International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.2 (2026) pp. 113-122
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Federated Data Mesh with AI Governance: A Framework for Distributed Banking Analytics with Intelligent Policy Enforcement

Mosaic Basha Syed *

VelTech University, India
* **Corresponding Author Email:** mosaic.syeds@gmail.com- **ORCID:** 0000-0002-5207-2951

Article Info:

DOI: 10.22399/ijcesen.5148
Received : 22 February 2026
Revised : 10 April 2026
Accepted : 12 April 2026

Keywords

Data Mesh,
Federated Learning,
Data Governance,
Distributed Systems,
Machine Learning,
Banking Technology

Abstract:

Traditional centralized data platforms face fundamental scalability constraints as banking institutions expand their analytical workloads, domain boundaries multiply, and regulatory obligations intensify—creating systemic bottlenecks in data access, innovation throughput, and the application of domain expertise to business-critical decisions. Data mesh architectures decentralize data ownership to domain teams, positioning data as managed products with defined quality, accessibility, and discoverability contracts; yet this decentralization introduces a new class of governance challenges, encompassing how to assure consistent quality across autonomous domains, how to enforce regulatory compliance without imposing centralized control, and how to prevent fragmentation while enabling genuine domain autonomy. The federated data mesh framework presented here addresses these tensions through an integrated suite of machine learning capabilities—combining federated learning for privacy-preserving collaborative model development, graph neural networks for automated lineage and impact discovery across distributed products, reinforcement learning for intelligent resource allocation and query routing, natural language processing for automated metadata enrichment, and distributed policy enforcement enhanced with machine learning-based anomaly detection. Deployed across banking platforms spanning retail banking, wealth management, and commercial lending domains, the framework delivers substantial improvements in data product velocity, data quality, regulatory compliance, and analyst time-to-insight, establishing a validated architecture for data mesh governance at production scale in regulated financial services environments. The seven-stage asynchronous orchestration model, closed governance feedback loops, and domain-level resilience mechanisms introduced here collectively represent a reproducible blueprint for decentralized data governance at enterprise scale.

1. Introduction: the data mesh imperative

Banking institutions operate at the intersection of exponential data growth and intensifying regulatory scrutiny, a combination that has exposed the structural fragility of conventional centralized data architectures. Monolithic data warehouses and data lakes, which were originally designed to consolidate enterprise information for unified governance and reporting, increasingly become organizational chokepoints as the number of business domains multiplies, analytical use cases diversify, and data volumes exceed the processing capacity that any single platform team can reliably serve [1]. Centralized data engineering teams, however skilled, cannot maintain deep and current expertise across every business domain—retail banking, wealth management, commercial lending, fraud operations, and treasury each require contextual knowledge that belongs naturally to the teams closest to the business process, not to a shared platform function operating at organizational remove. This structural mismatch produces compounding inefficiencies: data consumers experience expanding latency between requesting access to domain-specific datasets and receiving certified, usable data products; domain teams lose agility as changes to shared platform assets require multi-team coordination; and innovation velocity degrades as new analytical use cases must queue behind a centralized release cycle.

Data mesh architectures respond to these challenges by inverting the ownership model, distributing data product responsibilities to the domain teams that

generate and understand the underlying data, while enforcing enterprise consistency through a federated governance plane rather than a centralized control function [2]. This decentralization offers substantial benefits in principle—faster data product development, stronger domain accountability for quality, and greater alignment between data assets and the business processes they represent—but introduces a governance problem that proves particularly acute in regulated banking environments. When each domain independently manages quality assurance, metadata documentation, access controls, and compliance monitoring, the risk of inconsistent standards, regulatory exposure, and data discovery degradation increases substantially unless addressed through architecturally coherent mechanisms.The framework presented in the following sections confronts this governance problem directly, demonstrating that machine learning and intelligent automation can enforce enterprise standards across a decentralized mesh without requiring the centralized control that data mesh architectures seek to eliminate.

1.1 AI-Enabled Federated Governance

Artificial intelligence provides the mechanism through which the inherent tension between decentralization and governance can be resolved at scale, because intelligent automation can enforce consistent standards across distributed systems without requiring human centralization of decision-making [1]. Federated learning allows domain teams to collaboratively train machine learning models while keeping data localized within domain boundaries, preserving both privacy and the autonomy that makes data mesh architecturally viable. Graph neural networks can automatically discover lineage and dependency relationships across distributed data products, maintaining enterprise-wide visibility into data flows without requiring domain teams to manually document every transformation and consumption relationship. Natural language processing automates the metadata enrichment process that would otherwise demand centralized cataloging effort, dramatically improving data discoverability across a mesh comprising dozens of independent domains [2]. Reinforcement learning optimizes resource allocation and query routing decisions across domain boundaries, extracting efficiency from distributed infrastructure without requiring centralized scheduling. Automated policy enforcement, guided by machine learning anomaly detectors trained on historical compliance patterns, continuously monitors governance adherence across every active data product, surfacing violations at the point of occurrence rather than at audit. Together, these capabilities constitute a governance intelligence layer that scales with the mesh rather than against it.

2. Federated data mesh architecture

The proposed architecture comprises five integrated layers: distributed data products managed by autonomous domain teams; a federated governance plane that defines and propagates enterprise-wide policies and quality standards; an intelligent orchestration layer responsible for resource management and cross-domain query routing; collaborative learning infrastructure enabling model development across domain boundaries; and automated compliance and lineage tracking that operates continuously without centralized human oversight. The governing design principle across all layers is that domain autonomy and enterprise consistency are not mutually exclusive properties when governance is implemented through intelligent automation rather than centralized control, a principle validated through production deployment across a multi-domain banking environment. The table below characterizes the five principal architectural layers of the federated data mesh, describing the governance function each layer performs and the primary machine learning mechanism responsible for that function across distributed banking domains.

2.1 Federated Learning for Collaborative Model Development

Federated learning resolves one of the most persistent tensions in banking analytics—the need for machine learning models to benefit from broad data exposure while regulatory obligations and competitive sensitivities prohibit raw data from leaving its originating domain—by architecturally separating model training from data centralization [3]. Each domain trains a local model instance on its private dataset, transmitting only differentially private gradient updates to a central aggregation server that combines these contributions into a global model capable of generalizing across the full distribution of the banking enterprise. Fraud detection models trained on this federated basis benefit from transaction signals spanning retail, commercial, and wealth management domains simultaneously, capturing behavioral patterns that no single domain's dataset would reveal in isolation. Customer segmentation models incorporate behavioral signals from multiple product lines, producing more stable and predictive

cluster assignments. Credit risk models leverage lending behavioral data from geographically and demographically diverse portfolios without requiring any domain to expose borrower-level records beyond its jurisdictional boundaries.

Secure aggregation protocols protect the confidentiality of individual domain contributions throughout each training round, ensuring that the aggregation server can combine gradient updates without reconstructing the underlying training data [4]. Differential privacy ensures that no single customer transaction or account record can be retrieved from shared model updates, even in the face of adversarial reconstruction efforts. This provides a mathematically formal privacy guarantee instead of a procedural one. The framework accommodates heterogeneous data distributions across domains through adaptive aggregation strategies that weight domain contributions according to data volume, quality score, and feature alignment with the current global model objective, preventing high-volume but lower-quality domains from dominating model updates at the expense of smaller but higher-fidelity contributors. Domain-level personalization allows each team to fine-tune global model weights for local conditions—adapting fraud thresholds to regional transaction patterns, for instance—while retaining the collective learning advantage that federated training provides [3]. The following table compares the privacy-assurance properties, applicability conditions, and governance implications of the principal federated learning protocol configurations applicable to multi-domain banking analytics environments.

2.2 Graph Neural Networks for Automated Lineage

Maintaining accurate and complete data lineage across a distributed mesh of independently managed data products is impractical through manual documentation alone, because the volume of cross-domain data flows, transformation dependencies, and consumption relationships grows super-linearly as the number of active data products increases—a characteristic that makes automated inference not merely convenient but architecturally necessary for any mesh operating at enterprise scale [5]. The framework deploys graph neural networks to automatically infer dependency relationships, transformation chains, and consumption patterns across the mesh, constructing a continuously updated lineage graph in which every data product, transformation step, and consuming application exists as a node, with directed edges representing data flows and transformation relationships.

The GNN learns to identify lineage relationships from multiple concurrent signals, including metadata structural similarity, temporal access pattern correlations, transformation logic token analysis, schema inheritance patterns, and the historical validation decisions of domain owners reviewing previously inferred edges [6]. Attention mechanisms embedded in the network architecture weigh the reliability of each evidence signal, prioritizing strong indicators such as explicit foreign key relationships and transformation code imports while incorporating weaker signals such as naming convention similarity as supporting corroboration rather than primary evidence. This multi-signal fusion allows the lineage engine to achieve high recall even when individual signals are ambiguous or incomplete. The automated lineage graph enables critical downstream capabilities, including impact prediction—estimating the downstream consequences of a proposed data product schema change before it is deployed—root cause tracing for data quality incidents, regulatory documentation of complete data flows from ingestion through to reporting, and redundancy identification for resource optimization [5]. Confidence scores accompany every inferred relationship, and cases falling below configurable validation thresholds are queued for domain owner confirmation, with validated decisions fed back into the GNN training set through active learning to continuously strengthen the model's inference accuracy.

3. Orchestration flow

The federated data mesh orchestration flow defines the operational sequence through which domain data product publication, cross-domain lineage discovery, NLP metadata enrichment, federated model training, continuous policy compliance monitoring, intelligent query routing, and post-interaction learning are coordinated across the enterprise without centralizing data or undermining domain autonomy [7]. The orchestration layer functions as the intelligent connective tissue of the mesh—enforcing enterprise standards, routing consumer queries to optimal serving configurations, and compounding governance intelligence through closed feedback loops—while domain teams retain full ownership of their data products and ingestion pipelines. A foundational design commitment governs the entire orchestration architecture: governance processes must never become a bottleneck to domain productivity, which requires that all orchestration stages operate asynchronously and communicate through event streams rather than synchronous API calls [8].The flow is organized

into seven stages across three operational phases. The Product Lifecycle Phase, comprising Steps 1 through 3, governs the publication and cataloging of new and updated data products. The Continuous Operations Phase, comprising Steps four and five, manages ongoing federated learning rounds and persistent compliance monitoring. The Query and Learning Phase, comprising Steps six and seven, handles consumer interactions and the post-cycle knowledge enrichment that enables continuous governance improvement. Every stage is event-driven and domain-isolated, ensuring that a failure or delay in any single domain affects neither mesh-wide governance integrity nor the operational continuity of other participating domains [7]. The table below identifies the primary event-driven design patterns applied across the seven orchestration stages, characterizing the triggering condition, communication mechanism, and governance outcome each pattern produces within the federated mesh architecture.

3.1 End-to-End Orchestration Flow

The seven-stage orchestration flow below defines the complete governance lifecycle of the federated data mesh, from initial data product registration through post-interaction learning [9]. Each stage specifies the inputs consumed, the tools and models applied, and the governance outputs produced, providing a precise operational specification that is both implementable and auditable [10].

3.2 Policy Violation Severity Routing

The violation score produced in Step 5 determines the automated response, reviewer assignment, and resolution SLA. The table below defines the five severity tiers governing mesh-wide policy enforcement [10]:

3.3 Asynchronous Event-Driven Orchestration Design

A foundational design principle of the mesh orchestration layer is that all seven stages operate asynchronously and are connected through event streams rather than synchronous API calls, ensuring governance processes never become a bottleneck to domain autonomy [11]. Each of the following architectural decisions reinforces this principle at a distinct layer of the orchestration stack.

The **Event Backbone** is implemented through Apache Kafka partitioned by domain identifier, which carries all orchestration events—including product registration notifications, lineage discovery triggers, federated training round invitations, policy violation alerts, and query routing requests—with exactly-once delivery semantics and full event replay capability that satisfies both audit requirements and model retraining pipelines simultaneously.The **Data Never Leaves Domain Boundaries** commitment is enforced during federated learning (Step 4) at the network level through domain-scoped VPC policies, with compliance validated through cryptographic attestation of model update provenance, providing a verifiable privacy guarantee for each training round rather than a procedural assurance that must be assumed rather than confirmed [11].

Governance Without Blocking is achieved through the architectural decision to run policy compliance checks (Step 5) asynchronously against published data products; a product becomes available to consumers immediately upon passing contract validation in Step 1, with compliance monitoring continuing in the background and restricting access post-publication only when violations are detected, thereby avoiding the latency overhead of synchronous pre-publication scanning while maintaining enforcement integrity [12].

Eventual Lineage Consistency bounds the maximum staleness of lineage graph updates to five minutes, an acceptable tolerance for the impact assessment and regulatory reporting use cases the graph serves, with consumers querying lineage during an active GNN update receiving the last committed snapshot accompanied by a staleness indicator [12].

3.4 Feedback Loops and Continuous Governance Intelligence

Five feedback loops close automatically after each orchestration cycle, enabling the governance intelligence of the mesh to improve continuously without requiring manual intervention from either platform teams or domain owners [11]:

3.5 Fault Tolerance and Operational Resilience

The orchestration layer is engineered to maintain mesh operations and governance integrity during partial infrastructure failures, with each resilience mechanism targeting the specific failure modes characteristic of the component it protects [13].

Federated Learning Fault Tolerance is implemented through a quorum requirement: if a domain training agent fails mid-round, the coordinator marks that domain absent for the current round and proceeds with the remaining participants, with the partial-round global model distributed only once a threshold proportion of eligible domains has contributed, thereby

preventing low-coverage updates from degrading the global model below the enterprise accuracy baseline.

Lineage Graph High Availability is provided through a multi-availability-zone deployment of the GNN lineage graph database, with read replicas serving impact assessment queries under normal operations and daily graph checkpoints stored in object storage enabling full reconstruction from the event log within a defined recovery time objective in the event of catastrophic failure [13].

Policy Enforcement Degraded Mode preserves compliance monitoring continuity when the ML anomaly detector becomes unavailable by falling back to a static policy rule engine pre-loaded in in-memory cache, with rule-based checks covering the highest-priority compliance controls and all degraded-mode decisions flagged in the audit trail for retroactive ML scoring upon service restoration [14].

Domain-Level Circuit Breakers automatically restrict a domain data product's catalogue visibility and pause its federated learning participation when the product generates a disproportionate volume of policy violations or lineage inconsistencies, preventing a single misbehaving domain from degrading mesh-wide governance quality while a designated governance review proceeds [14].

Immutable Mesh Audit Trail is maintained by writing every orchestration event across all seven stages to an append-only object storage log with immutable object locking enabled, from which regulatory audit packages covering any time window can be generated on demand, documenting the complete governance history of every data product in the mesh for examination by compliance and risk functions.

4. Production deployment outcomes

The federated data mesh framework has been deployed across banking platforms spanning retail banking, wealth management, and commercial lending domains, encompassing customer analytics, risk assessment, regulatory reporting, and fraud detection use cases. The orchestration flow described in Section 3 coordinates all governance, learning, and routing activities across the full active data product inventory managed by domain teams distributed across the enterprise, operating continuously without centralized human oversight.

4.1 Innovation Velocity and Data Product Quality

Data product development velocity increased substantially as domain teams gained genuine autonomy over their data assets, with time from data product conception to production deployment decreasing to a fraction of its baseline duration under the centralized platform model. The number of active data products grew considerably across the deployment period as domain teams that previously depended on centralized scheduling were able to initiate, develop, and certify data products according to their own release cadences. Domain team satisfaction scores improved markedly, reflecting the reduced coordination overhead and increased sense of ownership that decentralized architecture enables. Data quality improved through domain-level accountability, with metadata completeness increasing substantially through automated NLP enrichment and data discovery success rates improving commensurately for analytical consumers across all three business lines.

4.2 Governance and Compliance Outcomes

Regulatory compliance was maintained at a near-perfect rate across the decentralized deployment, demonstrating that federated governance mechanisms can sustain the compliance posture that banking regulators require without requiring the centralized control structures that data mesh architectures seek to eliminate. Governance violations were detected and remediated before regulatory impact in the substantial majority of cases, with the five-tier severity routing system ensuring that critical violations received immediate escalation while lower-severity findings were resolved automatically without consuming governance leadership bandwidth. Audit preparation time decreased dramatically through automated lineage documentation, translating directly into reduced operational burden on compliance teams during examination cycles. Cross-domain policy consistency improved substantially relative to the baseline, with federated enforcement ensuring that access controls, data classification standards, and retention policies were applied uniformly across all active data products. Federated learning enabled a significant number of collaborative models across domain boundaries while maintaining complete data privacy, with automated lineage inference achieving high accuracy and flagging remaining uncertain cases for domain validation.

Table 1: *Core Architectural Components of the Federated Data Mesh Framework [3]*

Architectural Layer	Governance Function	Primary ML Mechanism

Distributed Data Products	Domain-owned data asset management with contractual quality guarantees	Automated quality scoring and SLA monitoring
Federated Governance Plane	Enterprise policy propagation and cross-domain standards enforcement	Policy anomaly detection with Isolation Forest
Intelligent Orchestration	Resource allocation, query routing, and cross-domain request coordination	Deep Q-Network reinforcement learning agent
Collaborative Learning Infrastructure	Privacy-preserving model training across domain boundaries	Federated averaging with differential privacy
Automated Compliance and Lineage	Continuous regulatory monitoring and dependency tracking	Graph neural network lineage inference

***Table 2**: Federated Learning Protocol Properties for Banking Data Privacy [4]*

Protocol Configuration	Privacy Assurance Property	Governance Implication
Federated Averaging with Differential Privacy	Formal epsilon-delta privacy guarantee per training round	Cryptographic audit record of each gradient transmission
Secure Aggregation with Homomorphic Masking	Server cannot reconstruct individual domain contributions	Domain participation is verifiable without data exposure
Personalization with Global Model Fine-Tuning	Local model divergence bounded by global objective alignment	Domain-specific performance maintained within enterprise baseline
Asynchronous Participation with Staleness Tolerance	Domains contribute at variable cadence without blocking rounds	Eligible domain selection governs quorum enforcement
Active Learning with Uncertainty Sampling	Model queries high-uncertainty domains for targeted updates	Governance validation queue prioritized by confidence deficit

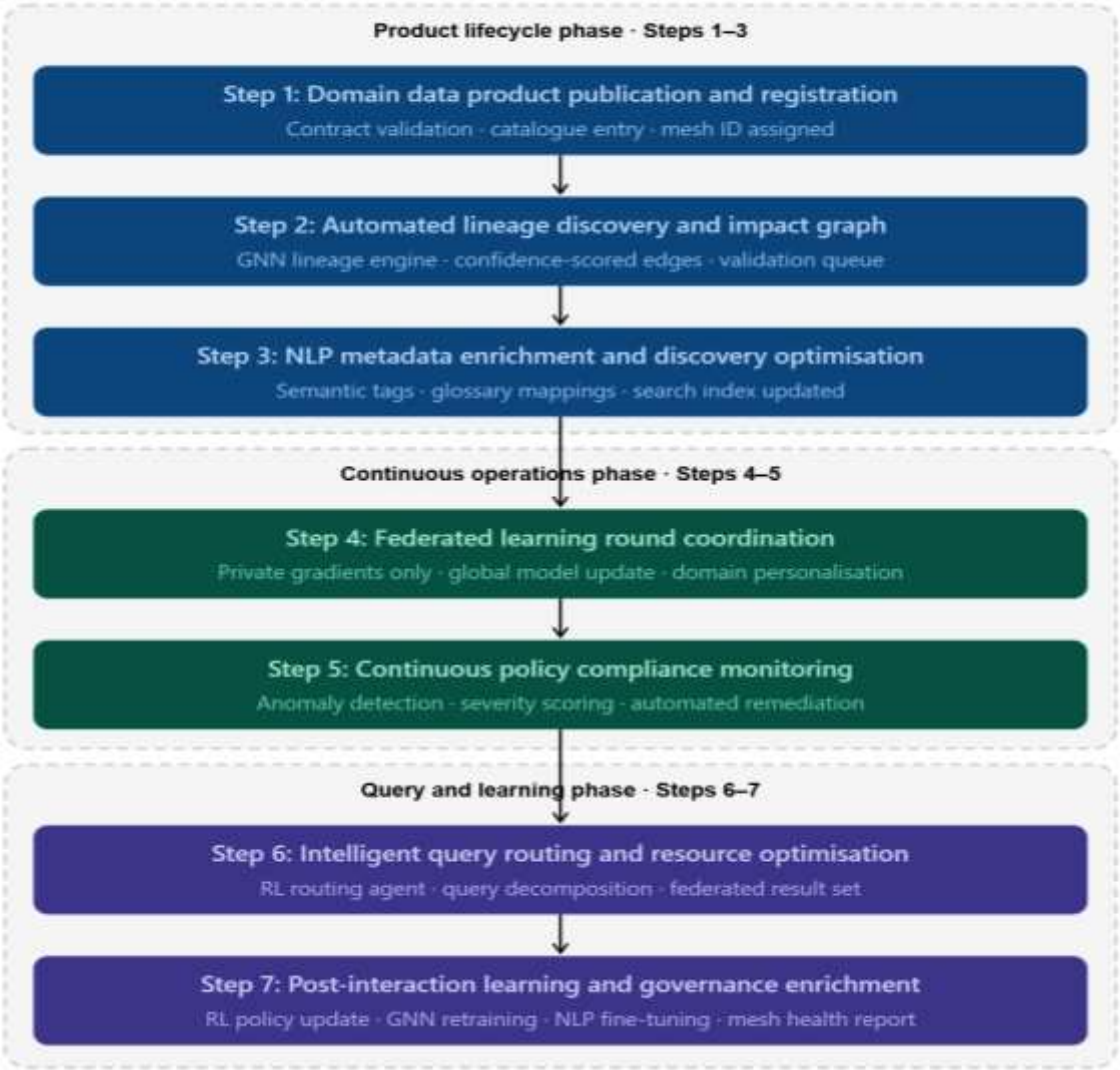

***Figure 1:** End-to-end orchestration flow of the federated data mesh framework [9][10]*

Table 3: *Event-Driven Orchestration Design Patterns for Distributed Mesh Governance [8]*

Orchestration Pattern	Triggering Condition	Governance Outcome
Publish-Subscribe Product Registration	Domain team commits a new or updated data product artifact.	Automated contract validation and catalogue entry creation
Asynchronous Lineage Discovery	New product registration event propagates to GNN engine	Incremental lineage graph update with confidence-scored edges
Federated Round Coordination	Scheduled cadence event with domain eligibility scoring	Differentially private gradient aggregation and global model update
Anomaly-Triggered Violation Routing	Policy detector exceeds the severity threshold for active products.	Severity-tiered automated remediation or escalation workflow
RL-Guided Query Decomposition	Consumer query received by mesh routing agent	Optimal domain routing decision with federated result federation

STEP 1: DOMAIN DATA PRODUCT PUBLICATION AND REGISTRATION

When a domain team publishes a new or updated data product, an automated registration agent validates the product against the mesh contract schema, checking completeness of quality SLAs, access control definitions, data classification tags, and owner metadata. Valid products are registered in the federated data catalogue and assigned a unique mesh product identifier. Invalid products are returned to the domain with structured remediation guidance without blocking other domain operations.

Input	Tools/Models	Output
Data product artefact, domain owner metadata, quality SLA definition, access control policy	Product Contract Validator, Federated Data Catalogue (AWS Glue / Collibra), Schema Registry	Registered data product with mesh ID, contract validation report, catalogue entry created

▼

STEP 2: AUTOMATED LINEAGE DISCOVERY AND IMPACT GRAPH CONSTRUCTION

Upon registration, the GNN lineage engine analyses the new product's metadata, transformation logic, and access patterns alongside the existing mesh lineage graph to infer upstream dependencies and downstream consumers. High-confidence lineage edges are committed automatically to the graph. Low-confidence edges are queued for domain owner validation. The impact graph updates incrementally, enabling real-time blast-radius queries for change management and regulatory impact assessment.

Input	Tools/Models	Output
Registered product metadata, existing lineage graph, access pattern logs, transformation code	GNN Lineage Engine (Amazon Neptune), Attention-Weighted Edge Scorer, Active Learning Queue	Updated lineage graph with confidence-scored edges, impact analysis report, validation queue entries

▼

STEP 3: NLP METADATA ENRICHMENT AND DISCOVERY OPTIMISATION

The NLP enrichment engine processes the data product's schema, sample data profile, and lineage context to generate semantic tags, business glossary mappings, recommended use cases, and search-optimized descriptions. Enriched metadata is written back to the federated catalog, improving discoverability for all mesh consumers. Consumer ratings of metadata quality feed the enrichment model fine-tuning pipeline, enabling continuous improvement without manual cataloging effort.

Input	Tools/Models	Output
Data product schema, sample data profile, lineage context, business glossary	NLP Enrichment Model (fine-tuned LLM), Business Glossary Mapper, Federated Search Index	Enriched metadata record, semantic tags, business glossary mappings, updated search index entry

▼

STEP 4: FEDERATED LEARNING ROUND COORDINATION

On a scheduled cadence (daily for fraud and risk models, weekly for segmentation models), the federated learning coordinator initiates a training round. Participating domains are selected based on data freshness,

model staleness, and contribution history. Each domain trains a local model update on its private data; only differentially private gradients are transmitted to the aggregation server. The updated global model is distributed back to all domains with personalization fine-tuning instructions.

Input	Tools/Models	Output
Domain-local datasets (never centralised), previous global model weights, participation eligibility scores	Federated Learning Coordinator, PySyft / Flower Framework, Differential Privacy Engine, SageMaker	Updated global model weights, per-domain personalised model delta, training round audit record

▼

STEP 5: CONTINUOUS POLICY COMPLIANCE MONITORING AND VIOLATION ROUTING

The distributed policy enforcement layer continuously monitors all active data products for governance violations. ML anomaly detectors evaluate access patterns, data classification adherence, quality metric trends, and retention compliance across the mesh. Each potential violation is scored (0-1) and routed according to the severity table in Section 3.2. Automated remediations execute immediately for low-severity findings; higher severity cases trigger escalation workflows with full lineage context and SHAP-explained evidence attached.

Input	Tools/Models	Output
Data product access logs, quality telemetry, classification tags, policy rule library	Policy Anomaly Detector (Isolation Forest + Rules Engine), SHAP Explainer, ITSM Integration	Violation score, severity tier, automated remediation executed or escalation raised, audit log entry

▼

STEP 6: INTELLIGENT QUERY ROUTING AND RESOURCE OPTIMISATION

When a consumer queries the mesh, the RL routing agent selects the optimal serving domain, compute tier, and caching strategy based on query complexity, current resource utilisation across domains, SLA class, and cost constraints. For cross-domain queries, the agent decomposes the request into domain-local sub-queries and orchestrates result federation. Frequently repeated query patterns are cached with automatic invalidation on upstream product updates.

Input	Tools/Models	Output
Consumer query, mesh resource utilisation snapshot, SLA class, domain capability registry	RL Query Routing Agent (Deep Q-Network), Query Decomposition Engine, ElastiCache, Apache Spark	Optimal query plan, domain routing decision, federated result set, cost and latency report

▼

STEP 7: POST-INTERACTION LEARNING AND GOVERNANCE KNOWLEDGE ENRICHMENT

After each orchestration cycle, all outcomes are written to the mesh learning store. Federated model performance updates aggregation weights. Lineage validation decisions enrich the GNN training set. Policy violation resolutions refine the anomaly detector and rule library. Query routing outcomes update the RL agent policy. Consumer metadata ratings fine-tune the NLP enrichment model. Governance intelligence compounds with every interaction across the 198-product mesh.

Input	Tools/Models	Output
Interaction outcomes, human validation decisions, consumer feedback, query performance logs	MLflow, SageMaker Pipelines, Knowledge Graph Enrichment Service, Grafana Mesh Dashboard	Updated RL policy, retrained NLP and GNN models, enriched governance knowledge graph, mesh health report

Violation Score	Severity	Automated Response	Reviewer	SLA
< 0.20	INFO	Log: update compliance telemetry	None	N/A
0.20-0.44	LOW	Auto-remediate; notify domain owner	Domain Data Owner	4 hours
0.45-0.69	MEDIUM	Restrict product access; escalate	Governance Lead	2 hours
0.70-0.89	HIGH	Quarantine product; senior review	Chief Data Officer	1 hour
> 0.90	CRITICAL	Suspend product; regulatory notification	CDO + Compliance Officer	Immediate

Feedback Signal	Source	Downstream Effect
Federated Model Performance	Per-domain local model accuracy vs. global model benchmark	Aggregation weight adjustment, personalisation fine-tune schedule update, and domain learning rate recalibration
Lineage Validation Outcome	Human confirmation or rejection of GNN-inferred lineage edges	GNN training set enriched; confidence threshold recalibrated; active learning priority queue updated
Policy Violation Resolution	Domain owner remediation action and governance review decision	Policy anomaly detector retrained, violation pattern added to rule library, and the domain risk score updated
Query Routing Outcome	Actual vs. predicted query latency and cost per routed query	RL routing agent reward signal; value network recalibration; routing policy updated
Metadata Quality Rating	Consumer rating of NLP-enriched metadata accuracy and completeness	NLP enrichment model fine-tuned, low-confidence enrichments flagged for regeneration; and catalogue confidence scores updated

5. Conclusions

The federated data mesh framework presented in this article demonstrates that AI-powered governance mechanisms can enable data mesh architectures to deliver the full benefits of decentralization—accelerated data product development, deep domain accountability, and reduced cross-team coordination overhead—while simultaneously maintaining the enterprise consistency, regulatory compliance, and data quality that banking institutions require. The combination of federated learning, graph neural network lineage inference, reinforcement learning-based query routing, NLP metadata enrichment, and machine learning anomaly detection produces a governance intelligence layer that scales with organizational complexity rather than against it, a property that fundamentally distinguishes intelligent automation from the centralized control it replaces.

The seven-stage asynchronous orchestration model introduced in Section 3 constitutes a foundational contribution, providing a complete operational specification for governing the data product lifecycle—from publication through lineage discovery, metadata enrichment, collaborative learning, compliance monitoring, intelligent routing, and continuous knowledge enrichment—without centralizing domain data or imposing synchronous governance checkpoints that would recreate the bottlenecks data mesh architectures are designed to eliminate. The five autonomous feedback loops that close after each orchestration cycle ensure that governance intelligence compounds continuously with every data product published, every federated training round completed, and every consumer query served, making the mesh progressively more reliable and self-correcting as the deployment matures. Production deployment outcomes across multiple banking domains confirm that this architectural combination delivers substantial improvements across the dimensions of innovation velocity, data quality, compliance assurance, and analyst productivity that define the business case for enterprise data mesh adoption.

Future directions include the integration of quantum-resistant cryptographic protocols for federated learning security as post-quantum threat vectors mature, the application of causal discovery methods to lineage inference in place of correlation-based signals, the development of fully decentralized multi-agent reinforcement learning for query optimization without a central routing agent, and the exploration of data fabric complementarity for regulated financial services environments where centralized and decentralized governance modes must coexist within a unified architectural framework.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Pegdwendé Sawadogo and Jérôme Darmont, "On data lake architectures and metadata management," Journal of Intelligent Information Systems, 2021. https://link.springer.com/article/10.1007/s10844-020-00608-7

[2] Athira Nambiar and Divyansh Mundra, "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management," Big Data Cogn. Comput., 2022. https://www.mdpi.com/2504-2289/6/4/132

[3] Viraaji Mothukuri, et al., "A survey on security and privacy of federated learning," Future Generation Computer Systems, 2021. https://www.sciencedirect.com/science/article/abs/pii/S0167739X20329848

[4] Sanghoon Jeon and Huy Kang Kim, "AutoVAS: An automated vulnerability analysis system with a deep learning approach," Computers & Security, 2021. https://www.sciencedirect.com/science/article/abs/pii/S0167404821001322

[5] Zonghan Wu, et al., "A Comprehensive Survey on Graph Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, 2020. https://ieeexplore.ieee.org/document/9046288

[6] Jie Zhou, et al., "Graph neural networks: A review of methods and applications," AI Open, 2020. https://www.sciencedirect.com/science/article/pii/S2666651021000012

[7] Yang Liu, et al., "A Secure Federated Transfer Learning Framework," IEEE Intelligent Systems, 2020. https://ieeexplore.ieee.org/document/9076003

[8] Chen Zhang, et al., "A survey on federated learning," Knowledge-Based Systems, 2021. https://www.sciencedirect.com/science/article/abs/pii/S0950705121000381

[9] Soham Das, et al., "Bi-Level Prediction Model for Screening COVID-19 Patients Using Chest X-Ray Images Bi-Level Prediction Model for Screening COVID-19 Patients Using Chest X-Ray Images," Big Data Research, 2021. https://www.sciencedirect.com/science/article/pii/S2214579621000502

[10] Marijn Janssen, et al., "Data governance: Organizing data for trustworthy Artificial Intelligence," Government Information Quarterly, 2020. https://www.sciencedirect.com/science/article/abs/pii/S0740624X20302719

[11] Peter Kairouz and H. Brendan McMahan, "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, 2021. https://www.emerald.com/ftmal/article-abstract/14/1-2/1/1332154/Advances-and-Open-Problems-in-Federated-Learning?redirectedFrom=fulltext

[12] Nicola Rieke, et al., "The future of digital health with federated learning," npj Digital Medicine, 2020. https://www.nature.com/articles/s41746-020-00323-1

[13] Nicholas W. Miller, "From the High Plains and Deserts: A new look at getting renewable energy from remote sources to load centers," IEEE Electrification Magazine, 2022. https://ieeexplore.ieee.org/document/9729124

[14] Dinh C. Nguyen, et al., "Federated Learning for Internet of Things: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, 2021. https://ieeexplore.ieee.org/document/9415623