A Multi-Dimensional Taxonomy of Data Governance Mechanisms for Modern Financial Data Engineering

Ramkumar Selvarathinam
Capital One
15050 Capital One Drive
Richmond, VA

ABSTRACT
The rapid evolution of cloud-native data engineering, real-time analytics, distributed data ownership, and AI-enabled decision systems creates governance challenges for financial institutions operating under stringent regulatory requirements. This paper proposes a multi-dimensional taxonomy of data governance mechanisms for modern financial platforms. The taxonomy classifies governance along four complementary dimensions: governance layers, enforcement modes, control paradigms, and automation maturity. A four-level automation maturity model and a reference architecture translate the taxonomy into operational capabilities spanning ingestion, storage, consumption, metadata and lineage, policy enforcement, and audit and monitoring. Practical applicability is examined through a representative fintech trading platform with four scenarios: customer onboarding, stock transaction processing, real-time market-data processing, and regulatory and business reporting. The analysis shows that heterogeneous financial workflows require different combinations of preventive, detective, and corrective controls while remaining classifiable within a common governance structure. The paper concludes by identifying research challenges in low-latency streaming governance, cross-domain policy consistency, explainable governance decisions, AI-driven automation, and governance for ML/AI pipelines.

Keywords
Data Governance; Financial Data Engineering; Governance Taxonomy; Policy-as-Code; Data Quality; Data Lineage; Data Mesh; Governance Automation; Fintech

1. INTRODUCTION
Financial institutions process large volumes of sensitive customer, transactional, market, and portfolio data while operating under regulations and standards such as the Gramm-Leach-Bliley Act (GLBA), California Consumer Privacy Act (CCPA), General Data Protection Regulation (GDPR), and Payment Card Industry Data Security Standard (PCI-DSS). Data governance therefore extends beyond data sharing: it encompasses the organizational structures, policies, processes, standards, technologies, roles, and decision rights required to manage data as a strategic enterprise asset [1], while also addressing responsible data use, privacy risk, stakeholder expectations, and transparency [2].

Cloud-native platforms, event-driven architectures, real-time analytics, and AI-enabled decisioning have made this governance problem more difficult. Financial activities are increasingly data-centric: transactions represent transfers of data, financial infrastructures operate as data networks, and institutions act as large-scale processors of customer and transactional information [3]. Governance mechanisms designed primarily for centralized warehouses or periodic batch controls do not fully capture the requirements of streaming pipelines, distributed ownership, API-driven services, and low-latency operational decisions.

Prior research has examined governance frameworks, data quality, metadata and lineage, policy enforcement, distributed ownership, and governance maturity, but these perspectives are often treated independently. Modern financial platforms require a framework that can simultaneously describe where a control operates, how it is enforced, who owns it, and how automated it is. This paper addresses that gap through a unified taxonomy and an architecture-oriented application of the taxonomy to financial data engineering.

The primary contributions are:

• A multi-dimensional taxonomy organized by governance layers, enforcement modes, control paradigms, and automation maturity.

• A four-level governance automation maturity model extending conventional maturity concepts toward rule-based, Policy-as-Code, and AI-assisted enforcement.

• A reference architecture that embeds metadata, policy enforcement, lineage, quality, security, and auditability throughout the data lifecycle.

• A scenario-based evaluation using four representative fintech workflows and an analysis of open research challenges.

2. RELATED WORK
Data governance has evolved from a predominantly organizational and compliance-oriented discipline toward a broader technical framework encompassing quality, privacy, ownership, metadata, policy enforcement, and lifecycle management. Recent systematic reviews report persistent organizational, technical, regulatory, cultural, and data-quality challenges [4], while comparative analysis of governance frameworks shows substantial variation in definitions, components, objectives, and application contexts [1]. This variation limits the usefulness of any single generic framework for heterogeneous modern financial platforms.

Distributed architectures have also changed governance responsibility. Data Mesh emphasizes domain ownership, data as a product, self-service platforms, and federated computational governance [5]. Dolhopolov et al. [6] examine federated governance in Data Mesh, while van Eijk et al. [7] show how shared platform capabilities can support governed self-service data products. These studies establish the importance of decentralized ownership and common platform controls, but do not provide a unified classification of technical governance mechanisms.

Research increasingly embeds governance within engineering workflows. Polimeno et al. [8] examine the interaction between data quality and data protection in service-based pipelines.

Costa e Silva et al. [9] demonstrate the importance of continuous quality monitoring for streaming analytics. Ahmad et al. [10] show how documented API rules can be complemented by automated conformance analysis. Chen et al. [11] highlight lineage graphs as a basis for traceability, impact analysis, and auditability. Together, these works show that governance is becoming executable and operational rather than remaining only procedural.

Financial data governance adds domain-specific pressure because institutions combine sensitive data, regulatory obligations, high-volume transactions, distributed ownership, and increasingly real-time systems [3]. At the same time, maturity research provides structured ways to assess governance capabilities [12,13,14], but existing maturity models vary in scope and generally focus on organizational or capability maturity rather than the automation level of individual controls. Policy-as-Code provides a technical mechanism for machine-executable, version-controlled governance [17], while Responsible AI Governance extends oversight to transparency, accountability, explainability, human oversight, and model lifecycle management [18].

The research gap is therefore not the absence of governance concepts but their fragmentation. The proposed taxonomy integrates lifecycle location, enforcement behaviour, organizational ownership, and automation into one comparative structure tailored to modern financial data engineering.

TABLE I. Comparison of representative data governance approaches and the proposed taxonomy across lifecycle, ownership, real-time processing, automation, and financial-domain dimensions

Approach	Life-cycle Orientation	Distributed Ownership	Streaming/ Real-Time	Automation Focus	Finance Specific
DAMA-DMBOK [16]	Strong	Limited	Limited	General Capability	No
Data Mesh [5]	Partial	Strong	Supported	Moderate	No
Federated Data Mesh [6]	Partial	Strong	Supported	Strong	No
Self-serve Data Mesh [7]	Partial	Strong	Supported	Strong	No
DG maturity literature [12,13,14]	Broad	Varies	Not primary focus	Maturity-Oriented	No
Proposed taxonomy	Strong	Explicit Dimension	Explicitly Addressed	Automation Maturity	Yes

3. MATERIALS AND METHODS

The taxonomy was developed through a multi-stage conceptual synthesis of established governance frameworks and modern data-engineering practices. The review considered DAMA-DMBOK [16], Data Mesh principles [5], governance maturity literature [12,13,14], and recent work on data quality, API governance, lineage, Policy-as-Code, and responsible AI. These concepts were organized around the financial data lifecycle and refined against recurring characteristics of cloud-native financial platforms: high-velocity streaming, API-based integration, sensitive data, distributed domain ownership, and strict auditability requirements.

The resulting taxonomy was then applied to a representative fintech trading platform. Four scenarios were selected to stress different governance concerns: customer onboarding, synchronous trading transactions, real-time market-data processing, and cross-domain reporting. The evaluation is scenario-based rather than a production benchmark; its purpose is to test whether the taxonomy consistently classifies heterogeneous governance requirements and provides a practical mapping from conceptual dimensions to architecture-level controls.

OpenAI ChatGPT (GPT-5 series) was used during manuscript preparation to support language refinement, manuscript organization, and the presentation of selected concepts and diagrams. The development of the proposed taxonomy and governance framework, selection and design of the financial scenarios, framework evaluation, and interpretation of the findings were performed by the author. All AI-assisted content was independently reviewed, verified, and edited by the author. Generative AI was not used to generate empirical data or independently conduct the evaluation presented in this study.

4. FINANCIAL DATA ECOSYSTEM

High-velocity financial systems also creates a governance trade-off between control strength and operational continuity. Producers of transactional data may be reluctant to introduce blocking checks that can fail customer transactions, while downstream risk and reporting functions require strong assurances about completeness, consistency, and provenance. This tension makes the choice of preventive versus detective controls a design decision rather than a purely compliance decision. Controls on synchronous transaction paths must be lightweight, whereas deeper reconciliation, anomaly detection, and audit checks can often execute asynchronously.

Regulatory obligations also affect engineering decisions at multiple lifecycle stages. Privacy and security requirements influence collection, classification, access, masking, retention, and deletion; payment-data standards influence storage and service boundaries; and audit requirements increase the importance of immutable evidence and reproducible transformations. For this reason, the proposed taxonomy treats governance as a set of mechanisms distributed across the platform rather than as a single compliance layer.

Financial data platforms exhibit four characteristics that make governance technically demanding. First, high-velocity streaming and event-driven processing require controls to operate continuously without creating unacceptable latency. Second, sensitive customer and payment data are subject to strict privacy, security, retention, and audit requirements under frameworks such as CCPA, GLBA, GDPR, PCI-DSS [19-22]. Third, end-to-end lineage is necessary to trace attributes across ingestion, transformation, storage, reporting, and analytical consumption [11]. Fourth, ownership is distributed across interconnected domains such as customer, accounts, transactions, market data, and portfolio, requiring domain accountability while preserving enterprise-wide standards [6,7].

These characteristics are amplified in latency-sensitive analytics and ML pipelines such as fraud detection, risk monitoring, and automated approvals. As organizations increasingly rely on data to support operational, analytical, and strategic decision-making, the ability to systematically collect, manage, analyse, and use data becomes an enterprise capability rather than an isolated technical function [15]. Governance must therefore function as an engineering capability embedded in pipelines and services, rather than as an external review step. The combination of velocity, regulatory sensitivity, traceability, and distributed ownership motivates the four-dimensional taxonomy presented next.

5. MULTI-DIMENSIONAL TAXONOMY

A useful property of the taxonomy is composability. A single business workflow may contain several governance mechanisms with different classifications. For example, an onboarding API may use preventive schema validation at ingestion, detect duplicate records after persistence, federated ownership across Customer and Account domains, and Level 2 or Level 3 automation depending on whether the rules are implemented as application logic or managed policy artifacts. The taxonomy therefore classifies individual controls rather than assigning a single fixed label to an entire platform.

This distinction also supports architecture reviews. Teams can examine whether a critical dataset has adequate lifecycle coverage, whether preventive controls are overused on latency-sensitive paths, whether domain autonomy is balanced by enterprise standards, and whether frequently repeated manual controls are candidates for automation. In this way, the taxonomy can serve both as a research classification and as a practical design checklist.

The proposed taxonomy characterizes governance mechanisms along four complementary dimensions: (1) governance layers, which identify where controls operate in the data lifecycle; (2) enforcement modes, which describe when and how violations are handled; (3) control paradigms, which describe how governance authority is distributed; and (4) automation maturity, which captures the degree to which governance is machine-enforced.

5.1 Governance Layers

Ingestion governance controls data at entry points through schema validation, data contracts, data-quality checks, deduplication, integrity validation, and real-time validation for streams [8,16]. Storage governance protects persistent assets through encryption, classification, retention, archival, isolation, backup, and disaster-recovery controls [16]. Compute governance validates transformations through lineage, pipeline testing, transformation monitoring, reproducibility, and version control. Consumption governance controls downstream use through RBAC/ABAC, masking, anonymization, query auditing, certified datasets, entitlement validation, and API-level controls. Together, these layers provide lifecycle coverage without requiring each control to be implemented by the same team or technology.

5.2 Enforcement Modes

Governance can be preventive, detective, or corrective. Preventive controls block violations before execution, for example schema enforcement, access checks, or CI/CD policy validation. Detective controls identify violations during or after execution through quality monitoring, audit logging, anomaly detection, and compliance reporting; these are particularly important for streaming environments where continuous observation is required [9]. Corrective controls remediate violations through rollback, data repair, policy-triggered actions, or pipeline recovery. A mature platform typically combines all three modes according to business criticality and latency constraints.

5.3 Control Paradigms

Governance authority can be centralized, federated, or decentralized. Centralized governance provides uniform policy definition and strong enterprise consistency but can create bottlenecks [16,23]. Federated governance distributes implementation and ownership to domains while maintaining common enterprise standards [6]. Decentralized governance gives domains greater autonomy but increases the need for shared metadata, interoperability, and policy synchronization. Modern financial platforms frequently use a federated model because it balances domain expertise with enterprise compliance.

5.4 Automation Maturity

The automation dimension distinguishes the sophistication of governance execution rather than overall organizational maturity. Level 1 is manual governance, where policies are documented and verified through human review. Level 2 uses deterministic rule-based systems for repeatable quality, security, and compliance checks. Level 3 represents Policy-as-Code, where governance rules are machine-executable, version-controlled, testable, deployable, and monitored as software artifacts [17]. Level 4 is AI-assisted governance, where machine learning and advanced analytics support classification, anomaly detection, recommendations, and adaptive control while maintaining appropriate human oversight.

The four dimensions are intentionally independent. A preventive access policy at the consumption layer, for example, may be centrally defined but enforced through Level 3 Policy-as-Code; a streaming quality rule may be domain-owned, detective, and Level 2 automated. This separation allows governance mechanisms to be compared without collapsing distinct architectural and organizational concerns.

6. REFERENCE ARCHITECTURE

The architecture separates business data movement from cross-cutting governance capabilities. This separation is important because metadata, policy evaluation, lineage, and monitoring must remain consistent even when domains use different ingestion technologies or storage engines. Shared governance services can expose reusable capabilities, while domain teams remain responsible for the quality and meaning of their data products. Such a design is aligned with self-service Data Mesh platforms in which common platform capabilities reduce duplicated infrastructure work [7].

The architecture does not require every control to block data movement. Preventive controls are appropriate when a violation would create immediate regulatory, security, or transactional risk. Detective controls are appropriate when continuous monitoring can identify issues without adding latency to critical paths. Corrective workflows can then quarantine, repair, replay, or otherwise remediate affected data. This layered use of enforcement modes is particularly relevant to streaming systems.

The reference architecture operationalizes the taxonomy across three primary data lifecycle layers—ingestion, storage, and consumption—supported by cross-cutting metadata and lineage, policy enforcement, and audit and monitoring services. It represents an advanced Level 2-to-Level 3 environment in

which many controls are automated and selected policies are executable.

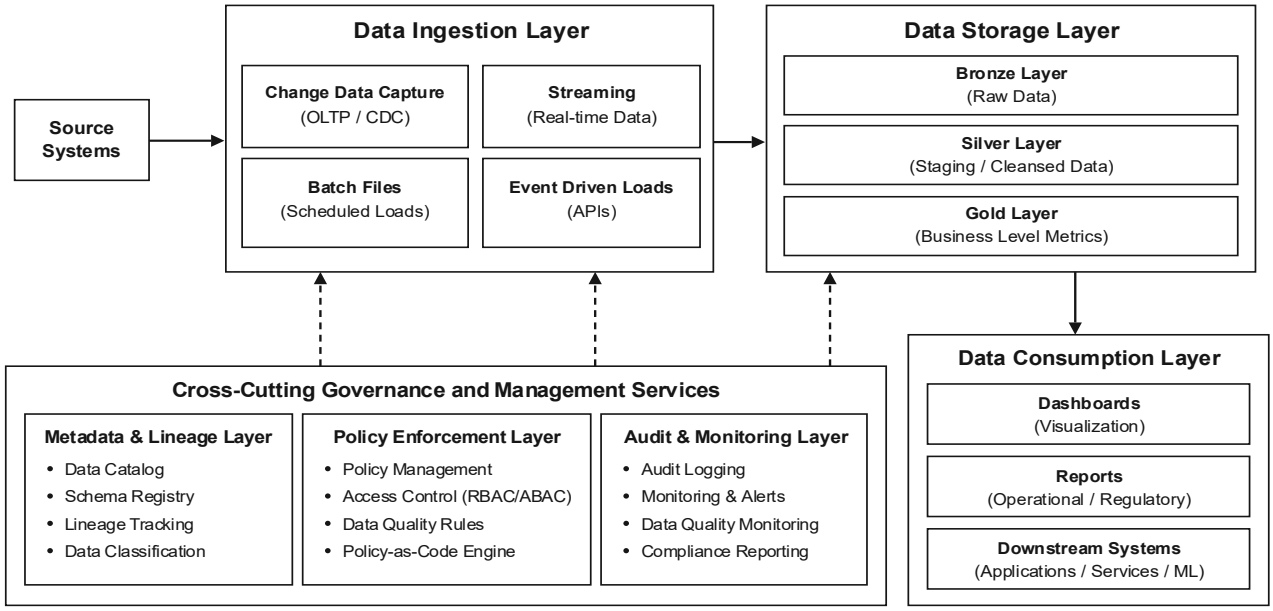


Fig. 1. Proposed reference architecture illustrating governance enforcement across ingestion, storage, consumption, metadata, policy enforcement, and audit layers

6.1 Architectural Principles

Five principles guide the architecture. Governance by design embeds controls directly into pipelines and services. Metadata-driven operations use ownership, lineage, classification, quality metrics, and policy assignments to drive decisions [16]. Policy-based enforcement translates selected requirements into executable rules. End-to-end traceability links source data, transformations, data products, and consumers to support audit and root-cause analysis. Federated governance enablement assigns domain ownership while preserving enterprise standards through shared platform capabilities [7].

6.2 Data Ingestion Layer

Financial institutions ingest data through CDC, real-time streams, scheduled batch files, and event-driven APIs. Regardless of pattern, ingestion governance should validate schemas and contracts, perform foundational quality checks, register metadata, and isolate invalid records for remediation. API-based ingestion additionally requires authentication and versioning, while standardized interface rules and automated conformance checks can support governance at scale [10]. Sensitive data should be classified early so that masking and downstream access policies can be applied consistently.

6.3 Data Storage Layer

The storage layer provides persistent organization of enterprise data, commonly separating raw, cleansed, and business-ready assets into bronze, silver, and gold zones. Governance at this layer focuses on classification, encryption, retention, immutability where required, lifecycle management, backup, and disaster recovery. The raw layer preserves source fidelity; the cleansed layer standardizes and validates data; and the business-ready layer exposes trusted datasets for reporting, analytics, and machine learning.

6.4 Data Consumption Layer

The consumption layer governs how business-ready data is used by different teams and applications. Operational teams need timely customer, account, payment, transaction, and portfolio data but should receive only the attributes required for their roles. RBAC, ABAC, masking, and tokenization therefore enforce least-privilege access. Risk, compliance, and audit teams require traceability, reproducibility, and evidence generation for KYC, AML, sanctions screening, control testing, and regulatory examination. Finance and regulatory reporting teams require approved data products, reconciled metrics, standardized definitions, versioned reporting logic, and reproducible historical outputs.

Analytics teams require discoverable and well-documented datasets with business glossaries, ownership, quality indicators, and certification status. Data science and ML teams additionally require authorization of training and inference data, source traceability, quality monitoring, and de-identification where appropriate. Customer-facing and downstream applications consume data through APIs, streams, or services and therefore require service-to-service authorization, interface contracts, and usage monitoring. Executive users generally consume aggregated dashboards and scorecards, but those metrics must still be derived from trusted data products and supported by lineage. The objective of consumption governance is therefore not merely access restriction; it is controlled, fit-for-purpose reuse across operational, analytical, regulatory, and AI-driven use cases, consistent with the broader view of data as a strategic organizational asset [15].

6.5 Cross-Cutting Governance Services

Metadata and lineage provide the foundation for ownership, discovery, impact analysis, quality rule configuration, and end-to-end traceability [11]. The policy enforcement layer evaluates data-quality, access, retention, and compliance policies and supports Policy-as-Code for rules suitable for automated enforcement. Audit and monitoring services observe pipeline health, latency, data quality, access violations, policy exceptions, retention compliance, and report delivery, producing detective controls and evidence for regulatory audits.

7. CASE STUDY: MODERN FINTECH PLATFORM

The scenario is illustrative and does not represent the architecture, systems, data, or governance practices of any specific financial institution.

7.1 Business Domain Overview

The case study considers a representative cloud-native fintech platform that enables brokerage account opening, bank-account linking, deposits and withdrawals, securities trading, real-time market monitoring, and portfolio insights. The platform integrates stock exchanges, market-data providers, banking networks, identity services, and customer applications. It is organized into domain-owned data products that operate under enterprise-wide governance policies [6].

The Transactions domain owns orders, executions, cash movements, dividends, and settlement events. The Accounts domain manages brokerage-account lifecycle activities such as opening, closure, suspension, bank linking, and regulatory status. The Stocks domain owns market prices, bid/ask quotes, volume, volatility, corporate actions, and reference information. The Customer domain is the authoritative source for identity, profile, KYC/AML, contact, sanctions, and risk information. The Portfolio domain represents holdings, cash balances, gains and losses, allocation, performance, and risk exposure.

7.2 Scenario 1: Customer Onboarding and Account Opening

Customer onboarding spans Customer, Account, and Portfolio domains and combines identity verification, contact validation, regulatory checks, registration, and portfolio initialization. Preventive controls validate mandatory fields, formats, duplicates, identity, KYC/AML status, and sanctions before persistence. PII is classified and protected through encryption and fine-grained access controls, while consent and transparency requirements govern appropriate collection and use [2]. As operational records enter analytical storage, schema validation, metadata registration, quality assessment, lineage capture, and policy enforcement ensure that only trusted and compliant data progress downstream.

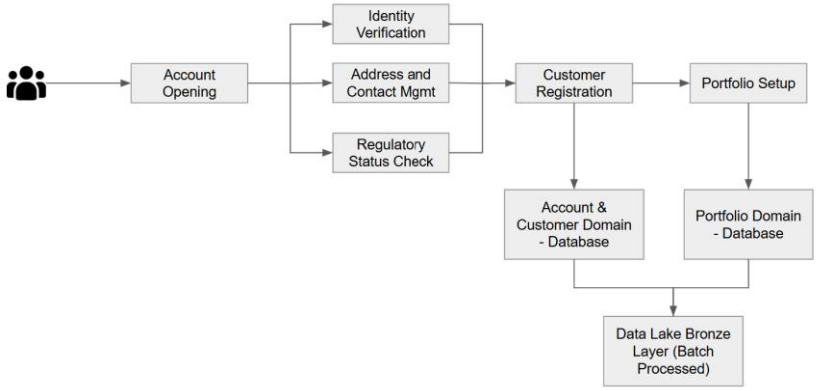


Fig. 2. Representative customer onboarding and account-opening workflow

TABLE II. Mapping of representative customer onboarding controls to ingestion, data quality, metadata and lineage, security and privacy, and compliance governance areas

Governance Area	Representative Controls
Ingestion	Schema and mandatory-field validation; duplicate detection; API authentication; metadata capture
Data quality	Identity/address validation; completeness; uniqueness; referential integrity; business rules
Metadata/ lineage	Glossary; ownership; classification; lineage registration
Security/ privacy	PII classification; encryption; tokenization; RBAC/ABAC; consent
Compliance	KYC; AML; sanctions screening; audit logging; regulatory status

7.3 Scenario 2: Stock Buy/Sell Transaction Processing

Trading transactions are latency-sensitive and depend on synchronous API calls across account, market-data, transaction, compliance, and portfolio services. Governance is therefore embedded at service boundaries. Authentication and authorization protect transaction initiation; schema and input validation ensure request integrity; business rules validate available cash, trading eligibility, supported securities, and market conditions. Standardized API rules and automated conformance checks provide consistent guardrails across independently developed services [10].

Operational resilience requires idempotency to prevent duplicate execution, rate controls to protect services, correlation identifiers for end-to-end traceability, and audit logs for every invocation, validation decision, and state transition. Metadata captures service ownership, API versions, execution timestamps, and dependencies. Data-quality and reconciliation controls validate price consistency, transaction completeness, cash updates, and portfolio positions before downstream publication. This scenario is predominantly preventive, federated, and suitable for Level 3 Policy-as-Code because controls must be consistent and low-latency.

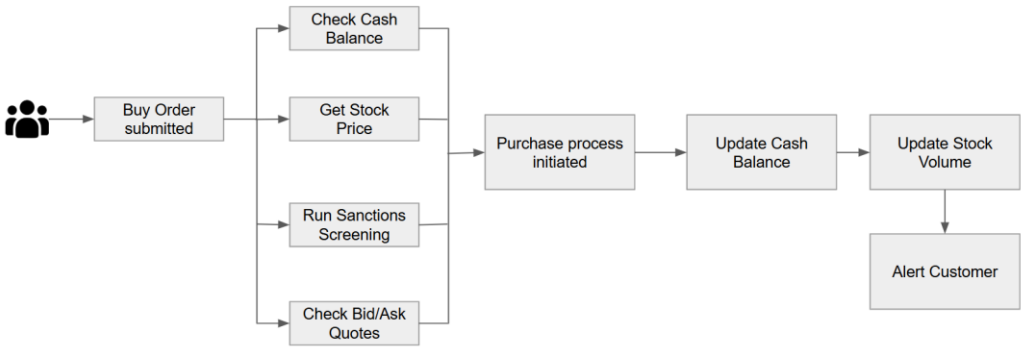


Fig. 3. Representative stock buy/sell transaction workflow

7.4 Scenario 3: Real-Time Market-Data Processing

The Stocks domain ingests high-frequency market feeds from external exchanges and providers through streaming APIs. Incoming events are validated before entering raw storage, then standardized and enriched through cleansed and business-ready layers before publication as governed data products for trading dashboards, portfolio valuation, risk systems, and ML models. Continuous quality monitoring is important because stale, incomplete, duplicated, or structurally invalid events can immediately affect downstream decisions [9].

Controls include source authentication, secure streaming connections, schema and timestamp validation, duplicate and stale-data detection, price-anomaly checks, and automated capture of source, exchange, API version, ingestion time, ownership, and lineage metadata. Observability monitors latency, throughput, completeness, service availability, and pipeline health. Storage and transformation governance enforce encryption, retention, reference-data mapping, corporate-action adjustments, quality checkpoints, and lineage. Publishing governed data products creates a controlled boundary between curated business data and downstream consumers while preserving domain ownership and enterprise standards [6].

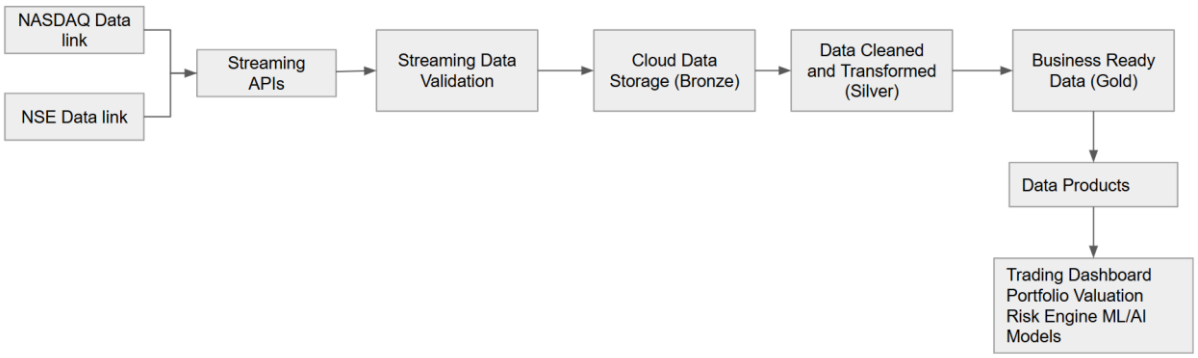


Fig. 4. Representative real-time market-data pipeline and governed data-product flow

7.5 Scenario 4: Regulatory and Business Reporting

Regulatory and business reporting consumes governed Customer, Account, Stock, and Portfolio data products to generate operational, management, and regulatory outputs. This scenario emphasizes consumption and cross-domain governance. Reporting logic is version-controlled and based on standardized business definitions; quality controls reconcile contributing domains before report generation; entitlement checks ensure that recipients are authorized; and audit records capture report generation, access, authorization, and distribution.

Governed data products reduce direct dependence on operational databases and raw storage by providing certified interfaces with ownership, metadata, quality expectations, access policies, and service-level objectives. Cross-domain lineage connects report elements to originating products and transformations, enabling reproducibility, regulatory investigation, and impact analysis. Monitoring detects processing failures, quality exceptions, entitlement violations, and delivery failures.

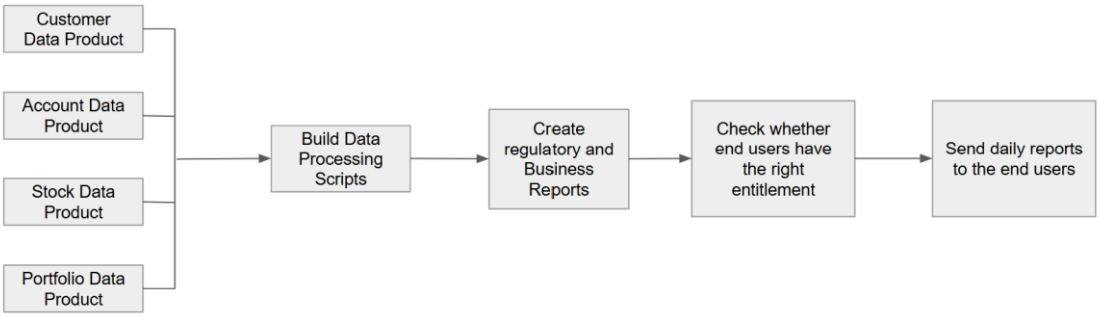


Fig. 5. Representative regulatory and business reporting workflow.

8. FRAMEWORK EVALUATION

8.1 The Evaluation Approach

The four representative workflows were examined using a common analytical procedure: identify the data lifecycle stages, enumerate governance risks and associated controls, classify each control by enforcement mode and ownership paradigm, and assess its plausible automation maturity. The unit of analysis is the governance mechanism, not the workflow as a whole. The scenarios are illustrative design cases, not observed production deployments; consequently, the findings describe classification coverage and architectural trade-offs rather than measured performance gains.

8.2 Cross-Scenario Findings

Customer onboarding emphasizes preventive identity, schema, consent, and sanctions checks at ingestion, complemented by detective duplicate and quality checks after persistence. Stock transactions prioritize synchronous authorization, eligibility, balance validation, and idempotency at API and compute boundaries, with audit and reconciliation performed downstream. Market-data streaming combines lightweight schema and timestamp checks with continuous detection of stale, duplicate, and anomalous events across ingestion and transformation. Regulatory reporting relies on governed consumption, cross-domain reconciliation, entitlement enforcement, reproducible transformations, and evidence capture. These distinctions demonstrate that a shared classification can represent different combinations of lifecycle layer, enforcement mode, governance ownership, and automation maturity without assigning one fixed configuration to all workflows.

8.3 Representative Failure-Condition Analysis

To probe applicability beyond nominal processing, four illustrative exceptions were considered. An onboarding request with missing identity attributes or incomplete KYC status calls for preventive rejection or controlled exception handling, with subsequent audit evidence. A repeated stock purchase request calls for idempotency checks and transaction-state validation before a duplicate execution can occur. A delayed or out-of-order market-price event calls for timestamp validation and detective freshness monitoring, followed by flagging, quarantine, or controlled replay as appropriate. A reporting request involving unreconciled source records or an unauthorized recipient calls for reconciliation and access enforcement before distribution. These cases are analytical expectations derived from the proposed controls, not experimentally observed detection rates or service-level outcomes.

8.4 Interpretation and Limitations

The comparison indicates that governance requirements are context-dependent yet remain comparable when lifecycle location, enforcement behavior, ownership, and automation are assessed separately. It also exposes an engineering trade-off: preventive controls protect high-risk transactions but may increase synchronous processing overhead, whereas detective and corrective controls can preserve throughput at the cost of delayed identification or remediation. The study does not measure detection coverage, false-positive rates, policy-enforcement latency, throughput impact, operational effort, or recovery time. Future work should validate the taxonomy against additional datasets and operational environments, using controlled policy violations and these quantitative measures. No claim of quantitative improvement or superiority of one maturity level follows from this qualitative evaluation.

TABLE III. Comparative classification of the four fintech scenarios by governance layer, enforcement mode, control paradigm, and automation maturity

Scenario	Primary Layers	Enforcement	Control Paradigm	Automation
Customer onboarding	Ingestion / Storage	Preventive + Detective	Federated	L2-L3
Stock transaction	Ingestion / Compute	Preventive	Federated	L3
Market-data streaming	Ingestion / Compute / Storage	Preventive + Detective	Decentralized / Federated	L2-L3
Regulatory reporting	Consumption	Preventive + Detective	Centralized / Federated	L2-L3

9. OPEN CHALLENGES

9.1 Real-Time Governance for Streaming Systems

Streaming platforms require schema validation, policy evaluation, lineage capture, quality assessment, and security controls to operate at data velocity without unacceptable processing overhead. Existing work supports continuous streaming-quality monitoring [9], but broader low-latency governance remains an open problem. Future work should evaluate stream-aware policy engines, incremental lineage, and

adaptive enforcement, while explicitly considering the latency and throughput constraints characteristic of distributed data-intensive systems [24].

9.2 Cross-Domain Policy Consistency

Federated architectures improve domain autonomy but create risks of inconsistent metadata, access models, policy definitions, and regulatory interpretation. Research is needed on interoperable policy models, semantic metadata, governance APIs, and synchronization mechanisms that preserve local ownership while maintaining enterprise-wide consistency [3,7].

9.3 Explainable Governance Decisions

As enforcement becomes more automated, governance decisions must remain traceable and defensible. Access approvals, classifications, retention decisions, and quality assessments should be supported by policy provenance, decision lineage, audit trails, and human-readable explanations. Explainability becomes especially important when AI contributes to governance decisions.

9.4 AI-Driven Governance and ML/AI Pipelines

AI can assist with data classification, anomaly detection, policy recommendations, and control monitoring, but introduces concerns about reliability, explainability, accountability, and regulatory acceptance. At the same time, ML/AI pipelines require governance of training data, features, model versions, inference services, performance, bias, and lifecycle decisions. Responsible AI principles must therefore be integrated with data governance across design, execution, monitoring, and evaluation [18]. Future frameworks should connect source-data lineage to feature engineering, model training, deployment, and inference while preserving human oversight.

10. CONCLUSION

Cloud-native architectures, distributed data ownership, real-time processing, and AI-enabled systems are changing how governance is designed in financial institutions. This paper proposed a multi-dimensional taxonomy that separates governance into four complementary dimensions: lifecycle layer, enforcement mode, control paradigm, and automation maturity. The taxonomy is supported by a four-level automation model and a reference architecture that embeds metadata, lineage, policy enforcement, quality, security, and auditability throughout the data lifecycle.

Application of the taxonomy to four representative fintech scenarios demonstrates that governance priorities vary by workflow: onboarding emphasizes privacy, identity, and regulatory controls; transaction processing requires low-latency API governance; market-data processing requires continuous quality, lineage, and observability; and regulatory reporting requires controlled consumption, entitlement management, reproducibility, and cross-domain traceability. Despite these differences, each scenario can be classified using the same dimensions, supporting the taxonomy's value as a common analytical structure.

The study is limited by its conceptual and scenario-based evaluation rather than a production deployment. Future empirical work should measure governance coverage, policy-violation detection, data-quality improvement, enforcement latency, and computational overhead. Additional research is needed on streaming governance, cross-domain policy consistency, explainable enforcement, AI-assisted governance, and integrated governance of ML/AI pipelines. The proposed taxonomy provides a foundation for reasoning about these challenges and for designing governance architectures suited to distributed, real-time, and increasingly intelligent financial data platforms.

11. ACKNOWLEDGEMENT

During the preparation of this manuscript, the author used OpenAI ChatGPT (GPT-5 series) for language refinement, manuscript organization, and assistance with the presentation of selected concepts and diagrams. The author reviewed and edited the generated output and takes full responsibility for the content of this publication. Zotero was used for reference management and bibliography preparation.

12. REFERENCES

[1] D. Acev, S. Biyani, F. Rieder, T.T. Aldenhoff, M. Blazevic, D.M. Riehle, M.A. Wimmer, "Systematic analysis of data governance frameworks and their relevance to data trusts," *Manag Rev Q*, Jul. 2025, doi: 10.1007/s11301-025-00545-1.

[2] T. Watanabe and T. Mima, "Data Governance," in *The Architecture of "Society 5.0,"* Hitachi-UTokyo Laboratory (H-UTokyo Lab.), Ed., Singapore: Springer Nature Singapore, 2025, pp. 79–90. doi: 10.1007/978-981-96-2929-9_6.

[3] D. W. Arner, G. G. Castellano, and E. K. Selga, "Financial Data Governance," *Hastings Law Journal*, vol. 74, no. 2, pp. 235–292, 2023.

[4] I. Acuña, A. Cravero, and M. Zorrilla, "From Compliance to Value: Data Governance Implementation Challenges to Turn Data into a Strategic Asset—A Systematic Literature Review," *Applied Sciences*, vol. 16, no. 11, p. 5289, May 2026, DOI: 10.3390/app16115289.

[5] Z. Dehghani and M. Fowler, *Data mesh: delivering data-driven value at scale*, First edition. in Software architecture. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2022.

[6] A. Dolhopolov, A. Castelltort, and A. Laurent, "Implementing Federated Governance in Data Mesh Architecture," *Future Internet*, vol. 16, no. 4, p. 115, Mar. 2024, DOI: 10.3390/fi16040115.

[7] T. Van Eijk, I. Kumara, D. Di Nucci, D. A. Tamburri, and W.-J. Van Den Heuvel, "Architectural Design Decisions for Self-Serve Data Platforms in Data Meshes," in *2024 IEEE 21st International Conference on Software Architecture (ICSA)*, Hyderabad, India: IEEE, Jun. 2024, pp. 135–145. DOI: 10.1109/ICSA59870.2024.00021.

[8] A. Polimeno, C. Braghin, M. Anisetti, and C. A. Ardagna, "Maximizing data quality while ensuring data protection in service-based data pipelines," *J Big Data*, vol. 12, no. 1, p. 62, Mar. 2025, DOI: 10.1186/s40537-025-01118-5.

[9] E. Costa E Silva, O. Oliveira, and B. Oliveira, "Enhancing Real-Time Analytics: Streaming Data Quality Metrics for Continuous Monitoring," in *Proceedings of the 2024 7th International Conference on Mathematics and Statistics*, Amarante Portugal: ACM, Jun. 2024, pp. 97–101. DOI: 10.1145/3686592.3686609.

[10] M. Ahmad, J. J. Geewax, A. Macvean, D. Karger, and K.-L. Ma, "API Governance at Scale," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, Lisbon Portugal: ACM, Apr. 2024, pp. 430–440. DOI:

10.1145/3639477.3639713.

[11] Y. Chen, Y. Zhao, X. Li, J. Zhang, J. Long, and F. Zhou, "An open dataset of data lineage graphs for data governance research," *Visual Informatics*, vol. 8, no. 1, pp. 1–5, Mar. 2024, DOI: 10.1016/j.visinf.2024.01.001.

[12] R. Mahanti, "Data Governance Maturity Models," in *Data Governance Success*, Singapore: Springer Singapore, 2021, pp. 99–126. DOI: 10.1007/978-981-16-5086-4_4.

[13] L. Guerreiro, J. Martins, M. Do Rosário Bernardo, H. Mamede, and F. Branco, "Impact of a Master Data Management Framework to Trigger Data Governance Maturity: A Systematic Literature Review," *IEEE Access*, vol. 13, pp. 190644–190667, 2025, DOI: 10.1109/ACCESS.2025.3627855.

[14] J. Merkus, R. Helms, and R. Kusters, "Data Governance Capabilities: Maturity Model Design with Generic Capabilities Reference Model:," in *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Online Streaming: SCITEPRESS - Science and Technology Publications, 2021, pp. 102–109. Doi: 10.5220/0010651300003064.

[15] B. Marr, *Data strategy: how to profit from a world of big data, analytics and artificial intelligence*, Second edition. London New York, NY: Kogan Page, 2022.

[16] D. Henderson, S. Earley, C. Bradley, and Data Administration Management Association, Eds., *DAMA-DMBOK: data management body of knowledge*, Second edition, Revised, First printing. Basking Ridge, New Jersey: Technics Publications, 2024.

[17] P. L. Foalem, L. D. Silva, F. Khomh, and E. Merlo, "An empirical study of Policy-as-Code adoption in open-source software projects," *Journal of Systems and Software*, vol. 242, p. 113028, Dec. 2025, doi: 10.1016/j.jss.2026.113028.

[18] E. Papagiannidis, P. Mikalef, and K. Conboy, "Responsible artificial intelligence governance: A review and research framework," *The Journal of Strategic Information Systems*, vol. 34, no. 2, p. 101885, Jun. 2025, doi: 10.1016/j.jsis.2024.101885.

[19] *California Consumer Privacy Act of 2018*, vol. §§ 1798.100–1798.199.100. 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[20] *Gramm-Leach-Bliley Act*, vol. 15 U.S.C. §§ 6801–6809, 6821–6827. 1999. [Online]. Available: https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act

[21] *Regulation (EU) 2016/679 (General Data Protection Regulation)*, vol. L 119. 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[22] "Payment Card Industry Data Security Standard: Requirements and Testing Procedures," PCI Security Standards Council, Wakefield, MA, Version 4.0.1, Jun. 2024. [Online]. Available: https://www.pcisecuritystandards.org/document_library/?Class=pcidss&doc=pci_dss

[23] J. Ladley, *Data governance: how to design, deploy and sustain an effective data governance program*, Second edition. Academic Press, 2020.

[24] M. Kleppmann, *Designing data-intensive applications: the big ideas behind reliable, scalable, and maintainable systems*, First edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2017.