

Data Mesh Architecture: A Decentralized Paradigm for Enterprise-Scale Data Infrastructure

Narendra Reddy Mudiya

HSquare IT Solutions Inc, USA

Abstract: As data volumes grow exponentially and traditional centralized architectures strain under the weight of modern enterprise demands, Data Mesh has emerged as a paradigm-shifting solution to scale data infrastructure and improve data ownership. This article explores the conceptual evolution, practical design principles, and implementation challenges associated with Data Mesh, a decentralized approach to data architecture that treats data as a product and distributes ownership to domain-specific teams. Unlike traditional monolithic data lakes or warehouses managed by centralized teams, Data Mesh advocates for federated governance, self-serve platform tooling, and domain-oriented decentralized data ownership. These principles enable organizations to unlock agility, scale, and quality by bringing data engineering responsibilities closer to subject matter experts. Drawing on production-proven use cases, the article dissects how Data Mesh resolves bottlenecks in ingestion, transformation, and discoverability, particularly in large-scale, multi-business-unit enterprises. The technological underpinnings essential for Data Mesh success include metadata-driven pipeline automation, policy-as-code enforcement, and scalable schema evolution. Frameworks such as Delta Lake, Apache Iceberg, and tools like AWS Lake Formation are examined in the context of enabling decentralized yet interoperable data products. Through real-world implementation insights, including experiences building modular, reusable frameworks such as Intelligent Data Foundation (IDF) on AWS, the article demonstrates how these serve as the foundation for Data Mesh success. Best practices for governance, lineage tracking, and cost-aware data ownership in distributed teams are outlined, contributing to the growing body of scholarly literature by clarifying the architectural, organizational, and cultural shifts required for sustainable Data Mesh adoption.

Keywords: Data Mesh, decentralized architecture, federated governance, domain-oriented ownership, self-serve platform.

INTRODUCTION

The Data Explosion and Architectural Strain

The unprecedented growth of data generation across industries has fundamentally challenged the scalability and effectiveness of traditional centralized data architectures. Organizations worldwide face mounting pressure to process, analyze, and derive insights from ever-expanding data volumes while maintaining agility and governance standards. This exponential data growth has exposed critical limitations in conventional approaches, prompting a fundamental reconsideration of how enterprises architect their data infrastructure.

Limitations of Traditional Centralized Approaches

Traditional data management paradigms, primarily centered around monolithic data lakes and data warehouses, have served organizations for decades. These centralized architectures typically funnel all organizational data through a single team or platform, creating a hub-and-spoke model where data producers push their data to a central repository, and data consumers pull from this consolidated source. However, recent frameworks (Li, J. *et al.*, 2022) demonstrate that these centralized approaches create significant bottlenecks in data ingestion, transformation, and

accessibility, particularly when dealing with complex, domain-specific data requirements. The centralized model inherently creates organizational dependencies, where domain teams must wait for central data teams to implement changes, leading to reduced agility and innovation speed.

Data Mesh as a Paradigm Shift

The emergence of Data Mesh represents a paradigm shift in how organizations conceptualize and implement data architecture. As explored in recent enterprise data strategies (Butte, V. K., & Butte, S. 2022), Data Mesh fundamentally reimagines data infrastructure through decentralization, treating data as a product and distributing ownership to domain-specific teams who best understand their data's context and value. This architectural philosophy moves away from the traditional centralized governance model toward a federated approach, where individual domains maintain ownership and accountability for their data products while adhering to global standards and interoperability requirements.

RESEARCH OBJECTIVES AND CONTRIBUTIONS

This article aims to provide a comprehensive examination of Data Mesh architecture, from its

conceptual foundations to practical implementation strategies. The primary objectives include elucidating the core principles that distinguish Data Mesh from traditional approaches, analyzing the technological infrastructure required for successful implementation, and synthesizing real-world experiences to provide actionable insights for practitioners. The contribution to the field lies in bridging the gap between theoretical frameworks and practical implementation, offering a holistic view of the organizational, technical, and cultural transformations necessary for successful Data Mesh adoption.

Article Structure

The article is structured to guide readers through both conceptual understanding and practical application. Following this introduction, Section 2 explores the conceptual foundations and evolution of Data Mesh, examining its core principles and philosophical underpinnings. Section 3 delves into the technological infrastructure and implementation components essential for Data Mesh success. Section 4 presents practical implementation insights through case studies and real-world applications. Section 5 addresses

governance, operational excellence, and best practices for sustainable adoption. The article concludes with Section 6, synthesizing key findings and outlining future directions for Data Mesh evolution.

CONCEPTUAL FOUNDATIONS AND EVOLUTION OF DATA MESH

Historical Context: From Monolithic to Distributed Data Architectures

The evolution of data architectures reflects the changing needs and technological capabilities of organizations over the decades. Early data management systems relied on monolithic designs where all data processing, storage, and governance resided within a single, tightly coupled system. As demonstrated in the transition from monolithic PLM landscapes to federated approaches (Hooshmand, Y. *et al.*, 2022), organizations have progressively moved toward more distributed architectures in response to increasing data complexity and organizational scale. This evolution began with the separation of operational and analytical systems, progressed through the era of data warehouses and lakes, and now culminates in the fully decentralized philosophy of Data Mesh.

Table 1: Evolution of Data Architecture Paradigms (Hooshmand, Y. *et al.*, 2022; Goedegebuure, A. *et al.*, 2024)

Era	Architecture Type	Key Characteristics	Primary Limitations
1980s-1990s	Monolithic Systems	Single system ownership, Tightly coupled components, Centralized processing	Limited scalability, Single point of failure, Difficult to modify
1990s-2000s	Data Warehouses	Separated analytical systems, ETL pipelines, Structured data focus	Batch processing delays, High infrastructure costs, and Schema rigidity
2000s-2010s	Data Lakes	Schema-on-read, Mixed data formats, Centralized storage	Data swamp risks, Governance challenges, Performance issues at scale
2010s-Present	Data Mesh	Distributed ownership, Domain-oriented, Product thinking	Cultural transformation required, Initial complexity, Skill gap challenges

Domain-Oriented Decentralized Data Ownership

The first core principle of Data Mesh fundamentally restructures how organizations think about data ownership. Rather than centralizing data management under a single team,

domain-oriented ownership distributes responsibility to the teams closest to the data's origin and business context. This principle recognizes that domain experts possess the deepest understanding of their data's semantics, quality requirements, and usage patterns. By aligning data

ownership with business domains, organizations can reduce communication overhead and accelerate decision-making while maintaining accountability for data quality and accessibility.

Data as a Product Mindset

The second principle transforms how organizations conceptualize and manage their data assets. Treating data as a product means applying product thinking to data assets, including clear ownership, defined consumers, quality guarantees, and lifecycle management. This shift requires domains to think beyond merely exposing raw data to actively managing discoverable, understandable, and trustworthy data products. Recent systematic reviews [4] highlight how this product mindset necessitates establishing service-level agreements, documentation standards, and feedback mechanisms similar to traditional software products.

Self-Serve Data Infrastructure as a Platform

The third principle addresses the technical complexity of implementing decentralized data ownership at scale. Self-serve data infrastructure provides domain teams with the tools and capabilities needed to create, maintain, and share their data products without requiring deep infrastructure expertise. This platform approach abstracts away the complexity of data storage, processing, and governance while providing standardized interfaces and tooling. The platform must balance autonomy with standardization, enabling domains to innovate while ensuring interoperability across the organization.

Federated Computational Governance

The fourth principle reconciles the tension between decentralization and the need for organizational coherence. Federated computational governance establishes global standards and policies while allowing domains to maintain local autonomy. This approach embeds governance rules into the platform through computational mechanisms rather than relying solely on manual processes and documentation. The systematic literature [4] emphasizes how this principle enables organizations to maintain security, privacy, and compliance standards across distributed domains while avoiding the bottlenecks of centralized approval processes.

Comparison with Traditional Centralized Approaches

Data Mesh represents a fundamental departure from traditional centralized data architectures in several key dimensions. While centralized approaches concentrate technical expertise and infrastructure in a single team, Data Mesh distributes these capabilities across domains. Traditional architectures often struggle with scalability as the central team becomes a bottleneck for all data-related changes, whereas Data Mesh scales naturally with organizational growth. The shift from monolithic to federated architectures (Hooshmand, Y. *et al.*, 2022) demonstrates how centralized models create dependencies and delays, while Data Mesh enables parallel development and faster time-to-value for data initiatives.

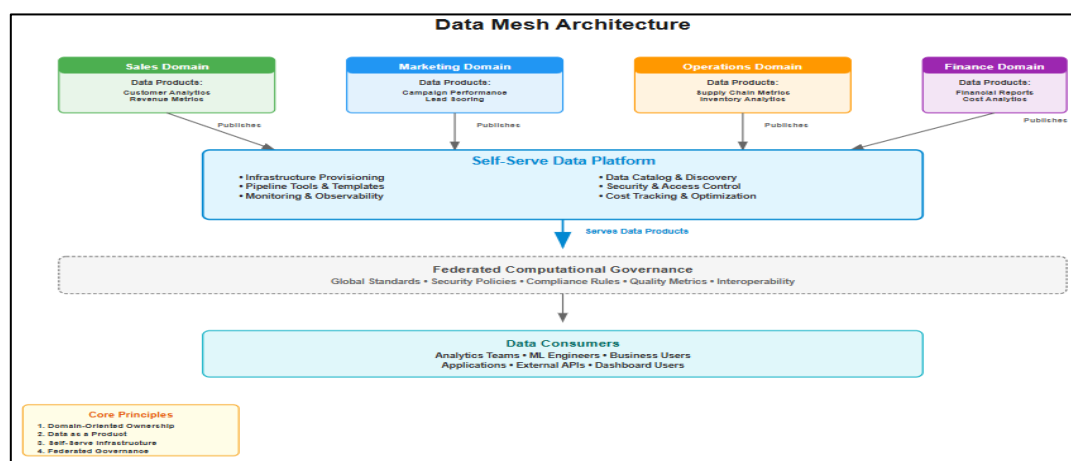


Fig. 1: Data Mesh Architecture Showing Domain Autonomy and Federated Governance (Butte, V. K., & Butte, S. 2022; Wider, A. 2023; BDGMM-IC 2020)

Table 2: Data Mesh Core Principles Comparison (Hooshmand, Y. *et al.*, 2022; Goedegebuure, A. *et al.*, 2024)

Principle	Traditional Approach	Data Mesh Approach	Key Benefits
Ownership	Centralized data team	Domain teams	Faster decision-making, Better context understanding
Data Treatment	Technical asset	Product with consumers	Quality focus, Clear accountability
Infrastructure	Centralized platform	Self-serve capabilities	Reduced bottlenecks, Domain autonomy
Governance	Top-down policies	Federated computation	Automated enforcement, Local flexibility

Theoretical Underpinnings and Architectural Philosophy

The theoretical foundations of Data Mesh draw from multiple disciplines, including distributed systems, organizational theory, and software architecture. Architectural philosophy embraces principles of loose coupling, high cohesion, and bounded contexts based on domain-driven design. This approach recognizes that complex systems benefit from decomposition into manageable, autonomous components that can evolve independently while maintaining clear interfaces. The philosophy extends beyond technical architecture to encompass organizational structure, team topologies, and cultural transformation, acknowledging that sustainable change requires alignment across all these dimensions.

TECHNOLOGICAL INFRASTRUCTURE AND IMPLEMENTATION COMPONENTS

Metadata-Driven Pipeline Automation

The foundation of Data Mesh implementation relies heavily on metadata-driven automation to manage the complexity of distributed data products. Reference architectures (van der Werf. *Et al.*, 2024) emphasize how metadata serves as the connective tissue between autonomous domains, enabling automated discovery, lineage tracking, and quality validation. This approach replaces manual pipeline configuration with declarative specifications that capture data product characteristics, transformation logic, and quality requirements. Metadata-driven automation enables domains to focus on business logic while the platform handles technical implementation details, ensuring consistency across diverse data products without sacrificing domain autonomy.

Policy-as-Code Enforcement Mechanisms

Policy-as-code represents a critical technological foundation for implementing federated governance in Data Mesh architectures. This approach translates organizational policies into executable code that automatically enforces compliance, security, and quality standards across all data products. Implementation experiences (Majchrzak, J. *et al.*, 2023) demonstrate how policy-as-code mechanisms enable organizations to maintain governance at scale without creating manual bottlenecks. These mechanisms include automated access control validation, data quality checks, privacy compliance verification, and cost allocation rules, all embedded within the data platform infrastructure.

Scalable Schema Evolution Strategies

Managing schema evolution across distributed domains presents unique challenges in Data Mesh implementations. Scalable schema evolution strategies must balance backward compatibility with the need for continuous innovation and adaptation. Modern architectures (van der Werf. *Et al.*, 2024) incorporate versioning mechanisms, compatibility checking, and automated migration tools to support schema changes without disrupting downstream consumers. These strategies enable domains to evolve their data products independently while maintaining interoperability through clear deprecation policies and migration paths.

Delta Lake Architecture and Capabilities

Delta Lake emerges as a key technology enabling Data Mesh implementations, providing ACID transactions, time travel capabilities, and unified batch and streaming processing. The architecture supports the data product lifecycle through features like schema enforcement, data versioning,

and audit logging. Practical implementations (Majchrzak, J. *et al.*, 2023) highlight how Delta Lake's open format and cross-platform compatibility enable domains to build reliable data products while maintaining flexibility in tool

selection. The technology's support for concurrent reads and writes proves essential for supporting multiple consumers accessing shared data products.

Table 3: Technology Stack for Data Mesh Implementation (van der Werf. *et al.*, 2024; Majchrzak, J. *et al.*, 2023)

Component	Technology Options	Primary Function	Key Features
Table Format	Delta Lake, Apache Iceberg, Apache Hudi	Data storage and versioning	ACID transactions, Schema evolution, Time travel
Orchestration	Apache Airflow, Prefect, Dagster	Pipeline automation	Dependency management, Monitoring, and Scheduling
Catalog	AWS Glue, Apache Atlas, DataHub	Metadata management	Discovery, Lineage, Classification
Compute	Spark, Presto, Snowflake	Data processing	Distributed processing, SQL interface, Performance
Governance	AWS Lake Formation, Privacera, Immuta	Access control	Fine-grained permissions, Policy enforcement, Audit trails

Apache Iceberg for Table Format Management

Apache Iceberg addresses critical requirements for managing large-scale analytical datasets in distributed environments. The table format provides schema evolution, partition evolution, and hidden partitioning capabilities that simplify data product management for domain teams. Reference architectures (van der Werf, *et al.*, 2024) incorporate Iceberg to enable efficient querying across massive datasets while maintaining performance through intelligent metadata management and file pruning. The format's support for multiple concurrent writers and snapshot isolation ensures data consistency in decentralized environments.

AWS Lake Formation for Data Mesh Implementation

AWS Lake Formation exemplifies how cloud platforms adapt to support Data Mesh principles through managed services. The platform provides centralized permissions management with fine-grained access control, enabling federated governance while maintaining security boundaries. Implementation case studies (Majchrzak, J. *et al.*, 2023) demonstrate how Lake Formation's data catalog, governed tables, and cross-account sharing capabilities facilitate the creation of self-serve data infrastructure. The service abstracts complex infrastructure management while

providing the flexibility required for domain autonomy.

Integration Patterns and Interoperability Considerations

Successful Data Mesh implementations require well-defined integration patterns that enable seamless interoperability between autonomous domains. Common patterns include event-driven synchronization, API-based data access, and shared data contracts that define interface specifications. The reference architecture (van der Werf, *et al.*, 2024) emphasizes the importance of standardized protocols for data exchange, discovery mechanisms for finding relevant data products, and compatibility layers that handle format conversions. These patterns must balance standardization with flexibility, allowing domains to choose appropriate technologies while maintaining ecosystem coherence.

Technical Requirements for Self-Serve Platforms

Building effective self-serve data platforms demands careful consideration of technical requirements that empower domain teams while maintaining operational excellence. Essential capabilities include automated provisioning of compute and storage resources, integrated development environments for data transformation, and comprehensive monitoring and

observability tools. Practical experiences (Majchrzak, J. *et al.*, 2023) reveal that successful platforms provide abstracted interfaces for common tasks, pre-built templates for standard patterns, and extensive documentation and training resources. The platform must hide infrastructure complexity while exposing sufficient control for advanced use cases, creating a graduated experience that supports users across different skill levels.

PRACTICAL IMPLEMENTATION: CASE STUDIES AND REAL-WORLD APPLICATIONS

Production-Proven Use Cases in Multi-Business-Unit Enterprises

Large enterprises with multiple business units represent ideal candidates for Data Mesh adoption due to their inherent organizational complexity and diverse data requirements. Engineering implementations in cloud environments (Deswandikar, A. 2024) demonstrate how multinational corporations leverage Data Mesh to overcome the limitations of centralized data teams serving disparate business units with varying priorities and timelines. These organizations typically operate across different geographic regions, regulatory environments, and market segments, creating natural boundaries that align with the Data Mesh's domain-oriented approach. The implementation patterns show how autonomous business units can maintain their data products while contributing to a larger organizational data ecosystem.

Architecture Design and Modular Framework

The Intelligent Data Foundation (IDF) on AWS exemplifies a modular approach to Data Mesh implementation that balances standardization with flexibility. The architecture employs a layered design where foundational services provide common capabilities while allowing domains to customize their specific implementations. Cloud adoption frameworks (Deswandikar, A. 2024) guide the creation of reusable components for data ingestion, processing, and serving, packaged as self-contained modules that domains can compose according to their needs. This modular approach accelerates domain onboarding while maintaining architectural consistency across the organization.

Implementation Challenges and Solutions

Real-world implementations reveal common challenges that organizations face when transitioning to Data Mesh architectures. Technical challenges include managing distributed transactions across domains, ensuring consistent data quality standards, and handling cross-domain dependencies. Organizational challenges prove equally significant, as detailed in implementation guides (Bunnell, J. 2022), including resistance to change from centralized teams, unclear ownership boundaries between domains, and the need for new skill sets within domain teams. Solutions involve phased migration strategies, comprehensive training programs, and the establishment of center-of-excellence teams that support domains during their transition.

Table 4: Implementation Challenges and Mitigation Strategies (Deswandikar, A. 2024; Bunnell, J. 2022)

Challenge Category	Specific Issues	Mitigation Strategies	Success Indicators
Technical	Cross-domain transactions, Data quality consistency, Performance optimization	Standardized interfaces, Automated quality checks, Caching strategies	Reduced latency, Quality metrics improvement, System stability
Organizational	Resistance to change, Unclear boundaries, Skill gaps	Phased migration, RACI matrices, Training programs	Adoption rates, Team satisfaction, Delivery velocity
Cultural	Loss of control fears, Accountability concerns, Collaboration barriers	Executive sponsorship, Success showcases, Incentive alignment	Cross-domain projects, Knowledge sharing, Innovation metrics

Lessons Learned and Best Practices

Practical experience from IDF implementations highlights critical success factors for Data Mesh

adoption. Starting with high-value, well-bounded domains increases the likelihood of early success and builds organizational confidence.

Implementation guides (Bunnell, J. 2022) emphasize the importance of investing in platform capabilities before scaling domain adoption, ensuring adequate tooling and automation are available. Clear data product specifications, including schema definitions, quality metrics, and SLAs, prove essential for maintaining interoperability. Regular cross-domain collaboration sessions help identify common patterns and prevent duplicate efforts across the organization.

Resolving Bottlenecks in Data Ingestion, Transformation, and Discoverability

Data Mesh implementations demonstrate significant improvements in addressing traditional data pipeline bottlenecks. Distributed ingestion allows domains to onboard new data sources without waiting for central team availability, while standardized ingestion patterns ensure consistency. Cloud-based implementations (Deswandikar, A. 2024) show how transformation logic moves closer to domain experts who understand business context, reducing iteration cycles and improving accuracy. Enhanced discoverability through comprehensive data catalogs and metadata management enables consumers to find relevant data products without extensive coordination, dramatically reducing time-to-insight.

Organizational Transformation and Cultural Shifts

The transition to Data Mesh requires fundamental changes in organizational culture and operating models. Traditional data teams must evolve from gatekeepers to enablers, focusing on platform capabilities rather than individual pipeline development. Domain teams assume new responsibilities for data quality and accessibility, requiring investment in data literacy and engineering skills. Leadership guides (Bunnell, J. 2022) highlight how successful transformations involve executive sponsorship, clear communication of benefits, and recognition systems that reward data product quality and reuse. The cultural shift emphasizes collaboration over control, with success measured by the value delivered to data consumers rather than the volume of data processed.

GOVERNANCE, OPERATIONAL EXCELLENCE, AND BEST PRACTICES

Federated Governance Models and Implementation

Federated governance in Data Mesh represents a delicate balance between autonomy and standardization, requiring new models that distribute decision-making while maintaining organizational coherence. Decentralized governance concepts (Wider, A. 2023) introduce computational mechanisms that embed policies directly into the platform infrastructure, enabling automated enforcement without manual intervention. This approach establishes global standards for security, privacy, and interoperability while allowing domains to define local policies specific to their business context. The implementation typically involves governance guilds comprising representatives from different domains who collaborate to establish and evolve organizational standards.

Data Lineage Tracking in Distributed Environments

Tracking data lineage across distributed domains presents unique challenges that require sophisticated technical and organizational solutions. Modern implementations leverage automated metadata capture at every transformation point, creating comprehensive lineage graphs that span domain boundaries. Standards and best practices (BDGMM-IC, 2020) emphasize the importance of standardized lineage formats that enable cross-platform compatibility and unified visualization. The distributed nature of Data Mesh necessitates federated lineage systems where each domain maintains local lineage information while contributing to a global lineage graph through well-defined interfaces.

Cost-Aware Data Ownership and Accountability Models

Data Mesh implementations must address the economic implications of distributed ownership through transparent cost allocation and accountability mechanisms. Cost-aware models attribute infrastructure and operational expenses directly to the domains that generate them,

creating natural incentives for efficient resource utilization. Governance frameworks (Wider, A. 2023) incorporate automated cost tracking and allocation systems that provide real-time visibility into data product expenses. This approach transforms cost from an abstract organizational concern to a concrete consideration for domain teams, fostering responsible resource management and encouraging optimization efforts.

Quality Assurance and Monitoring Strategies

Quality assurance in decentralized environments requires a shift from centralized validation to distributed responsibility with standardized metrics and monitoring. Each domain implements quality checks specific to its data products while adhering to organization-wide quality standards. Industry standards (BDGMM-IC 2020) provide frameworks for defining quality dimensions including accuracy, completeness, timeliness, and consistency. Monitoring strategies employ automated quality scoring systems that continuously assess data products against defined criteria, with alerts and dashboards providing visibility to both producers and consumers.

Security and Compliance in Decentralized Architectures

Security and compliance considerations become more complex in distributed architectures, requiring comprehensive approaches that maintain protection without hindering agility. Decentralized platforms (Wider, A. 2023) implement security through multiple layers, including encryption at rest and in transit, fine-grained access controls, and automated compliance validation. Regulatory requirements often necessitate data localization and sovereignty controls, which Data Mesh architectures address through policy-driven data placement and access restrictions. The approach shifts from perimeter-based security to zero-trust models, where every interaction requires authentication and authorization.

Team Structure and Skill Requirements

Successful Data Mesh adoption requires restructuring teams and developing new skill sets that blend domain expertise with data engineering capabilities. Domain teams expand to include embedded data engineers, product managers for

data products, and data quality analysts. Standards roadmaps (BDGMM-IC 2020) identify essential competencies including cloud platform expertise, distributed systems understanding, and data modeling skills. Organizations often establish platform teams that provide tooling and support while domain teams focus on business-specific data products. This structure requires investment in training programs and potentially recruiting new talent with appropriate skill combinations.

Change Management and Adoption Strategies

The transformation to Data Mesh demands comprehensive change management strategies that address technical, organizational, and cultural dimensions. Successful adoptions begin with pilot programs in receptive domains, demonstrating value before broader rollout. Governance frameworks (Wider, A. 2023) recommend establishing clear communication channels, success metrics, and feedback mechanisms throughout the transition. Change management must address resistance from existing centralized teams by redefining their roles as platform enablers rather than data pipeline builders. Regular showcases of successful data products build momentum and encourage adoption across the organization, while executive sponsorship ensures sustained commitment through inevitable challenges.

CONCLUSION

Data Mesh represents a fundamental shift in how organizations architect, manage, and derive value from their data assets in an era of unprecedented scale and complexity. The transition from monolithic, centralized data platforms to distributed, domain-oriented architectures addresses critical bottlenecks that have long plagued enterprise data initiatives, enabling organizations to scale their data capabilities in alignment with business growth. Through the core principles of domain ownership, data as a product, self-serve infrastructure, and federated governance, Data Mesh provides a framework that balances autonomy with standardization, empowering domain experts while maintaining organizational coherence. The technological foundations, including metadata-driven

automation, policy-as-code enforcement, and modern data formats, create the necessary infrastructure for sustainable implementation. Real-world experiences demonstrate that successful adoption requires not only technical transformation but also significant organizational and cultural change, with new team structures, skill development, and governance models. As organizations continue to grapple with expanding data volumes and increasing demands for agility, Data Mesh offers a proven path forward that aligns technical architecture with organizational structure, creating sustainable foundations for data-driven innovation. The journey toward Data Mesh implementation presents challenges, yet the benefits of improved scalability, reduced time-to-value, and enhanced data quality position it as an essential evolution for enterprises seeking to maximize the value of their data assets in an increasingly complex digital landscape.

REFERENCES

1. Li, J., Cai, S., Wang, L., Li, M., Li, J., & Tu, H. "A novel design for data processing framework of park-level power system with data mesh concept." *2022 IEEE International Conference on Energy Internet (ICEI)*. IEEE, (2022).
2. Butte, V. K., & Butte, S. "Enterprise data strategy: a decentralized data mesh approach." *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*. IEEE, (2022).
3. Hooshmand, Y., Resch, J., Wischnewski, P., & Patil, P. "From a monolithic PLM landscape to a federated domain and data mesh." *Proceedings of the Design Society 2* (2022): 713-722.
4. Goedegebuure, A., Kumara, I., Driessen, S., Van Den Heuvel, W. J., Monsieur, G., Tamburri, D. A., & Nucci, D. D. "Data mesh: a systematic gray literature review." *ACM Computing Surveys* 57.1 (2024): 1-36.
5. van der Werf, D., Moreira, J., & Piest, J. P. S. "Towards a Data Mesh Reference Architecture." *International Conference on Enterprise Design, Operations, and Computing*. Cham: Springer Nature Switzerland, (2024).
6. Majchrzak, J., Balnojan, S., & Siwiak, M. "Data mesh in action. Simon and Schuster", (2023)
7. Deswandikar, A. "Engineering Data Mesh in Azure Cloud: Implement Data Mesh Using Microsoft Azure's Cloud Adoption Framework". *Packt Publishing Ltd*, (2024)
8. Bunnell, J. "What Is Data Mesh: A Simple Guide to Implementing the Latest Industry," IEEE Computer Society, (2022). <https://www.computer.org/publications/tech-news/trends/guide-to-implement-data-mesh>
9. Wider, A., Verma, S., & Akhtar, A. "Decentralized data governance as part of a data mesh platform: concepts and approaches." *2023 IEEE International Conference on Web Services (ICWS)*. IEEE, (2023).
10. IEEE Big Data Governance and Metadata Management Industry Connections Activity (BDGMM-IC), "Big Data Governance and Metadata Management: Standards Roadmap," *IEEE Standards Association White Paper*, (2020). <https://standards.ieee.org/wp-content/uploads/import/governance/iccomm/bdgmm-standards-roadmap-2020.pdf>

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:

Mudiyala, N. R. "Data Mesh Architecture: A Decentralized Paradigm for Enterprise-Scale Data Infrastructure." *Sarcouncil Journal of Multidisciplinary* 5.7 (2025): pp 578-586.