Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

From Centralized Data Lakes to Data Mesh: Lessons from Large-Scale Enterprise Transformations

Naveena Kumari Nandale Vadlamudi
Independent Researcher, USA

ARTICLE INFO	ABSTRACT

Centralized data lake architectures have long served as foundational infrastructure supporting enterprise analytics capabilities across diverse organizational contexts. Contemporary organizations, however, face increasingly significant challenges inherent in monolithic data models, including unclear ownership boundaries, diminished governance agility, and progressively degraded data quality as organizational scale expands beyond the capacity of centralized management paradigms. Domain-oriented data mesh architectures offer viable alternatives to these rigid centralized frameworks by fundamentally shifting architectural philosophy from technology-centric consolidation strategies toward domain-driven distribution models that align data ownership with organizational expertise boundaries. This article comprehensively addresses the structural deficiencies inherent in consolidated repository architectures while articulating the paradigm shift that distinguishes data mesh implementations from incremental improvements to existing centralized models. The data-as-a-product conceptualization receives detailed analytical attention alongside federated governance mechanisms that enable distributed policy enforcement without creating centralized bottlenecks. Self-serve infrastructure platform requirements form critical components of successful transformations, providing domain teams with standardized capabilities while maintaining autonomy in data product development and operation. Metadata-driven policy enforcement combined with standardized interoperability contracts enables organizational scalability without compromising stringent compliance standards across regulatory domains. Migration strategies supporting incremental transformation while preserving operational stability throughout transition periods constitute essential considerations for enterprise adoption, as organizations cannot afford disruption to critical analytical workloads during architectural evolution. The architectural philosophy underlying data mesh implementations strategically rebalances centralized control mechanisms with domain autonomy through explicit separation of platform capabilities from domain-specific responsibilities. A comprehensive enterprise transformation case study demonstrates practical implementation patterns across a multi-year migration, providing empirical evidence of achievable outcomes. Architectural diagrams visualize structural distinctions between paradigms, while a detailed maturity assessment framework enables organizations to evaluate transformation preparedness systematically. Data mesh represents an evolutionary advancement in enterprise data management rather than a complete infrastructure replacement requiring abandonment of existing investments. Enterprise-scale data platforms increasingly benefit from

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

distributed ownership models aligned with business capability boundaries, enabling horizontal scaling that grows naturally with organizational expansion. Federated computational governance ensures consistency across organizational domains without creating the centralized bottlenecks that characterize traditional governance approaches. The synthesis illustrates pathways through which organizations can achieve sustainable data architecture capable of accommodating growing operational complexity while maintaining quality standards and regulatory compliance.

Keywords: Data Mesh Architecture, Federated Data Governance, Domain-Oriented Ownership, Self-Serve Data Platforms, Enterprise Data Transformation, Interoperability Standards

I. Introduction

Data lakes emerged as architectural responses to fundamental limitations inherent in traditional data warehouse paradigms, which struggled to accommodate the volume, velocity, and variety characteristics of contemporary data ecosystems. The data lake paradigm fundamentally transformed data management by storing raw information in native formats without predetermined schema requirements, thereby enabling flexible consumption patterns aligned with diverse analytical needs. This architectural approach facilitates the unified storage of structured, semi-structured, and unstructured data within consolidated repositories that defer transformation decisions until consumption time. Pwint Phyu Khine and Zhao Shun Wang from the University of Science and Technology Beijing, China define data lakes as centralized storage systems that accommodate diverse data types while implementing schema-on-read strategies rather than the rigid schema-on-write requirements characteristic of traditional warehouse architectures [1]. The schema-on-read approach provides substantially greater flexibility compared to the predetermined structure requirements that constrained earlier warehouse implementations, enabling organizations to capture diverse data without upfront schema design commitments. Organizations across industries have widely adopted data lake architectures to consolidate data from varied operational sources into unified repositories, motivated by promises of simplified data discovery mechanisms and reduced infrastructure complexity through centralization. However, this consolidation approach simultaneously promised benefits while introducing structural tensions that would only become apparent at organizational scale.

The inherently centralized nature of data lake architectures creates architectural tensions that intensify as organizational complexity increases beyond the coordination capacity of central management teams. Single repository models inherently require central platform teams to manage ingestion pipelines, quality validation processes, and access control mechanisms across all organizational domains, creating coordination dependencies that grow proportionally with both data volume and source system diversity. Domain experts who generate source data within operational contexts increasingly operate at substantial distance from the consumption contexts where data ultimately drives analytical insights and business decisions. This organizational separation between data producers embedded in operational workflows and repository managers focused on technical infrastructure introduces significant coordination overhead that compounds as organizations scale. Quality issues propagate through lengthy pipeline chains before detection mechanisms can identify problems, while central teams inevitably lack the deep contextual knowledge required for effective data stewardship across disparate business domains. The structural separation of ownership from expertise creates accountability gaps where no single entity possesses both the authority and knowledge necessary for ensuring data quality aligned with domain-specific requirements.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Data mesh architecture addresses these fundamental structural limitations through comprehensive reorganization of data responsibilities that aligns ownership with domain expertise rather than technical specialization. Abhijit Joshi from the Enterprise Architecture Research Group describes data mesh as a decentralized sociotechnical approach that strategically distributes data ownership to domain teams possessing relevant contextual expertise for effective stewardship [2]. Unlike incremental improvements to centralized architectures that preserve existing organizational structures while upgrading underlying technologies, data mesh represents a fundamental paradigm shift in how organizations conceptualize data management as an organizational design challenge rather than merely a technical infrastructure problem. Traditional centralized approaches frame data management as primarily technical infrastructure challenges requiring specialized platform teams with deep technology expertise but limited domain knowledge. Data mesh comprehensively reframes data management as organizational design challenges requiring distributed sociotechnical solutions that align technical capabilities with business domain boundaries, thereby positioning domain teams as accountable owners rather than passive data producers serving central repositories.

Four foundational principles collectively define the data mesh paradigm and distinguish it from prior architectural approaches. Domain-oriented ownership assigns data product responsibility to teams possessing contextual domain knowledge essential for effective stewardship, establishing clear accountability boundaries aligned with business capabilities. Data-as-a-product thinking establishes explicit quality and usability expectations through service-level commitments, transforming datasets from passive repositories into actively managed products with defined consumer interfaces. Self-serve infrastructure platforms enable autonomous domain operation by providing standardized technical capabilities without creating dependencies on centralized teams for routine operations. Federated computational governance ensures organizational consistency through automated policy enforcement distributed across domains rather than manual centralized review processes that create bottlenecks. The fundamental distinction between data mesh and prior architectural approaches warrants explicit articulation to prevent misunderstanding of the paradigm shift involved. Centralized architectures position dedicated data platform teams as service providers responding reactively to organizational data needs articulated through request processes. Data mesh fundamentally repositions domain teams as proactive product owners bearing direct accountability for data quality, usability, and consumer satisfaction. Centralized models achieve scaling through vertical expansion of central team capacity and infrastructure investment, creating linear growth constraints as organizational complexity increases. Data mesh enables horizontal scaling through distributed ownership that grows organically with organizational expansion, as new domains naturally assume responsibility for their data products without proportionally increasing central coordination burden. This fundamental reorientation of organizational responsibilities distinguishes data mesh from technology-focused solutions that preserve centralized organizational structures while merely upgrading technical components.

The architectural transition from centralized data lakes toward distributed mesh implementations requires carefully coordinated evolution across both technical infrastructure and organizational capability dimensions. Existing infrastructure investments representing substantial sunk costs must be preserved and leveraged during transitions rather than abandoned through disruptive replacement strategies. Domain teams require systematic capability development to effectively assume data product ownership responsibilities that extend beyond their traditional operational focus. Platform teams must evolve their role from direct data management toward providing abstracted infrastructure capabilities that enable autonomous domain operation without ongoing dependencies. Governance frameworks must achieve a delicate balance between standardization requirements ensuring organizational consistency and domain flexibility enabling contextual adaptation. Migration strategies must maintain unwavering operational

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

continuity throughout extended transformation periods, as business-critical analytical workloads cannot tolerate disruption during architectural evolution.

This article systematically examines lessons emerging from enterprise-scale data mesh transformations, synthesizing practical implementation experience with theoretical foundations. The analysis comprehensively addresses centralized architecture limitations that motivate paradigm shifts toward distributed models, providing empirical evidence of the problems organizations encounter at scale. Foundational mesh principles and detailed implementation requirements receive thorough examination grounded in both theoretical frameworks and practical experience. Federated governance mechanisms and interoperability frameworks constitute core analytical focus areas, as these components prove critical for achieving consistency without centralization. A comprehensive enterprise case study demonstrates practical transformation patterns across a multi-year migration involving fifteen domains and eighty-seven data products. Architectural visualizations clarify structural distinctions between centralized and distributed paradigms through detailed diagrams. A maturity assessment framework provides systematic organizational readiness evaluation criteria enabling transformation planning. Migration strategies enabling incremental evolution while maintaining operational stability complete the investigation scope. The synthesis provides comprehensive architectural guidance for organizations navigating this fundamental transformation in enterprise data management philosophy, enabling informed decision-making based on both theoretical understanding and practical experience.

II. Related Work

Prior scholarly literature addressing enterprise data management has extensively examined centralized data lake architectures and their associated governance challenges across multiple analytical dimensions. Khine and Wang established a foundational conceptual understanding of data lake architectural patterns within big data ecosystems, articulating the schema-on-read paradigm that distinguishes lakes from traditional warehouses [1]. Athira Nambiar and Divyansh Mundra from the Department of Computer Engineering at the National Institute of Technology Karnataka, India conducted comprehensive comparative analysis examining architectural characteristics distinguishing data warehouses from data lakes in contemporary enterprise contexts, highlighting governance complexity as a primary implementation challenge [4]. Michael Stonebraker from the Massachusetts Institute of Technology and Ugur Çetintemel from Brown University fundamentally challenged prevailing monolithic architectural assumptions through critical evaluation of one-size-fits-all database paradigms, arguing that diverse workload requirements necessitate specialized optimization strategies rather than compromise solutions [3]. Joshi introduced decentralized data management principles emphasizing organizational scalability and enhanced agility benefits achievable through distributed ownership models aligned with business capabilities [2].

This article builds substantively upon existing scholarly foundations by synthesizing comprehensive architectural transition patterns from centralized toward federated organizational models. A cohesive framework emerges connecting domain-oriented ownership principles with computational governance mechanisms that enable distributed enforcement of organizational policies. Patricia Cupoli's comprehensive DAMA-DMBOK governance standards provide essential foundational reference for policy automation strategies that enable computational governance at scale [7]. Pooyan Jamshidi and colleagues from Carnegie Mellon University contributed microservices contract patterns directly applicable to data product interoperability requirements, demonstrating how autonomous component evolution can proceed while preserving integration stability [8]. Rick Cattell's systematic examination of scalable data storage architectures informs migration sequencing considerations essential for maintaining operational

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

continuity during transitions [9]. Florian Rademacher and colleagues from the University of Kassel, Germany identified domain-driven design challenges particularly relevant to organizational capability development requirements during mesh adoption [10]. Shantenu Jha from Rutgers University and colleagues examined data-intensive computational paradigms across high-performance computing and Apache Big Data ecosystems, providing empirical performance analysis informing platform abstraction strategies [6].

Key scholarly contributions of this work include comprehensive integration of federated governance principles with self-serve platform enablement strategies that collectively enable distributed ownership at scale. The article presents detailed incremental migration frameworks specifically designed to preserve operational continuity during extended transformation periods. A comprehensive enterprise case study provides empirical evidence of practical implementation patterns across a multi-year transformation involving substantial organizational change. Architectural visualization diagrams clarify structural distinctions between paradigms more effectively than textual description alone. A systematic maturity assessment framework enables transformation readiness evaluation across multiple organizational dimensions. Novel theoretical connections between microservices architecture patterns and data mesh interoperability standards extend current understanding within enterprise data management scholarly discourse, suggesting broader applicability of distributed system principles to data management challenges.

III. Limitations of Centralized Data Lake Architectures

A. Architectural Rigidity and Processing Constraints

Centralized data architectures fundamentally operate under organizational and technical assumptions that have become increasingly problematic as enterprise data ecosystems evolve in complexity and scale. Stonebraker and Çetintemel persuasively argue that monolithic database approaches systematically fail to address the diverse workload requirements effectively, as single architectural patterns cannot simultaneously optimize for fundamentally different processing characteristics [3]. The underlying assumption that unified architectural patterns can adequately serve all organizational processing needs proves increasingly inadequate for modern enterprise environments characterized by heterogeneous workload portfolios. Analytical workloads exploring large datasets through ad-hoc queries demand fundamentally different optimization strategies compared to transactional processing systems prioritizing consistency and durability. Stream processing requirements for real-time data ingestion and analysis directly conflict with batch-oriented pipeline designs optimized for throughput over latency. Machine learning workloads impose distinctive computational patterns emphasizing iterative processing and large-scale matrix operations that differ substantially from traditional query execution models optimized for set-based operations.

Data lake architectures inevitably inherit these architectural tensions despite their schema flexibility advantages over rigid warehouse structures, as centralization forces compromise across competing optimization objectives. Ingestion pipelines carefully optimized for high-throughput batch loading demonstrate substantially degraded performance when confronted with real-time streaming requirements demanding low-latency ingestion. Query engines meticulously designed for exploratory analytical workloads involving full dataset scans struggle significantly when deployed for production serving workloads requiring predictable sub-second response times. Storage formats optimized for compression ratios minimizing storage costs directly conflict with random access patterns requiring rapid retrieval of individual records. Empirical evaluations systematically demonstrate that processing framework misalignment can degrade computational performance by factors ranging from 10 to 100

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

times compared to workload-optimized implementations specifically designed for particular processing patterns [6]. Central platform teams must continuously balance these inherently competing requirements across all organizational domains, making optimization decisions that inevitably create winners and losers. Optimization choices benefiting one domain's performance characteristics frequently degrade performance for other domains with different requirements. The centralized architectural model fundamentally forces compromise solutions attempting to satisfy diverse requirements simultaneously, ultimately satisfying no domain's specific needs completely while preventing domains from implementing specialized optimizations aligned with their unique workload characteristics.

Processing bottlenecks emerge predictably as accumulated data volumes and organizational domain diversity increase beyond the capacity of centralized infrastructure to serve all requirements effectively. Central pipeline processing capacity becomes a shared organizational constraint creating resource competition across organizational units with varying priority levels and business criticality. Priority conflicts arise inevitably when multiple domains simultaneously require processing resources for time-sensitive analytical workloads, necessitating coordination mechanisms that introduce delays. Pipeline failures affecting shared infrastructure components propagate impact across all downstream consumers regardless of domain boundaries or business criticality, creating an organizational blast radius extending beyond the originating failure point. Recovery procedures must coordinate response across disparate business contexts with different stakeholder groups, operational priorities, and success criteria. Central teams inherently lack the organizational authority necessary for making domain-specific prioritization decisions that require deep business context understanding, forcing escalation processes that introduce compounding delays during incident response when rapid resolution proves most critical for business continuity.

B. Governance Complexity and Ownership Ambiguity

Enterprise data management fundamentally requires comprehensive governance frameworks systematically addressing quality assurance, security controls, and regulatory compliance requirements across diverse regulatory domains and geographic jurisdictions. Nambiar and Mundra identify governance complexity as a primary operational challenge in modern data lake implementations, as organizations struggle to maintain consistent policy enforcement across growing data estates [4]. Centralized governance models concentrate policy definition authority, interpretation responsibility, and enforcement execution within dedicated governance teams separated from domain operational contexts. These specialized governance teams must develop and maintain comprehensive expertise spanning compliance requirements across all business domains, regulatory environments, and geographic jurisdictions where the organization operates. Regulatory environments vary substantially across geographic regions through differing privacy regulations, data sovereignty requirements, and industry-specific compliance frameworks. Central governance teams cannot realistically maintain deep expertise spanning all applicable regulatory requirements while simultaneously understanding business context necessary for appropriate policy application, creating systematic gaps in governance effectiveness as organizational scope expands.

Ownership ambiguity significantly compounds governance challenges inherent in centralized architectural approaches by diffusing accountability across extended value chains. Data producers generate operational information within source systems carefully optimized for transactional processing characteristics emphasizing consistency and durability. Central platform teams subsequently extract, transform, and load this operational data into repository structures optimized for analytical access patterns. Analytical consumers ultimately access transformed data through query interfaces and APIs managed by platform engineering teams focused on technical operation rather than business context. Accountability for data quality becomes systematically diffuse across this extended value chain involving multiple organizational handoffs where responsibility transfers between teams with different objectives and expertise. Producers lack meaningful visibility into downstream consumption patterns and quality requirements, preventing them from understanding how their operational data ultimately drives business decisions. Central teams lack essential contextual knowledge necessary for effective quality validation aligned with domain-specific requirements and business rules. Consumers cannot efficiently trace quality issues to root causes through complex pipeline chains involving multiple transformation stages and system boundaries, prolonging resolution times and eroding trust in data assets.

Access control management becomes increasingly operationally complex as consumer populations grow and diversify across organizational boundaries, creating a substantial coordination burden for central governance teams. Central governance teams must systematically evaluate access requests against comprehensive policies spanning diverse regulatory frameworks, geographic restrictions, and data sensitivity classifications. Request backlogs accumulate progressively as evaluation capacity fails to match exponentially growing demand, creating delays that impede analytical initiatives and reduce organizational responsiveness to emerging business opportunities. Delayed access to critical data assets directly impedes analytical initiatives requiring timely insights, reducing organizational agility in competitive environments. Self-service access mechanisms enabling autonomous consumer provisioning require sophisticated governance automation capabilities that centralized organizational models struggle to implement effectively at scale, as manual review processes cannot achieve the throughput necessary for supporting large consumer populations. Policy enforcement remains fundamentally dependent on central team availability, expertise, and processing capacity, creating inherent bottlenecks. Governance increasingly becomes an organizational bottleneck actively constraining productive data utilization across the enterprise rather than enabling compliant usage through streamlined processes.

Metadata management challenges further complicate centralized governance models by requiring continuous catalog maintenance across datasets originating from disparate domains with different semantic conventions. Data catalogs require continuously accurate documentation of schema definitions, quality characteristics, lineage relationships, and usage patterns to remain valuable for discovery purposes. Central catalog maintenance teams must sustain documentation accuracy across extensive datasets originating from disparate domains with different business terminologies and semantic conventions. As organizational distance increases between documentation maintainers focused on catalog infrastructure and domain experts possessing business context, documentation quality correspondingly degrades through accumulation of staleness and inaccuracy. Consumers increasingly encounter stale metadata failing to reflect current schema structures or inaccurate documentation misrepresenting actual data characteristics, progressively leading to diminished trust in catalog information and abandonment of discovery mechanisms. Discovery processes become fundamentally unreliable when catalog information fails to accurately reflect current data states, forcing consumers to rely on informal tribal knowledge networks rather than systematic discovery infrastructure.

Limitation Category	Challenge Area	Impact on Enterprise Operations
Architectural Rigidity	Single processing paradigm	Inability to optimize for diverse workload types
Processing Constraints	Shared pipeline capacity	Resource conflicts across organizational domains
Ownership Ambiguity	Separated producer-consumer accountability	Progressive data quality degradation
Governance Bottlenecks	Centralized policy enforcement	Delayed access and reduced organizational responsiveness

Metadata Management	Distant domain expertise	Outdated documentation and unreliable discovery
Access Control Complexity	Manual request evaluation	Accumulated backlogs impede analytical initiatives

Table 1. Structural Constraints in Monolithic Data Repository Models [3, 4].

IV. Comparative Analysis: Centralized vs. Data Mesh Architectures

The architectural distinction between centralized data lake implementations and distributed data mesh paradigms manifests across multiple fundamental dimensions encompassing organizational structure, technical architecture, and operational characteristics. Understanding these systematic differences proves essential for organizations evaluating potential transformation pathways and assessing alignment with organizational objectives. Table 2 presents a comprehensive systematic comparison highlighting fundamental distinctions in design philosophy, organizational accountability models, scaling characteristics, and governance approaches that collectively define the paradigm shift from centralized to distributed data management

Dimension	Centralized Data Lake	Data Mesh Architecture
Design Philosophy & Architecture	Technology-centric infrastructure consolidation with monolithic unified repository	Domain-centric organizational alignment with federated network of autonomous data products
Ownership & Accountability	Central platform team manages all data assets; diffuse responsibility across producer-platform-consumer chains	Distributed ownership by domain teams with contextual expertise; clear domain accountability for quality
Scaling & Change Velocity	Vertical scaling through central team capacity expansion; systematically bottlenecked by coordination requirements	Horizontal scaling through distributed domain ownership; parallel domain evolution with minimal cross-domain coordination
Governance & Compliance	Centralized policy definition and manual enforcement creating bottlenecks; manual review processes and approval workflows	Federated policy definition with computational enforcement; automated policy validation embedded in infrastructure
Processing & Optimization	Compromise solutions attempting to serve all domains simultaneously	Domain-specific optimization aligned with workload requirements; modern frameworks achieve 10-100x performance improvements [6]
Data Access & Consumer Experience	Central team evaluates all access requests through manual review; single access point with variable quality	Automated policy-based access with domain stewardship; standardized interfaces with consistent quality guarantees
Interoperability & Standards	Implicit through shared repository structure and common schemas	Explicit through standardized service contracts defining syntactic, semantic, and operational characteristics
Infrastructure & Dependencies	High operational dependency on central platform team for all data operations	Self-serve platform capabilities enabling autonomous domain operation without ongoing dependencies

Table 2: Architectural Comparison: Centralized Data Lakes vs. Data Mesh Paradigms

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

This comprehensive comparative framework systematically illustrates the fundamental paradigm shift distinguishing data mesh implementations from incremental improvements to centralized architectural models. The architectural transition addresses deep structural limitations inherent in centralized approaches rather than merely optimizing existing centralized models through technology upgrades or process refinements. Organizations must systematically evaluate these dimensions against their specific operational complexity, anticipated growth trajectories, governance requirements, and organizational culture when seriously considering transformation initiatives. The decision to pursue data mesh adoption represents a strategic commitment to distributed organizational models rather than simply a technology platform selection.

V. Foundational Principles of Data Mesh Architecture

A. Paradigm Shift from Technology-Centric to Domain-Centric Design

Data mesh represents a fundamental philosophical departure from conventional data architecture thinking rather than merely an incremental technology improvement or platform upgrade building upon existing approaches. Traditional centralized approaches consistently frame data management primarily as technical infrastructure challenges requiring specialized platform teams with deep technology expertise but necessarily limited domain knowledge. Data mesh comprehensively reframes data management as fundamentally organizational design challenges requiring careful alignment between data ownership responsibilities and business domain expertise boundaries. This paradigm shift fundamentally distinguishes data mesh implementations from previous architectural evolutions that preserved centralized organizational structures and control hierarchies while simply upgrading underlying technologies or implementing newer platform components. The distinction between technology-centric and domain-centric approaches extends far beyond surface-level organizational chart modifications to encompass fundamental changes in accountability structures, decision-making authority, and resource allocation patterns.

The architectural distinction between paradigms manifests clearly across multiple interconnected dimensions affecting both technical implementation and organizational operation. Centralized architectures systematically create monolithic data planes where all organizational data inevitably flows through unified technical infrastructure managed by central teams. Data mesh deliberately establishes distributed data plane architectures where domain-specific data products operate as autonomous architectural nodes within carefully coordinated federated ecosystems. Centralized models explicitly optimize for technical efficiency metrics through consolidated infrastructure enabling economies of scale in technology operations. Data mesh intentionally optimizes for organizational effectiveness through distributed accountability precisely aligned with business capability boundaries, accepting some technical redundancy in exchange for organizational agility. The optimization objective shift from pure technical efficiency toward organizational effectiveness represents a fundamental reorientation of architectural priorities reflecting mature understanding that business value derives from effective domain operation rather than infrastructure optimization alone.

Scaling characteristics fundamentally differ between paradigms in ways that become increasingly significant as organizations grow in size and complexity. Centralized architectures achieve scaling primarily through vertical expansion mechanisms increasing central team capacity and infrastructure investment proportionally with organizational growth. Organizational growth inevitably creates proportional increases in central team coordination burden as more domains, data sources, and consumers require central team involvement. Data mesh enables horizontal scaling through distributed ownership that grows organically and naturally with organizational expansion without proportionally

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

increasing central coordination requirements. New domains progressively assume direct responsibility for their specific data products without creating corresponding increases in central team workload or coordination complexity. This horizontal scaling model strategically aligns data management organizational capacity with natural organizational growth trajectories, enabling sustainable scaling as enterprises expand geographically, enter new markets, or acquire additional business units.

B. Domain-Oriented Ownership and Organizational Alignment

Data mesh systematically redistributes data management responsibilities to domain teams possessing essential contextual expertise rather than concentrating ownership within central technical teams. This fundamental redistribution carefully aligns data ownership with well-defined business capability boundaries reflecting how organizations naturally structure operational work. Domain teams inherently understand the complex semantics, nuanced quality requirements, typical usage patterns, and business context relevant to their specific data with depth that central teams cannot realistically achieve across all organizational domains. Elena Piskun and colleagues' organizational quality research empirically demonstrates that strategic alignment between assigned responsibility and possessed expertise significantly improves outcome reliability across diverse organizational contexts [5]. Teams bearing direct accountability for data product quality possess the deep contextual knowledge required for effective stewardship including understanding appropriate quality thresholds, recognizing anomalous patterns, and interpreting business rule exceptions. Quality issues receive substantially faster detection and more appropriate resolution when domain experts maintaining direct ownership can leverage their contextual understanding rather than relying on coordination with distant technical teams lacking business context.

Domain boundaries require careful systematic definition to enable effective ownership assignment that avoids gaps and excessive overlaps in responsibility. Business capability mapping exercises identify natural organizational divisions reflecting operational realities and information flow patterns rather than imposing artificial technical boundaries. Data products align strategically with these capability boundaries to ensure coherent ownership where single teams possess both the expertise and authority necessary for effective management. Cross-cutting data requirements spanning multiple domains demand explicit coordination mechanisms between domain teams through federated governance processes. Shared data concepts referenced across domain boundaries require explicit governance agreements carefully specifying ownership responsibilities, evolution procedures, and change notification protocols to prevent coordination failures.

Domain teams must systematically develop comprehensive capabilities for managing data as valuable organizational assets rather than merely producing data as operational byproducts. Accountability structures must provide appropriate support enabling domain ownership effectiveness rather than simply assigning responsibility without corresponding authority or resources. Data product owners bear explicit responsibility for quality maintenance, comprehensive documentation, responsive consumer support, and continuous improvement based on usage feedback. Service level definitions establish clear expectations for critical characteristics including availability guarantees, freshness commitments, and accuracy standards that consumers can rely upon for their analytical workloads. Consumer feedback mechanisms enable continuous improvement cycles based on actual usage experience rather than assumptions about requirements. Performance measurement frameworks systematically track data product health across multiple defined dimensions including quality metrics, usage patterns, and consumer satisfaction. Domain teams require organizational authority commensurate with assigned accountability, including resource allocation decisions, technology selection choices, and operational procedure determination within boundaries established by federated governance.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

C. Self-Serve Data Infrastructure and Platform Enablement

Effective domain ownership fundamentally requires robust infrastructure enabling autonomous operation without creating ongoing dependencies on centralized teams for routine operational activities. Jha and colleagues comprehensively describe platform abstractions that successfully enable data-intensive applications across diverse computational paradigms while maintaining consistent operational characteristics [6]. Self-serve platforms provide carefully standardized capabilities spanning storage services, processing frameworks, cataloging systems, and access management controls that domain teams can readily consume. Domain teams leverage these abstracted capabilities without requiring deep infrastructure expertise or specialized technical knowledge beyond their domain focus. Platform abstractions systematically reduce cognitive overhead for domain teams while simultaneously ensuring architectural consistency across organizational boundaries, enabling both autonomy and coherence.

The platform layer establishes the critical architectural boundary between centralized infrastructure management responsibilities and distributed domain-specific responsibilities, enabling clear separation of concerns. Platform teams maintain standardized infrastructure components serving all organizational domains through reusable abstractions and shared services. Domain teams productively utilize platform capabilities to build and independently operate domain-specific data products aligned with their business requirements. This deliberate separation enables valuable infrastructure specialization and economies of scale without creating problematic domain dependencies on central teams for routine operations. Platform evolution can proceed independently from domain data product development cycles, enabling parallel advancement across architectural layers without tight coordination requirements.

Infrastructure platforms must carefully balance standardization requirements ensuring organizational consistency against flexibility requirements enabling domain-specific optimization. Common capabilities establish essential baseline consistency across domain implementations through shared abstractions and standard interfaces. Storage services provide durable, scalable repositories with appropriate access controls, encryption capabilities, and lifecycle management features. Processing frameworks support diverse transformation, aggregation, and derivation workloads spanning batch, streaming, and interactive processing patterns. Cataloging services maintain searchable metadata repositories enabling efficient data discovery across organizational boundaries. Access management services enforce authentication requirements and authorization policies consistently across all data products.

The selection of appropriate processing frameworks significantly impacts overall platform effectiveness and domain productivity. Jha and colleagues systematically examined performance characteristics across diverse data-intensive computing paradigms, analyzing implementations ranging from traditional MapReduce batch processing to advanced iterative frameworks optimized for machine learning workloads [6]. Their rigorous empirical evaluation of K-Means clustering algorithms across multiple platform implementations demonstrated substantial performance variations driven by architectural choices. Traditional Hadoop MapReduce implementations exhibited execution times ranging from 10 to 100 times slower than optimized alternatives effectively employing efficient collective operations and in-memory processing capabilities reducing intermediate data persistence overhead. Modern frameworks incorporating iterative processing abstractions and optimized communication patterns achieved computational performance approaching specialized high-performance computing implementations while simultaneously maintaining accessible programming models suitable for domain teams without deep systems expertise. These empirical findings underscore the critical importance of platform teams providing diverse processing framework options enabling domain teams to select implementations optimally aligned with their specific workload characteristics rather than forcing compromise solutions.

The contemporary data platform ecosystem encompasses substantial functional diversity enabling rich capability selections. Research systematically analyzing Apache Big Data Stack implementations identified over 110 distinct frameworks and tools supporting varied analytical workloads across storage, processing, and machine learning functional domains [6]. This extensive ecosystem provides domain teams with rich capability selections for building data products addressing diverse analytical requirements spanning batch processing, stream processing, interactive queries, graph analytics, and machine learning model training. Platform teams must carefully curate and standardize subset selections that balance capability breadth enabling diverse use cases against operational complexity and support overhead that grows with technology proliferation. Standardization decisions should reflect actual domain requirements gathered through systematic engagement rather than theoretical capability assessments.

Platform evolution requires continuous capability development systematically aligned with emerging domain needs rather than technology-driven feature additions. Initial platform implementations provide foundational storage and access capabilities enabling basic data product development. Subsequent platform extensions add enhanced processing frameworks, quality monitoring systems, and governance automation capabilities as organizational maturity increases. Advanced capabilities emerge progressively as organizational experience grows and more sophisticated use cases become common. Platform teams must maintain close ongoing engagement with domain consumers to understand emerging requirements through regular feedback sessions, usage pattern analysis, and formal requirement gathering processes. Structured feedback mechanisms ensure platform evolution addresses actual domain needs validated through usage rather than assumed requirements based on industry trends or vendor marketing.

Domain teams require substantial enablement support during capability development periods as they transition from passive data producers to active data product owners. Comprehensive training programs systematically build skills required for effective data product management spanning technical capabilities and organizational competencies. Documentation resources provide essential reference materials for platform capability utilization including quickstart guides, detailed technical specifications, and troubleshooting procedures. Responsive support channels effectively address domain team questions and implementation challenges through multiple mechanisms including chat-based support, ticketing systems, and office hours. Centers of excellence systematically disseminate best practices across domain boundaries through communities of practice, pattern libraries, and case study documentation. Sustained enablement investment significantly accelerates domain team readiness for ownership responsibilities, reducing time-to-productivity and improving initial data product quality.

Principle	Description	Implementation Requirement
Domain-Oriented Ownership	Data responsibility assigned to domain teams	Business capability mapping and boundary definition
Data-as-a-Product	Datasets treated as products with quality guarantees	Service level definitions and consumer support mechanisms
Self-Serve Infrastructure	Abstracted platform capabilities for autonomous operation	Standardized storage, processing, and cataloging services
Accountability Structures	Clear responsibility assignment for data quality	Data product owner roles and performance measurement
Platform Enablement	Reduced cognitive overhead through abstraction	Training programs and documentation resources
Capability Development	Skills building for data product management	Center of excellence functions and peer learning

Table 3. Core Components of Domain-Oriented Data Management [5, 6]

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

VI. Architectural Visualization: Centralized vs. Data Mesh Models

The structural differences between centralized data lake architectures and distributed data mesh implementations warrant explicit visual clarification to complement textual descriptions. Figure 1 systematically illustrates the monolithic architecture characteristic of centralized data lake implementations, while Figure 2 depicts the federated organizational structure of data mesh implementations. These architectural diagrams highlight critical distinctions in data flow patterns, ownership boundary definitions, and governance distribution mechanisms that collectively define the paradigm shift.

A. Centralized Data Lake Architecture

The centralized architectural model consolidates all data ingestion processes, transformation logic, and consumer serving through unified infrastructure managed exclusively by central platform teams. Domain data sources distributed across organizational units feed into shared pipeline infrastructure, creating systematic dependencies and potential bottlenecks concentrated at the central platform layer where all data flows converge.

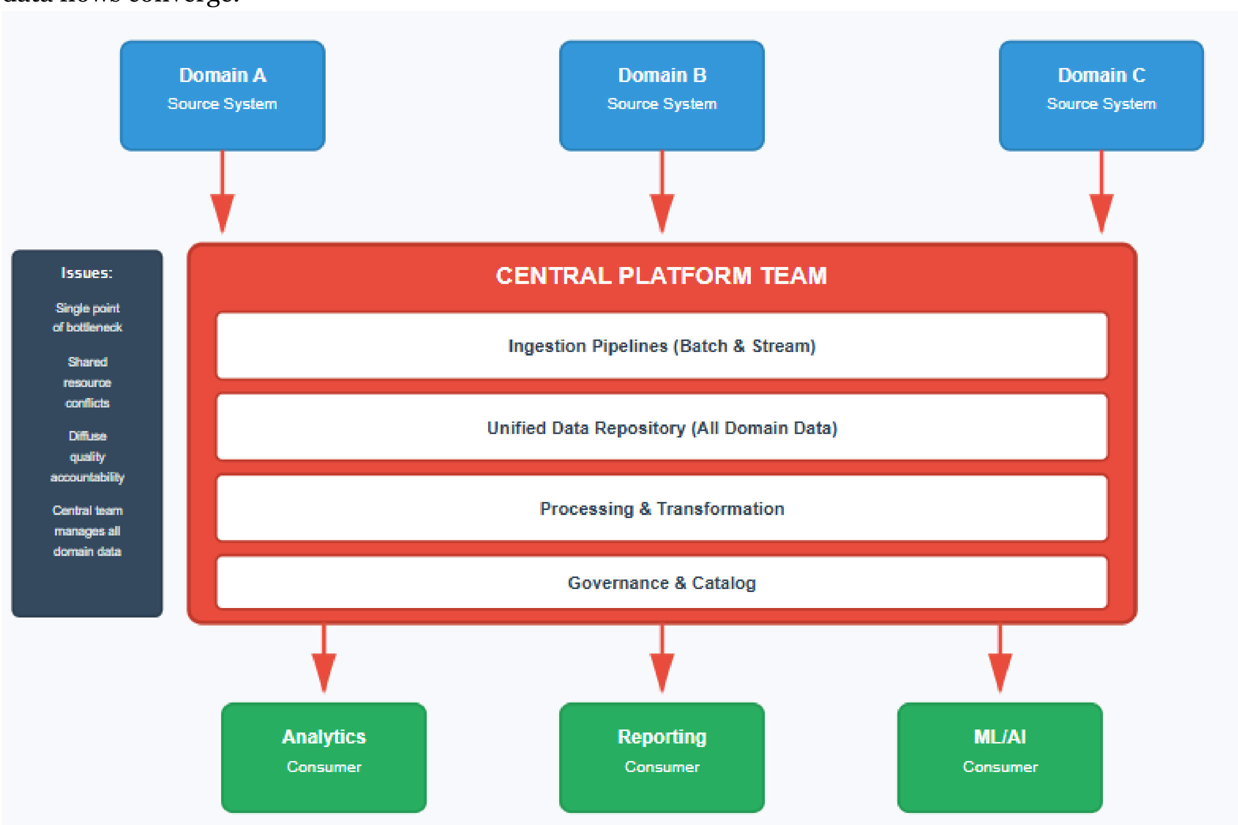

Figure 1: Centralized Data Lake Architecture - Monolithic Structure

B. Data Mesh Architecture

The data mesh architectural model systematically distributes data product ownership to domain teams while simultaneously providing standardized platform capabilities through self-serve infrastructure. Each domain operates with substantial autonomy while maintaining clear accountability, interconnected through federated governance mechanisms and standardized interface contracts.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

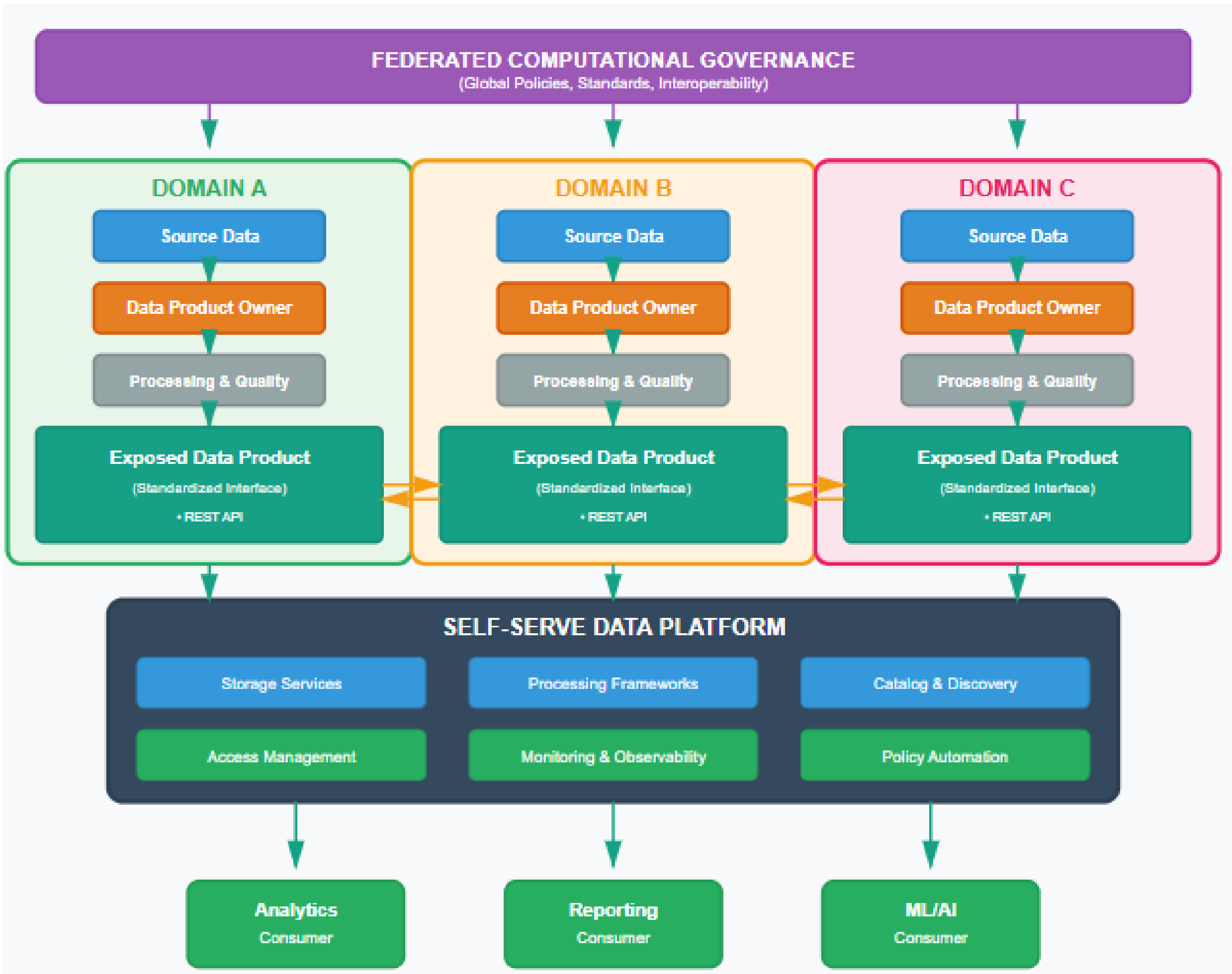

Figure 2: Data Mesh Architecture - Federated Domain Structure

These architectural visualizations systematically clarify the fundamental structural differences between organizational paradigms beyond what textual description alone can convey. The centralized model creates single points of dependency and systematic bottlenecks where all organizational data flows converge, while the mesh model distributes responsibility across autonomous domain teams supported by standardized platform capabilities and federated governance mechanisms that enable consistency without centralization.

VII. Federated Governance and Interoperability Frameworks

A. Computational Governance and Policy Automation

Federated governance strategically distributes policy enforcement execution while simultaneously maintaining essential organizational consistency across domain boundaries through shared standards. Cupoli's comprehensive DAMA Data Management Body of Knowledge establishes foundational frameworks for enterprise data governance spanning quality assurance, security controls, and compliance domains that organizations must systematically address [7]. Traditional governance models rely heavily on manual policy enforcement through review processes and approval workflows requiring human

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

judgment for each decision. These manual governance approaches systematically fail to scale effectively as data product volumes and consumption pattern diversity expand beyond the processing capacity of centralized governance teams. Computational governance fundamentally transforms enforcement by embedding policy validation within automated infrastructure processes that execute consistently without requiring ongoing human intervention for routine compliance verification.

The federated governance model represents another critical dimension fundamentally distinguishing data mesh implementations from centralized architectural approaches beyond the technical infrastructure differences. Centralized governance concentrates policy definition authority, interpretation responsibility, and enforcement execution within dedicated governance teams operating separately from operational domains. Federated governance deliberately separates global policy definition from distributed policy enforcement execution, enabling scalability while maintaining consistency. Global governance bodies comprising cross-domain stakeholders establish minimum standards and interoperability requirements applicable across all organizational contexts. Domain teams implement and enforce these global policies within their specific data products through automated mechanisms. Automated validation mechanisms verify compliance without requiring central team involvement in routine operations, dramatically increasing governance throughput capacity while reducing coordination overhead.

Policy-as-code approaches systematically express governance requirements in machine-executable specifications that infrastructure can automatically evaluate and enforce. Schema validation rules enforce structural consistency requirements at ingestion boundaries, preventing malformed data from entering data products. Quality checks execute automatically during data product update cycles, validating completeness, accuracy, and consistency against defined thresholds. Access control policies evaluate dynamically based on comprehensive requestor attributes including organizational role, geographic location, and data sensitivity classifications. Compliance requirements translate into automated audit logging and retention enforcement mechanisms that execute consistently without manual oversight. Policy violations trigger automated alerts enabling rapid remediation response through notification systems integrated with operational workflows, reducing time from detection to resolution.

Governance automation requires standardized policy definition languages enabling consistent expression and interpretation across organizational domains. Common vocabularies enable consistent policy expression across domain boundaries while accommodating domain-specific extensions where contextually appropriate. Policy repositories maintain version-controlled specifications enabling comprehensive audit trails and evolution tracking over time. Inheritance mechanisms allow global policies to propagate automatically across domain implementations, ensuring baseline consistency while enabling local customization. Override capabilities enable domain-specific policy extensions where business context justifies deviation from global defaults, with appropriate approval workflows. Policy testing frameworks validate enforcement behavior before production deployment, preventing policy errors from impacting operational data products.

Federated governance carefully balances global consistency requirements against domain flexibility needs through layered policy architectures. Global policies establish minimum requirements applicable uniformly across all organizational domains regardless of business context. Data classification standards ensure consistent sensitivity categorization across domains using shared taxonomies. Retention policies comply with regulatory requirements spanning organizational scope including geographic jurisdictions and industry sectors. Security controls meet enterprise risk management thresholds established through formal risk assessment processes. Domain-specific policies extend global requirements based on particular contextual needs justified through business rationale, enabling appropriate flexibility while maintaining baseline organizational consistency.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

B. Interoperability Standards and Service Contracts

Cross-domain data consumption requires carefully standardized interfaces enabling reliable integration while permitting autonomous domain evolution. Jamshidi and colleagues systematically examine service contract patterns that enable autonomous component evolution while preserving integration stability across distributed systems [8]. Data product interfaces define comprehensive consumption contracts specifying schema formats, semantic definitions, and quality guarantees that consumers depend upon. Consumers depend fundamentally on contract stability for reliable analytical workloads with predictable characteristics and consistent quality. Contract evolution must maintain backward compatibility to avoid consumer disruption during domain product updates, requiring careful versioning strategies and deprecation policies.

The interoperability framework establishes rigorous architectural formalization for cross-domain data exchange enabling autonomous domain evolution without breaking consumer integrations. Data products expose standardized interfaces comprising three distinct contract layers addressing different interoperability concerns. Syntactic contracts define structural schemas and access protocols specifying how consumers technically interact with products. Semantic contracts specify business meaning and contextual interpretation ensuring consumers correctly understand data content. Operational contracts establish quality guarantees and availability commitments defining service-level expectations. This carefully layered contract architecture enables independent evolution at each layer while maintaining integration stability across domain boundaries, as changes at one layer do not necessarily require changes at other layers.

Schema standards establish essential structural interoperability across domain boundaries enabling technical integration. Common type systems ensure consistent data representation across diverse technology platforms and implementation languages. Naming conventions enable intuitive schema interpretation reducing cognitive overhead for consumers encountering new data products. Documentation standards specify required metadata for schema elements including descriptions, constraints, and examples. Schema evolution rules govern permitted changes and versioning requirements to prevent breaking consumer integrations. Breaking changes require coordinated migration across affected consumers following established deprecation timelines and communication protocols, ensuring adequate time for adaptation.

Semantic standards address consistency beyond structural compatibility concerns, ensuring consumers correctly interpret data content. Business glossaries define authoritative term definitions preventing semantic ambiguity across organizational boundaries. Data element dictionaries specify precise attribute semantics including measurement units, calculation methodologies, and business rule applications. Reference data standards ensure consistent code value interpretation across domains using shared enumeration definitions. Semantic versioning tracks meaning evolution over time as business definitions change and organizational understanding deepens. Semantic conflicts between domain interpretations require resolution through governance coordination processes involving stakeholders from affected domains, ensuring agreed-upon definitions that all parties can implement consistently.

Quality contracts establish explicit expectations for data product reliability that consumers can depend upon for their analytical workloads. Freshness guarantees specify maximum latency between source changes and product updates, enabling consumers to understand data currency. Completeness standards define acceptable missing value thresholds for critical attributes, informing consumer expectations. Accuracy expectations establish error rate boundaries based on domain requirements and technical capabilities. Availability commitments specify uptime targets and planned maintenance windows, enabling consumers to plan around expected downtime. Quality monitoring continuously validates

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

ongoing contract compliance through automated validation processes, triggering alerts when thresholds are breached and requiring remediation plans from domain product owners.

Governance Aspect	Mechanism	Function
Policy-as-Code	Machine-executable specifications	Automated enforcement at data product boundaries
Schema Validation	Structural consistency rules	Enforcement during ingestion processes
Quality Checks	Automated validation routines	Execution during data product update cycles
Access Control	Dynamic policy evaluation	Authorization based on requester attributes and sensitivity
Semantic Standards	Business glossaries and data dictionaries	Consistent meaning interpretation across domains
Quality Contracts	Service level agreements	Freshness, completeness, and accuracy expectations

Table 4. Computational Governance Mechanisms for Data Mesh Environments [7, 8].

VIII. Data Mesh Maturity and Readiness Assessment Framework

Successful data mesh transformation requires systematic evaluation of organizational readiness across multiple interconnected dimensions. The following maturity framework enables organizations to assess current capabilities and identify development priorities requiring investment before initiating transformation efforts.

A. Maturity Model Structure

Maturity Dimension	Level 1: Initial	Level 2: Developing	Level 3: Defined	Level 4: Managed	Level 5: Optimizing
Domain Ownership & Accountability	No clear data ownership structures; central team manages all organizational data reactively	Some domains identified but ownership boundaries remain unclear with ambiguous accountability	Clear domain boundaries with assigned owners bearing explicit accountability	Full domain accountability for data products with comprehensive service-level commitments	Consumer-driven continuous improvement cycles with proactive quality management
Platform & Technical Capabilities	Predominantly manual processes; no standardized tooling or shared infrastructure	Basic automation in isolated domains with emerging shared infrastructure	Self-serve platform providing core services enabling autonomous domain operation	Comprehensive monitoring, observability, and advanced features supporting diverse use cases	AI-driven optimization and cognitive assistance providing intelligent recommendations
Governance &	Manual	Basic	Federated	Policy-as-code	Predictive quality

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Quality	approval processes; reactive quality management addressing issues after detection	automated checks and quality metrics tracked inconsistently	policies with proactive quality monitoring and automated validation	with automated SLA enforcement and comprehensive compliance validation	management and adaptive governance anticipating issues before occurrence
Interoperability & Culture	Inconsistent interfaces with custom integrations; siloed teams resisting collaboration	Emerging interface standards; growing awareness of collaboration benefits	Documented contracts for all data products; established cross-functional teams	Automated contract testing; embedded data-as-product mindset across organization	Dynamic contract evolution with backward compatibility; innovation culture supporting experimentation

Table 5: Data Mesh Organizational Maturity Framework

B. Readiness Assessment Criteria

Organizations should systematically evaluate readiness across five critical dimensions. Organizational readiness requires executive sponsorship with sustained commitment to multi-year transformation, clear business domain structures with defined capability boundaries, established cross-functional collaboration patterns, and willingness to redistribute authority to domain teams. Technical readiness encompasses cloud infrastructure enabling platform abstraction, established DevOps practices with CI/CD pipelines, API-first architecture patterns, comprehensive observability tooling, and operational data catalog capabilities. Domain team readiness demands identified data product owners with clear accountability, adequate technical skills for data modeling and pipeline development, understanding of consumer needs with service orientation, and access to training and center of excellence resources. Governance readiness includes documented data policies, regulatory requirement understanding, policy automation capabilities or development roadmap, cross-domain coordination mechanisms, and comprehensive metrics frameworks. Migration readiness necessitates complete inventory of existing data assets and dependencies, thorough risk assessments, parallel operation capabilities, well-defined rollback procedures, and clear success criteria with measurable thresholds.

C. Maturity Progression Strategies

Organizations typically progress through four distinct stages. Foundation stage (Levels 1-2) establishes domain boundaries through business capability mapping, selects pilot domains with high readiness and low risk, develops minimum viable platform with core capabilities, documents baseline governance policies, and builds initial training programs. Expansion stage (Levels 2-3) onboards additional domains with refined processes, expands platform capabilities based on domain feedback, implements computational governance for critical policies, establishes contract standards for interfaces, and scales training programs organization-wide. Optimization stage (Levels 3-4) achieves comprehensive domain coverage, implements advanced platform features including ML support, deploys automated quality monitoring, establishes federated governance with minimal central coordination, and embeds data-as-product mindset across culture. Innovation stage (Levels 4-5) enables continuous experimentation, implements predictive quality management, deploys adaptive governance with dynamic optimization, achieves seamless contract evolution, and establishes data mesh as competitive differentiator. Regular

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

maturity assessments enable tracking advancement and identifying capability gaps requiring targeted investment.

IX. Enterprise Case Study: Multi-Year Data Mesh Transformation

A. Initial State and Transformation Motivation

A global financial services organization with operations spanning forty-three countries operated a centralized data lake architecture supporting comprehensive enterprise analytics infrastructure for seven years following initial deployment. The consolidated infrastructure systematically aggregated data from over forty distinct source systems across retail banking operations, wealth management services, and corporate banking divisions serving diverse client segments. The central data platform team consisting of sixty-five specialized engineers managed ingestion pipeline operations, data quality validation processes, catalog maintenance activities, and consumer access provisioning for approximately two thousand analytical users distributed across business divisions and geographic regions.

Growing operational challenges increasingly motivated serious transformation evaluation as existing architectural limitations became progressively more apparent. Pipeline backlogs for new data source onboarding averaged six weeks from initial request to production availability, severely constraining analytical initiative velocity and preventing rapid response to emerging business opportunities. Quality incident resolution required extensive coordination across producer teams generating source data, platform engineers managing infrastructure, and consumers depending on data products, averaging fourteen days from initial detection to complete remediation. Access request backlogs accumulated progressively as manual evaluation capacity failed to match growing consumer demand driven by expanding analytical capabilities and broader data democratization initiatives. Metadata staleness reached alarming forty percent of catalog entries according to systematic audits, substantially diminishing discovery reliability and forcing consumers to rely on informal knowledge networks. Central team headcount growth consistently failed to match organizational expansion rate, creating fundamentally unsustainable scaling trajectory as coordination burden increased faster than team capacity.

Executive leadership commissioned comprehensive architectural assessment examining viable alternatives to centralized model limitations that increasingly constrained business agility. The systematic evaluation identified data mesh principles as closely aligned with existing organizational structure and strategic growth objectives requiring global expansion and acquisition integration. Business divisions already operated as substantially autonomous capability centers with well-established accountability frameworks and profit-and-loss responsibility. Technology platform investments supporting containerized workloads and API-first integration patterns provided technical foundations for distributed operation. DevOps maturity across application development teams enabled distributed team autonomy through established continuous integration and deployment practices. These favorable organizational and technical foundations collectively supported strong data mesh readiness assessment, justifying substantial transformation investment.

B. Transformation Strategy and Phased Implementation

Transformation planning deliberately adopted incremental phased approach carefully preserving operational continuity throughout extended transition periods. The comprehensive strategy divided execution into four distinct phases spanning thirty-six months with explicit success criteria and governance checkpoints at each stage boundary.

Phase 1: Foundation and Pilot (Months 1-9)

Initial phase systematically established foundational platform capabilities while simultaneously validating architectural approach through carefully selected pilot domain implementation. Platform team developed

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

essential self-serve infrastructure providing storage services, processing frameworks, catalog integration mechanisms, and access management components enabling autonomous domain operation. Retail banking customer domain volunteered as pilot candidate given strong existing ownership culture and manageable technical complexity relative to more operationally complex domains.

Platform capabilities launched initially with deliberately constrained minimum viable functionality addressing core requirements without attempting comprehensive feature coverage. Object storage services provided durable repositories with automated lifecycle management and versioning capabilities. Containerized processing frameworks supported both batch and stream workloads through unified abstractions. Catalog API enabled automated metadata publishing and discovery through programmatic interfaces. Identity federation integrated seamlessly with enterprise authentication systems ensuring consistent access control. These carefully selected components collectively enabled genuinely autonomous domain operation without requiring custom infrastructure development or ongoing central team dependencies for routine operations.

Pilot domain team received intensive enablement support significantly exceeding typical training investments to maximize success probability. Comprehensive three-week training program systematically covered data product design principles, platform utilization techniques, and quality management practices. Embedded coaches with deep mesh experience provided continuous implementation guidance throughout extended pilot period. Weekly retrospectives systematically captured lessons learned informing subsequent domain onboarding processes and platform capability refinement.

Pilot domain successfully delivered three production data products serving customer demographic information, transaction histories, and account status datasets to diverse consumer populations. Initial development required five months including substantial learning curve and iterative platform capability refinement based on real usage. Data products systematically replaced existing centralized datasets, enabling direct performance comparison between architectural approaches. Consumer migration completed over two months with parallel operation period ensuring continuity and enabling rollback if quality issues emerged.

Phase 2: Early Adoption Expansion (Months 10-18)

Pilot success convincingly validated architectural approach, enabling controlled expansion to additional carefully selected domains. Three additional domains systematically onboarded following refined processes incorporating lessons from pilot experiences. Wealth management portfolio domain, corporate banking credit risk domain, and fraud detection operational domain joined the expanding mesh ecosystem. Platform capabilities expanded systematically based on pilot feedback, adding enhanced monitoring dashboards, quality automation frameworks, and governance enforcement mechanisms. Processing framework diversity increased to support varied workload requirements, with empirical evaluations demonstrating that optimized iterative frameworks achieved 10 to 100 times faster execution compared to traditional batch processing implementations for analytical workloads requiring multiple iterations [6].

Domain onboarding duration accelerated significantly as platform matured and training programs scaled beyond individual coaching. Average onboarding duration decreased from five months for pilot domain to three months for subsequent domains through improved platform maturity and refined enablement processes. The Enablement model evolved from resource-intensive embedded coaching toward scalable self-service documentation supplemented by center of excellence support for complex scenarios. Each onboarded domain delivered four to six data products systematically addressing their primary analytical requirements and replacing legacy centralized datasets.

Federated governance framework emerged organically during this expansion phase through collaborative policy development processes. Global policies covering data classification taxonomies, retention

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

requirements aligned with regulations, and access controls meeting risk thresholds received policy-as-code implementation enabling automation. Domain teams implemented and enforced these global policies within their specific products through platform automation mechanisms. Automated validation prevented production deployment of non-compliant products through integration with deployment pipelines. Governance automation dramatically reduced manual review burden by seventy percent compared to centralized model while simultaneously improving consistency through elimination of human judgment variability.

Phase 3: Scaled Adoption (Months 19-30)

Expansion accelerated substantially with demonstrated value from early adopters and increasingly refined processes reducing friction. Eight additional domains systematically onboarded during this critical scaling phase, encompassing operations across all business divisions and major geographic regions. Platform capabilities reached comprehensive operational maturity including advanced quality monitoring dashboards, lineage tracking across product dependencies, cost management tools enabling chargeback, and performance optimization features. The platform ecosystem expanded to provide domain teams with diverse framework selections from over 110 available implementations spanning storage technologies, processing paradigms, analytics engines, and machine learning capabilities [6]. This extensive tooling diversity enabled domain teams to select optimal implementations precisely aligned with specific workload characteristics without requiring custom infrastructure development or extensive specialized expertise.

Interoperability standards are formalized systematically through community-driven governance processes involving representatives from all active domains. Schema conventions, semantic standards, and contract evolution patterns received comprehensive documentation and supporting tooling. Cross-domain consumption patterns emerged naturally as domains increasingly leveraged each other's products for analytical workflows. Consumer experience improved substantially through consistent interfaces and reliable quality guarantees defined in service-level agreements.

Migration from legacy centralized lake proceeded systematically domain by domain. Domain data products replaced legacy datasets following rigorous validation of functional equivalence through parallel processing and result comparison. Critical analytical workloads migrated with comprehensive testing protocols and well-defined rollback procedures. Parallel operation periods varied by domain complexity, ranging from four to twelve weeks depending on consumer population size and workload criticality. Central platform team role transitioned fundamentally from direct data management toward platform capability evolution and domain enablement support.

Phase 4: Optimization and Continuous Improvement (Months 31-36)

The final phase achieved comprehensive domain coverage and operational excellence across the organization. Fifteen domains operated autonomous data product portfolios totaling eighty-seven production data products serving diverse analytical consumers. Central lake infrastructure decommissioned following complete consumer migration and validation period. The platform team refocused entirely on continuous capability evolution and domain support rather than operational data management.

Performance metrics systematically demonstrated substantial transformation impact across multiple dimensions. New data source onboarding accelerated dramatically from six weeks in a centralized model to one week in mesh architecture through domain autonomy. Quality incident resolution decreased from fourteen days to two days through clear domain accountability enabling rapid response. Access provision automated completely through policy-based mechanisms, eliminating manual backlogs that previously averaged three weeks. Metadata accuracy improved to ninety-five percent through domain ownership

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

leveraging contextual expertise. Consumer satisfaction scores increased thirty-eight percentage points based on systematic surveys tracking perceived data quality and usability.

Organizational capability transformation accompanied and enabled technical architectural evolution. Domain teams developed strong data product management competencies through sustained training and practical experience. Cross-functional collaboration patterns strengthened substantially through federated governance participation and cross-domain consumption. Innovation velocity increased measurably as domains experimented freely with advanced analytical capabilities without central bottlenecks constraining initiative approval. Data-as-product mindset embedded deeply across organizational culture through sustained leadership communication and recognition programs.

C. Key Lessons and Critical Success Factors

Transformation experience yielded valuable insights applicable to enterprise-scale implementations contemplating similar architectural evolution.

Executive Sponsorship Proved Absolutely Essential

Sustained executive commitment maintained throughout multi-year transformation preserved organizational focus during inevitable challenges and setbacks. Leadership communication consistently reinforced strategic importance and celebrated domain team contributions publicly. Resource allocation decisions prioritized transformation investments despite competing initiatives and budget pressures.

Incremental Approach Managed Risk Effectively

Phased domain onboarding systematically validated architectural patterns while carefully preserving operational stability throughout transition. Pilot domain lessons prevented broader-scale issues from affecting many domains simultaneously. Parallel operation periods enabled confident migration without exposure to disruption risk. Rollback capabilities provided essential safety mechanisms for recovering from unsuccessful attempts.

Platform Investment Enabled Domain Autonomy

Comprehensive self-serve capabilities systematically reduced domain team cognitive overhead enabling focus on business logic. Standardized tooling ensured consistency across domains without constraining contextual flexibility. Continuous platform evolution addressed emerging domain requirements through regular feedback cycles. Platform team focus on capability enablement rather than operational data management scaled effectively as domains increased.

Organizational Change Management Determined Success

Technical architecture transformation alone proved insufficient without corresponding cultural evolution and capability building. Sustained training investments built required domain capabilities across technical and organizational dimensions. Continuous change communication maintained organizational awareness of transformation progress and upcoming changes. Success recognition programs celebrated domain team achievements reinforcing desired behaviors. Leadership engagement demonstrated sustained organizational priority for transformation success.

Governance Automation Scaled Without Bottlenecks

Policy-as-code implementation enabled genuinely consistent enforcement without manual review burden constraints. Automated validation prevented compliance violations at domain boundaries through deployment pipeline integration. Federated governance model successfully balanced global standards with domain flexibility requirements. Continuous policy evolution addressed regulatory changes systematically through version-controlled specifications.

Community Building Accelerated Adoption

Cross-domain forums enabled valuable peer learning and pattern sharing reducing duplicated effort. Center of excellence functions effectively disseminated best practices organization-wide. Internal case

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

studies demonstrated concrete value and built confidence in architectural approach. Collaborative problem-solving through communities strengthened organizational cohesion across domain boundaries.

The comprehensive transformation positioned the organization for sustained competitive advantage through dramatically improved data management capabilities. Domain autonomy enabled substantially faster innovation cycles and market responsiveness. Quality improvements increased analytical reliability and business confidence in data-driven decisions. Governance automation significantly reduced compliance risk and audit burden. Organizational scaling model aligned data management capacity naturally with business growth trajectories enabling sustainable expansion.

X. Migration Strategies for Incremental Transformation

A. Parallel Architecture Operation and Risk Management

Enterprise transformations require carefully designed strategies that maintain unwavering operational continuity throughout extended transition periods. Existing analytical workloads depend fundamentally on centralized infrastructure investments representing substantial sunk costs and deeply embedded in business processes. Disruption to these business-critical workloads creates unacceptable business impact that severely constrains transformation velocity and organizational risk tolerance. Parallel architecture operation enables systematic incremental migration while carefully preserving critical capabilities throughout transition periods. Cattell and colleagues systematically examine architectural patterns enabling gradual evolution across data storage paradigms while maintaining operational stability [9]. Coexistence strategies allow centralized and distributed models to operate simultaneously during extended transition periods, enabling gradual consumer migration without forcing disruptive immediate cutover.

Migration sequencing systematically prioritizes domains based on carefully evaluated readiness factors and business impact assessments. High-readiness domains with strong existing ownership cultures and technical capabilities provide optimal initial transformation candidates minimizing implementation risk. Low-risk domains with limited consumer populations and non-critical workloads enable organizational learning without endangering mission-critical business processes. Success demonstrations from early domains build organizational confidence supporting broader adoption across more complex domains. Migration waves expand scope progressively as organizational capabilities mature and confidence increases through demonstrated success.

Data synchronization mechanisms carefully bridge centralized and mesh architectures during extended transition periods enabling parallel operation. Replication mechanisms maintain rigorous consistency between legacy repositories and emerging domain products through automated synchronization processes. Consumers migrate gradually from centralized to domain-oriented access patterns following their individual readiness timelines. Dual-write approaches enable parallel operation without data divergence through coordinated update mechanisms. Synchronization complexity requires careful technical management to avoid inconsistency issues that could undermine data quality and consumer confidence.

Risk management frameworks systematically govern transformation execution through defined checkpoints and success criteria. Rollback capabilities enable rapid recovery from unsuccessful migration attempts without extended business disruption. Comprehensive monitoring systems detect performance degradation or quality issues early in migration processes enabling proactive intervention. Explicitsuccess criteria define measurable thresholds for migration completion, ensuring objective evaluation rather than subjective assessment. Governance checkpoints validate readiness systematically before subsequent

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

migration phases commence, preventing premature advancement that could jeopardize transformation success.

B. Domain Capability Development and Organizational Change

Domain teams require systematic capability development to assume data product responsibilities effectively across technical and organizational dimensions. Rademacher and colleagues identify significant challenges in domain-driven design adoption requiring sustained organizational learning investments and cultural adaptation [10]. Technical skills encompass data modeling competencies, quality management practices, and platform utilization proficiency enabling autonomous operation. Organizational skills include stakeholder management capabilities, consumer engagement techniques, and service orientation mindset essential for product management. Cultural shifts move teams from passive data production toward active data product stewardship with accountability for consumer satisfaction.

Capability assessment systematically identifies gaps between current competencies and required capabilities for effective ownership. Technical assessments evaluate data engineering proficiency and platform utilization skills across domain teams. Organizational assessments examine collaboration capabilities and communication effectiveness with stakeholders and consumers. Cultural assessments gauge readiness for ownership accountability and willingness to embrace product management responsibilities. Gap analysis prioritizes development investments based on transformation requirements and organizational readiness levels.

Training programs build required capabilities systematically through structured curricula addressing multiple competency dimensions. Technical curricula comprehensively cover platform utilization techniques and data engineering practices including pipeline development and quality management. Organizational curricula address stakeholder engagement strategies and service management principles adapted from product management disciplines. Leadership development programs prepare managers for ownership accountability frameworks and resource allocation responsibilities. Peer learning programs enable valuable knowledge sharing across domain boundaries through communities of practice and regular knowledge exchange sessions.

Organizational structure evolution supports sustained mesh operation through deliberate role definition and resource allocation. Role definitions clearly articulate data product owner responsibilities distinguishing them from traditional data producer roles. Career paths recognize data product management as valued organizational contributions warranting appropriate compensation and advancement opportunities. Performance frameworks systematically incorporate data product health metrics including quality indicators, consumer satisfaction measures, and usage patterns. Resource allocation models fund domain data capabilities appropriately, ensuring teams possess adequate capacity for effective product management without unsustainable workload burdens.

Change management sustains organizational momentum through inevitable transformation challenges and setbacks. Communication programs maintain broad organizational awareness of transformation progress, upcoming changes, and achievement milestones. Success celebrations publicly recognize domain team achievements, reinforcing desired behaviors and building organizational confidence. Challenge acknowledgment validates implementation difficulties while simultaneously reinforcing sustained commitment to transformation objectives. Leadership engagement consistently demonstrates organizational priority for transformation success through resource allocation decisions, public communications, and personal involvement in key initiatives.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Strategy Element	Description	Implementation Consideration
Parallel Operation	Centralized and mesh architectures run simultaneously	Preservation of critical analytical workloads
Migration Sequencing	Domain prioritization based on readiness	High-readiness domains as initial candidates
Data Synchronization	Replication between legacy and domain products	Consistency maintenance during transition
Risk Management	Rollback capabilities and monitoring systems	Success criteria validation before phase progression
Capability Assessment	Gap identification between current and required competencies	Technical, organizational, and cultural evaluation
Change Management	Communication programs and success recognition	Sustained momentum through transformation challenges

Table 4. Incremental Transformation Approaches for Enterprise Data Architectures [9, 10].

Conclusion

Enterprise data management has reached a critical inflection point requiring fundamental architectural reconsideration beyond incremental optimization of existing approaches. Centralized data lake models served consolidation objectives effectively during initial adoption phases when organizational complexity remained manageable. However, growing organizational complexity systematically exposes inherent structural weaknesses within monolithic architectural designs that become increasingly problematic at scale. Ownership ambiguity creates persistent accountability gaps that progressively degrade data quality across extended pipeline chains involving multiple organizational handoffs. Governance bottlenecks restrict organizational responsiveness and systematically delay analytical initiatives requiring timely data access. Processing constraints generate ongoing conflicts among diverse domain requirements competing for shared infrastructure resources.

Data mesh architecture offers sustainable pathways forward through systematic redistribution of data responsibilities to domain teams possessing contextual expertise. The paradigm shift from technology-centric toward domain-centric design philosophy fundamentally distinguishes data mesh implementations from incremental improvements to existing centralized models that preserve underlying organizational structures. Product-oriented stewardship establishes clear quality and usability expectations at domain boundaries through explicit service-level commitments. Self-serve platforms strategically empower autonomous domain operation without creating problematic dependencies on centralized technical teams for routine operations. Federated governance ensures organizational consistency through computational policy enforcement rather than manual review processes that create bottlenecks.

The comprehensive architectural comparison systematically demonstrates fundamental differences across design philosophy, ownership models, scaling approaches, and governance structures that collectively define the paradigm shift. Visual architectural diagrams effectively clarify structural distinctions between monolithic centralized architectures and federated mesh implementations beyond textual description alone. The maturity framework provides systematic evaluation criteria enabling organizations to assess transformation readiness and track progress across multiple dimensions.

The detailed enterprise case study demonstrates practical transformation patterns across a comprehensive multi-year migration involving fifteen domains and eighty-seven production data products. Phased implementation successfully preserved operational continuity while systematically onboarding domains with progressively refined processes. Strategic platform investment enabled genuinely autonomous domain operation without creating ongoing central dependencies. Governance automation effectively scaled enforcement without creating bottlenecks through policy-as-code implementation. The transformation yielded substantial improvements including accelerated onboarding from six weeks to one week, reduced quality resolution from fourteen days to two days, automated access provisioning eliminating manual backlogs, and improved metadata accuracy to ninety-five percent.

Successful transformation requires carefully coordinated change across infrastructure, organizational structure, and cultural dimensions progressing in parallel. Domain capability development demands sustained investment in comprehensive skills training and enablement support spanning technical and organizational competencies. Migration sequencing must carefully balance transformation ambition against operational continuity requirements through deliberate phasing. Executive sponsorship proves essential for sustaining organizational momentum through inevitable implementation challenges and resource constraints. Incremental approaches effectively manage transformation risk through systematic pilot validation and carefully controlled phased expansion.

The architectural transition from centralized lakes toward mesh paradigms strategically positions enterprises for improved innovation velocity and enhanced data quality outcomes across increasingly complex operational environments. Organizations navigating this fundamental transformation benefit substantially from systematic assessment frameworks, comprehensive architectural visualization, and empirically validated implementation patterns. The synthesis provides essential architectural guidance for enterprise data management evolution, addressing contemporary scalability and governance challenges through distributed ownership models aligned with business capabilities while maintaining organizational consistency through federated computational governance.

References

[1] Pwint Phyu Khine and Zhao Shun Wang, "Data lake: a new ideology in big data era," ITM Web of Conferences, 2018. [Online]. Available: https://www.itm-conferences.org/articles/itmconf/pdf/2018/02/itmconf_wcsn2018_03025.pdf

[2] Abhijit Joshi, "Data Mesh: Decentralizing Data Management for Scalability and Agility," Journal of Scientific and Engineering Research, 2024. [Online]. Available: https://www.researchgate.net/profile/Abhijit-Joshi-9/publication/384056747

[3] Michael Stonebraker and Ugur Çetintemel, "One Size Fits All": An Idea Whose Time Has Come and Gone," Proceedings of the 21st International Conference on Data Engineering, 2005. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1410100

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

[5] Elena Piskun et al., "ECONOMIC DEVELOPMENT OF SEVASTOPOL AND ITS INFLUENCE ON THE POPULATION'S LIFE QUALITY," International Journal for Quality Research, 2019. [Online]. Available: https://ijqr.net/journal/v13-n2/15.pdf

[6] Shantenu Jha et al., "A Tale of Two Data-Intensive Paradigms: Applications, Abstractions, and Architectures," arXiv, 2014. [Online]. Available: https://arxiv.org/pdf/1403.1528

[7] Patricia Cupoli, "DAMA-DMBOK2 Framework," DAMA International, 2014. [Online]. Available: https://www.dvbi.ru/portals/0/DOCUMENTS_SHARE/BOOKS/DATA%20GOVERNANCE/DAMA-DMBOK2-Framework-v2.pdf

[8] Pooyan Jamshidi et al., "The Journey So Far and Challenges Ahead," IEEE Software, 2018. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8354433
[9] Rick Cattell et al., "Scalable SQL and NoSQL Data Stores," ACM, 2010. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1978915.1978919
[10] Florian Rademacher et al., "Challenges of Domain-Driven Microservice Design," IEEE Software, 2018. [Online]. Available: https://www.academia.edu/117996423/Challenges_of_Domain_Driven_Microservice_Design_A_Model_Driven_Perspective