distributed, time-critical, and regulated environments. The next most useful research would be empirical validation through production use cases.

Author Contributions

Conceptualization, R.S.; Methodology, R.S.; Software, R.S.; Validation, R.S.; Formal Analysis, R.S.; Investigation, R.S.; Resources, R.S.; Data Curation, R.S.; Writing—Original Draft Preparation, R.S.; Writing—Review and Editing, R.S.; Visualization, R.S. The author has read and agreed to the published version of the manuscript.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Synthetic data and proof-of-concept artifacts were created for the feasibility study described in Section 7.4. These materials can be made available by the author upon reasonable request; the author intends to deposit the proof-of-concept code and synthetic datasets in a public repository with a permanent identifier prior to publication.

Appendix A. Illustrative OpenLineage Event Schema

This appendix illustrates the FLMIF implementation blueprint with a standards-compliant synthetic OpenLineage event emitted by a Layer 1 domain agent and captured by the Layer 2 ingestion engine. It expands the simplified event shown in Section 5.1 with run, environment, and input/output schema facets. The event is synthetic and illustrative; it is not telemetry from a production deployment. The PII-sensitivity annotation on the account_number field illustrates the input to the Layer 2 ML-based PII-detection capability (Section 5.2), and the column-level lineage facet (reconciled_amount ← amount) illustrates the attribute-level provenance required for BCBS 239 evidence (Section 5.5).

```
{
  "$schema": "https://openlineage.io/spec/2-0-2/OpenLineage.json",
  "eventType": "COMPLETE",
  "eventTime": "2026-06-19T23:15:00.000Z",
  "run": {
    "runId": "d4b3e2a1-c0a8-4321-9876-fedcba987654",
    "facets": {
      "environment": { "environment": "production" },
      "nominalTime": { "nominalStartTime": "2026-06-19T23:00:00.000Z" }
    }
  },
  "job": {
    "namespace": "enterprise-finance-domain",
    "name": "ledger_reconciliation_pipeline"
  },
  "inputs": [{
    "namespace": "aws_s3_raw_zone",
    "name": "daily_transactions_csv",
    "facets": {
      "schema": {
        "fields": [
          { "name": "transaction_id", "type": "VARCHAR" },
          { "name": "account_number", "type": "VARCHAR", "description": "PII-Sensitive" },
          { "name": "amount", "type": "DECIMAL" }
        ]
      }
    }
  }]
}
```

```

    }
  }
},
"outputs": [{
  "namespace": "snowflake_gold_layer",
  "name": "reconciled_ledger_reporting",
  "facets": {
    "columnLineage": {
      "fields": {
        "reconciled_amount": {
          "inputFields": [{
            "namespace": "aws_s3_raw_zone",
            "name": "daily_transactions_csv",
            "field": "amount"
          }],
          "transformationType": "AGGREGATION"
        }
      }
    }
  }
}]
}

```

Listing A1. Synthetic, schema-compliant OpenLineage (2-0-2) event for a regulated financial reconciliation pipeline, with input schema facets (including a PII-sensitive field) and a column-level lineage facet.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

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

- Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. *Int. J. Inf. Manag.*, 49, 424–438. <https://doi.org/10.1016/j.ijinfomgt.2019.07.008>
- Apache Software Foundation. (n.d.). Apache Atlas—Data Governance and Metadata Framework for Hadoop. <https://atlas.apache.org/> (accessed on 12 June 2026).
- Armbrust, M., Das, T., Paranjpye, A., Xin, R., Zhu, S., Ghodsi, A., Yavuz, S., Murthy, S., & Zaharia, M. (2020). Delta Lake: High-performance ACID table storage over cloud object stores. *Proc. VLDB Endow.*, 13, 3411–3424. <https://doi.org/10.14778/3415478.3415560>
- Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of the 11th Annual Conference on Innovative Data Systems Research (CIDR)*, Chaminade, CA, USA, 11–15 January 2021. https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf (accessed on 12 June 2026).
- Basel Committee on Banking Supervision. (2013). BCBS 239: Principles for effective risk data aggregation and risk reporting. Bank for International Settlements. <https://www.bis.org/publ/bcbs239.pdf>
- Batool, A., Zowghi, D., & Bano, M. (2025). AI governance: A systematic literature review. *AI Ethics*, 5, 3265–3279. <https://doi.org/10.1007/s43681-024-00653-w>
- Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. In *Proceedings of the IEEE International Conference on Big Data*, Boston, MA, USA, 11–14 December 2017; pp. 1123–1132. <https://doi.org/10.1109/BigData.2017.8258038>
- Buneman, P., Khanna, S., & Tan, W.-C. (2001). Why and where: A characterization of data provenance. In *Proceedings of the 8th International Conference on Database Theory (ICDT)*, London, UK, 4–6 January 2001; pp. 316–330. https://doi.org/10.1007/3-540-44503-X_20
- Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink: Stream and batch processing in a single engine. *IEEE Data Eng. Bull.*, 38, 28–38. <http://sites.computer.org/debull/A15dec/p28.pdf> (accessed on 12 June 2026).

- Collibra. (n.d.). Collibra Data Lineage. <https://www.collibra.com/products/data-lineage> (accessed on 12 June 2026).
- Databricks. (n.d.). Data Lineage in Unity Catalog. <https://docs.databricks.com/aws/en/data-governance/unity-catalog/data-lineage> (accessed on 12 June 2026).
- DataHub Project (LF AI & Data). (n.d.). About DataHub Lineage. <https://docs.datahub.com/docs/features/feature-guides/lineage> (accessed on 12 June 2026).
- Dehghani, Z. (2022). *Data mesh: Delivering data-driven value at scale*. O'Reilly Media. ISBN 978-1492092391.
- Dolhopolov, A., Castelltort, A., & Laurent, A. (2024). Implementing federated governance in data mesh architecture. *Future Internet*, 16, 115. <https://doi.org/10.3390/fi16040115>
- European Parliament and Council. (2016). Regulation (EU) 2016/679—General Data Protection Regulation (GDPR). *Official Journal of the European Union*, L 119, 4 May 2016, pp. 1–88. <https://gdpr-info.eu/>
- European Parliament and Council. (2022). Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA). *Official Journal of the European Union*, L 333, 27 December 2022, pp. 1–79. <https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng>
- European Parliament and Council. (2024). Regulation (EU) 2024/1689—Artificial Intelligence Act. *Official Journal of the European Union*, L, 2024/1689, 12 July 2024. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689>
- Fragkoulis, M., Carbone, P., Kalavri, V., & Katsifodimos, A. (2023). A survey on the evolution of stream processing systems. *VLDB J.*, 33, 507–541. <https://doi.org/10.1007/s00778-023-00819-8>
- Goedegebuure, A., Kumara, I., Driessen, S., Van Den Heuvel, W.-J., Monsieur, G., Tamburri, D.A., & Di Nucci, D. (2024). Data mesh: A systematic gray literature review. *ACM Comput. Surv.*, 57, 1–36. <https://doi.org/10.1145/3687301>
- Halevy, A., Rajaraman, A., & Ordille, J. (2006). Data integration: The teenage years. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, Seoul, Korea, 12–15 September 2006; pp. 9–16. <https://www.vldb.org/conf/2006/p9-halevy.pdf> (accessed on 12 June 2026).
- Harby, A.A., & Zulkernine, F. (2022). From data warehouse to Lakehouse: A comparative review. In *Proceedings of the IEEE International Conference on Big Data*, Osaka, Japan, 17–20 December 2022; pp. 389–395. <https://doi.org/10.1109/BigData55660.2022.10020719>
- Hartig, O., & Pérez, J. (2018). Semantics and complexity of GraphQL. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, Lyon, France, 23–27 April 2018; pp. 1155–1164. <https://doi.org/10.1145/3178876.3186014>
- Herschel, M., Diestelkämper, R., & Ben Lahmar, H. (2017). A survey on provenance: What for? What form? What from? *VLDB J.*, 26, 881–906. <https://doi.org/10.1007/s00778-017-0486-1>
- Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Q.*, 28, 75–105. <https://doi.org/10.2307/25148625>
- Hutterer, A. (2025). The future of data management: A delimitation of data platforms, data spaces, data meshes, and data fabrics. *Inf. Syst. E-Bus. Manag.*, 23, 971–997. <https://doi.org/10.1007/s10257-025-00707-4>
- Kazman, R., Klein, M., & Clements, P. (2000). *ATAM: Method for architecture evaluation (CMU/SEI-2000-TR-004)*. Software Engineering Institute, Carnegie Mellon University.
- Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the 6th International Workshop on Networking Meets Databases (NetDB)*, Athens, Greece, 12 June 2011; pp. 1–7. <https://www.cs.princeton.edu/~jrex/publications/netdb2011.pdf> (accessed on 12 June 2026).
- Machado, I.A., Costa, C., & Santos, M.Y. (2022). Data mesh: Concepts and principles of a paradigm shift in data architectures. *Procedia Comput. Sci.*, 196, 263–271. <https://doi.org/10.1016/j.procs.2021.12.013>
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54, 1–35. <https://doi.org/10.1145/3457607>
- Microsoft. (n.d.). Data Lineage in Microsoft Purview Data Catalog. <https://learn.microsoft.com/en-us/purview/data-gov-classic-lineage> (accessed on 12 June 2026).
- OpenLineage Contributors. (2021). *OpenLineage: An open standard for lineage metadata collection [Technical specification]*. <https://openlineage.io/docs/>
- OpenMetadata. (n.d.). *OpenMetadata Documentation*. <https://docs.open-metadata.org/> (accessed on 12 June 2026).
- Ransbotham, S., Khodabandeh, S., Fehling, R., LaFountain, B., & Kiron, D. (2019). *Winning with AI*. MIT Sloan Management Review & BCG. <https://sloanreview.mit.edu/projects/winning-with-ai/>

- Sawadogo, P.N., & Darmont, J. (2021). On data lake architectures and metadata management. *J. Intell. Inf. Syst.*, 56, 97–120. <https://doi.org/10.1007/s10844-020-00608-7>
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Proceedings of the NeurIPS, Montreal, Canada, 7–12 December 2015; Volume 28*. <https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fc2674f757a2463eba-Abstract.html> (accessed on 12 June 2026).
- Sugimoto, R.H., Meirelles, P., & Braghetto, K. (2025). Metadata management in data mesh: Toward federated discovery and governance. In *Proceedings of the 40th Brazilian Symposium on Databases (SBBd), Fortaleza, Brazil, 2025*; pp. 823–829. <https://doi.org/10.5753/sbbd.2025.247722>
- Talati, D. (2025). Secure financial data lake architecture: Balancing regulatory compliance and analytics capabilities. *J. Inf. Syst. Eng. Manag.*, 10, 58s. <https://jisem-journal.com/index.php/journal/article/download/12728/5909/21410> (accessed on 12 June 2026).
- 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. *J. Syst. Softw.*, 230, 112584. <https://doi.org/10.1016/j.jss.2025.112584>
- Webber, J. (2012). A programmatic introduction to Neo4j. In *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity (SPLASH), Tucson, AZ, USA, 19–26 October 2012*; pp. 217–218. <https://doi.org/10.1145/2384716.2384777>
- Wider, A., Verma, S., & Akhtar, A. (2023). Decentralized data governance as part of a data mesh platform: Concepts and approaches. In *Proceedings of the IEEE International Conference on Web Services (ICWS), Chicago, IL, USA, 2–8 July 2023*. <https://arxiv.org/abs/2307.02357> (accessed on 12 June 2026).
- Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S.A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., et al. (2018). Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.*, 41, 39–45. https://people.eecs.berkeley.edu/~matei/papers/2018/ieee_mlflow.pdf (accessed on 12 June 2026).