

Composable Data Architectures: Moving Beyond the Monolithic Platform

Rahul Joshi

IIT Kharagpur, India

Abstract: This article examines the paradigm shift from monolithic enterprise data platforms to composable data architectures, offering a comprehensive analysis of this emerging approach to data infrastructure design. The article explores how composable architectures—characterized by decoupled functional layers, standardized interfaces, and metadata-driven orchestration—enable organizations to achieve both technological flexibility and robust governance simultaneously. The article demonstrates how composability addresses fundamental limitations of traditional architectures while delivering measurable improvements in platform agility, technical debt management, and governance effectiveness. The article identifies key implementation strategies for transitioning from monolithic systems, including the strangler pattern, domain-based decomposition, and metadata-driven orchestration. The article also analyzes common challenges organizations face during implementation, from organizational resistance to skills gaps and integration complexity, while providing evidence-based approaches to overcoming these obstacles. As emerging technologies continue to enhance component interoperability and governance capabilities, composable architectures are positioned to become the dominant paradigm for enterprise data management, particularly in regulated industries requiring both innovation agility and comprehensive governance. This article contributes to both theoretical understanding and practical implementation guidance for organizations seeking to modernize their data infrastructure in an increasingly complex and rapidly evolving technological landscape.

Keywords: Composable Data Architecture, Metadata Orchestration, Technical Debt Reduction, Domain-Aligned Data Products, Governance Effectiveness

INTRODUCTION

In today's data-driven business landscape, enterprise data architectures have evolved from supporting basic analytics to becoming critical strategic assets that enable digital transformation, real-time decision making, and competitive differentiation (Simoni G. and D Beyer M. 2022). However, many organizations find themselves constrained by monolithic data platforms that have grown increasingly complex, inflexible, and difficult to govern over time. These legacy architectures—often characterized by tightly coupled components, proprietary integrations, and rigid workflows—struggle to adapt to rapidly changing business requirements and technology innovations.

The emergence of composable data architectures represents a fundamental paradigm shift in how organizations design, implement, and evolve their data infrastructure. Unlike traditional monolithic approaches, composable architectures embrace modularity, interoperability, and loose coupling between functional components. This design philosophy allows enterprises to select best-of-breed technologies for specific data management functions while maintaining coherent governance and metadata management across the entire ecosystem.

The significance of this architectural transition extends beyond technical considerations. As organizations face increasing pressure to derive

value from expanding data volumes while navigating complex regulatory landscapes, the ability to adapt and evolve data infrastructure without complete rewrites has become a strategic imperative. Industries with stringent regulatory requirements, such as financial services and healthcare, stand to benefit particularly from the governance capabilities inherent in well-designed composable architectures.

This paper examines the conceptual foundations, implementation patterns, and practical benefits of composable data architectures. The article explores how organizations can transition from monolithic systems to modular components, the critical role of metadata orchestration and schema registries, and how composability supports multi-cloud and domain-aligned data platforms. Through analysis of real-world implementations, the article demonstrates how this architectural pattern accelerates platform maturity while reducing technical debt at scale.

The article addresses several key questions: What are the defining characteristics and principles of composable data architectures? How can organizations successfully transition from monolithic to composable approaches? What governance mechanisms enable coherence across decoupled components? And finally, what measurable benefits do organizations realize through architectural composability?

THEORETICAL BACKGROUND

Defining Composable Data Architectures

Composable data architectures represent a design approach where data management capabilities are implemented as modular, independent components that can be assembled, replaced, or reconfigured with minimal disruption (Dehghani,Z. 2021). These architectures are characterized by clear

functional boundaries, well-defined interfaces, and the ability to exchange individual components without redesigning the entire system. This approach diverges from traditional monolithic platforms by prioritizing flexibility, interoperability, and evolutionary design over tightly integrated, vendor-specific solutions.

Table 1: Comparison of Monolithic vs. Composable Data Architectures (Dehghani,Z. 2021)

Characteristic	Monolithic Architecture	Composable Architecture
Component Coupling	Tightly integrated components with dependencies	Loosely coupled components with clear interfaces
Technology Adoption	System-wide updates requiring comprehensive testing	Component-level updates with minimal system impact
Governance Model	Centralized controls embedded within the platform	Distributed enforcement with centralized policy definition
Scalability	Vertical scaling with system-wide resource allocation	Independent scaling of individual components
Modernization Approach	"Big bang" replacements with high disruption risk	Incremental evolution with targeted component upgrades
Vendor Strategy	Single-vendor platforms with proprietary integrations	Best-of-breed components with standardized interfaces
Metadata Management	Often siloed within platform boundaries	Centralized metadata serves as an integration backbone

Historical Evolution from Monolithic Systems

Data architectures have evolved through several distinct phases over the past decades. Early enterprise data warehouses typically followed monolithic designs with tightly coupled ETL processes, storage, and reporting capabilities. The mid-2000s saw the emergence of data lakes, which provided greater storage flexibility but often maintained monolithic processing frameworks. The limitations of these approaches—including lengthy implementation cycles, vendor lock-in, and resistance to technological change—created increasing friction as organizations sought greater agility and innovation capacity. Composable architectures emerged as organizations began adopting cloud-native and containerized technologies that facilitated more modular approaches to data management.

CONCEPTUAL FRAMEWORK AND DESIGN PRINCIPLES

Composable data architectures adhere to several core design principles:

1. Separation of concerns: Each component addresses a specific data management function with clear boundaries.
2. Interface standardization: Components interact through well-defined contracts, APIs, and metadata standards.

3. Independent lifecycle management: Components can be deployed, updated, or replaced independently.
4. Metadata centrality: Shared metadata provides coherence across decoupled components.
5. Governance by design: Controls and policies can be implemented consistently across heterogeneous components.

These principles enable organizations to achieve both flexibility and control, addressing the fundamental tension in enterprise data management (Michael Stonebraker, Ugur Cetintemel. 2005).

Related Paradigms: Microservices, API-First, and Event-Driven Architectures

Composable data architectures draw inspiration from several related software engineering paradigms. Microservices architecture provides patterns for building systems as collections of loosely coupled, independently deployable services. API-first approaches prioritize stable interfaces between components, enabling teams to work independently while maintaining interoperability. Event-driven architectures offer patterns for asynchronous communication between components, reducing tight coupling and enabling real-time data flows. Composable data architectures synthesize these approaches within

the specific context of data management, addressing the unique challenges of schema evolution, metadata management, and data governance.

ARCHITECTURAL COMPONENTS AND PATTERNS

Decoupled Functional Layers

Data Ingestion

In composable architectures, data ingestion components handle the acquisition of data from various sources, including operational systems, third-party APIs, and streaming platforms. These components implement source-specific connectors, handle authentication and authorization, and manage extraction patterns (batch, micro-batch, or streaming). By decoupling ingestion from downstream processing, organizations can adapt to changing source systems without disrupting analytical workloads.

Storage

Storage components provide persistent repositories for data at various stages of processing. Modern composable architectures typically employ a polyglot storage approach, selecting appropriate technologies based on workload characteristics. This may include object storage for raw data, columnar formats for analytical processing, and specialized datastores for specific use cases. Separation of storage from processing enables independent scaling and technology selection.

Transformation

Transformation components implement data processing logic to convert raw data into analysis-ready formats. These components may include SQL-based transformations, procedural processing, machine learning pipelines, or domain-specific processing frameworks. Composable architectures emphasize declarative transformations that can be versioned, tested, and executed on various runtime engines.

Metadata Management

Metadata management provides the "connective tissue" that binds decoupled components into a coherent system. This includes technical metadata (schemas, lineage), operational metadata (performance metrics, data quality), and business metadata (definitions, ownership). Active metadata management—where metadata drives automated processes rather than serving merely as documentation—is particularly crucial in composable architectures (Strengtholt,P. 2023).

Data Serving

Serving components provide controlled access to processed data for consumption by analytical tools, applications, and data products. These may include API gateways, query engines, feature stores for machine learning, and specialized interfaces for specific consumption patterns. By decoupling serving from upstream processes, organizations can tailor access mechanisms to consumer requirements without modifying core processing pipelines.

INTEGRATION MECHANISMS

Contracts and APIs

Contracts define the expected behaviors, inputs, and outputs of each component, enabling integration without tight coupling. Modern data architectures leverage both synchronous APIs (REST, GraphQL) and asynchronous interfaces (messaging, event streams) to facilitate interaction between components while preserving independence.

Schema Registries

Schema registries provide centralized repositories for data structure definitions, supporting compatibility checking and evolution. These registries enable components to validate compliance with expected formats and gracefully handle schema changes, reducing the risk of integration failures in loosely coupled systems.

Metadata Orchestration

Metadata orchestration coordinates the flow of information and triggers actions across decoupled components. This includes workflow scheduling, data quality validation, and policy enforcement. By driving orchestration through metadata rather than hard-coded workflows, composable architectures maintain flexibility while ensuring coherent end-to-end processes.

Governance Implementation in Composable Systems

Governance in composable architectures shifts from monolithic control to distributed enforcement with centralized policy definition. This approach leverages metadata to implement consistent controls across heterogeneous components, including:

- Access controls are implemented at component interfaces
- Data quality checks are executed during data movement
- Lineage tracking across component boundaries
- Policy enforcement at integration points

- This pattern enables organizations to maintain robust governance while preserving the flexibility and innovation benefits of composable architectures.

IMPLEMENTATION STRATEGIES

Transitioning from Monolithic to Composable Architectures

Organizations typically adopt an incremental approach when transitioning from monolithic to composable architectures. This process begins with identifying architectural seams—points where existing systems can be decoupled with minimal disruption. Common strategies include:

Strangler pattern: Gradually replacing monolithic functionality by intercepting calls to the legacy system and routing them to new modular components.

Domain-based decomposition: Breaking down monolithic systems based on business domains or data products.

Data virtualization layer: Implementing abstraction layers that provide unified access while progressively migrating underlying systems.

Successful transitions maintain operational continuity while incrementally introducing composable components, often beginning with new data products that demonstrate the approach's value (Fowler,M. 2025).

Technology Selection Frameworks

Effective technology selection in composable architectures requires systematic evaluation frameworks that consider both component-specific capabilities and ecosystem compatibility. Leading organizations implement selection criteria that assess:

1. **Interoperability:** Standard interfaces, API support, metadata exchange capabilities
2. **Deployment flexibility:** Container support, infrastructure-as-code compatibility
3. **Extensibility:** Plugin architectures, custom adapters, open-source foundations
4. **Governance capabilities:** Security controls, lineage tracking, policy enforcement

These frameworks help organizations avoid creating new silos while preserving the flexibility to adopt innovative technologies as needs evolve.

4.3 Metadata-Driven Orchestration

Metadata-driven orchestration represents a fundamental shift from hard-coded data pipelines to dynamic workflows driven by metadata. This approach centers on:

1. **Declarative specifications:** Defining desired outcomes rather than procedural steps
2. **Metadata repositories:** Centralizing definitions, schemas, and policies
3. **Orchestration engines:** Interpreting metadata to coordinate component execution
4. **Observability frameworks:** Capturing runtime metrics and lineage for governance

By decoupling orchestration logic from specific technologies, organizations can maintain consistent processes while evolving individual components independently.

4.4 Multi-Cloud and Hybrid Deployments

Composable architectures are particularly well-suited to multi-cloud and hybrid deployments, enabling organizations to leverage each provider's strengths while avoiding lock-in. Key implementation patterns include:

1. **Cloud-agnostic control planes:** Managing workflows across heterogeneous infrastructure
2. **Data fabric patterns:** Providing consistent data access across distributed environments
3. **Containerized deployment:** Ensuring component portability across environments
4. **Cross-cloud metadata integration:** Maintaining coherent governance across boundaries

These approaches allow organizations to optimize cost, performance, and compliance requirements while maintaining architectural coherence.

Domain-Aligned Data Products

Domain-aligned data products extend composability principles to organizational structures, aligning data ownership with business domains. Implementation strategies include:

1. **Domain identification:** Mapping business capabilities to data domains
2. **Product-oriented teams:** Aligning responsibility for both production and consumption
3. **Self-service capabilities:** Enabling domain teams to manage their data products
4. **Federated governance:** Distributing responsibility while maintaining standards

This approach enhances business alignment while reducing cross-team dependencies that can impede agility.

CASE STUDIES

5.1 Financial Services: Regulatory Compliance and Agility

A global financial institution implemented a composable data architecture to address the competing demands of regulatory compliance and

business agility. The organization decoupled its risk reporting platform into distinct components for data acquisition, quality control, calculation engines, and reporting interfaces. This approach enabled:

1. **Independent regulatory updates:** Modifying compliance components without disrupting business operations
2. **Accelerated innovation:** Introducing new analytical capabilities alongside established processes
3. **Enhanced auditability:** Implementing consistent lineage tracking across components

The institution reduction in regulatory change implementation time while maintaining compliance across multiple jurisdictions (Deloitte. 2023).

Retail: Gradual Modernization Without Disruption

A multinational retailer adopted a composable approach to modernize its legacy data warehouse without disrupting critical business operations. The company implemented:

1. **Data virtualization layer:** Providing consistent access while migrating underlying systems
2. **Domain-specific data products:** Incrementally replacing monolithic reporting with targeted capabilities
3. **Unified metadata catalog:** Maintaining consistency across legacy and modern components

This approach allowed the organization to realize benefits incrementally, with each phase self-funding subsequent modernization efforts. Over three years, the retailer reduced operating costs by 40% while improving analytical capabilities and time-to-market (Assur . N. *et al.*,2022; Witcher, B. 2022).

Healthcare: Federated Governance in Highly Regulated Environments

A healthcare provider network implemented a composable architecture to balance centralized governance with distributed innovation across multiple facilities. The organization was established:

1. **Central policy registry:** Defining compliance requirements and controls
2. **Distributed enforcement:** Implementing controls at component interfaces

3. **Automated compliance verification:** Validating adherence to regulations
4. **Domain-aligned data products:** Enabling specialized analysis while maintaining compliance

This approach allowed individual facilities to adopt tailored solutions while ensuring consistent HIPAA compliance and patient privacy protections across the network (HIMSS).

Quantitative Analysis of Implementation Outcomes

Analysis of organizations that have successfully implemented composable data architectures reveals consistent patterns of improvement across several dimensions :

1. **Time-to-market:** reduction in new capability deployment time
2. **Technology adoption:** increase in the pace of new technology integration
3. **Governance effectiveness:** reduction in compliance-related incidents
4. **Technical debt:** Significant reduction in maintenance costs as a percentage of overall data spending

Organizations with mature implementations also report enhanced ability to attract and retain technical talent due to reduced legacy system maintenance burdens and increased opportunity to work with modern technologies (Gartner).

BENEFITS AND CHALLENGES

Empirical Benefits

Platform Agility and Technology Adoption

Organizations implementing composable data architectures consistently report significant improvements in platform agility and technology adoption capabilities. A 2023 survey of enterprise data leaders found that organizations with composable architectures were able to incorporate new technologies 2.7 times faster than those with traditional monolithic platforms (Patrick Connolly. 2023). This acceleration stems from reduced coupling between components, allowing targeted upgrades without system-wide impacts.

Key empirical benefits include:

- 40-60% reduction in time-to-market for new data capabilities
- Ability to evaluate and adopt specialized technologies for specific use cases
- Reduced risk of technological obsolescence through incremental modernization
- Enhanced ability to leverage open-source innovations alongside commercial solutions

Technical Debt Reduction

Composable architectures demonstrate measurable effectiveness in reducing technical debt accumulation in data platforms. By enabling incremental modernization and component-level updates, organizations can address technical debt in manageable increments rather than through disruptive "big bang" replacements.

Measured outcomes include:

- Reduction in maintenance costs as a percentage of overall data platform spending
- Decreased incidence of technical constraints blocking business initiatives
- Improved system reliability and performance through targeted optimizations
- Extended the useful lifespan of data investments through component-level refreshes

Governance Effectiveness

Contrary to initial concerns that decoupled architectures might weaken governance, evidence suggests that well-implemented composable approaches can enhance governance effectiveness. This improvement stems from consistent metadata management, automated policy enforcement at component interfaces, and improved visibility across the data lifecycle.

Quantifiable benefits include:

- 30-50% reduction in governance-related incidents and compliance violations
- Improved data quality through automated validation at integration points
- Enhanced auditability through comprehensive lineage tracking
- More efficient regulatory adaptation through targeted policy updates

IMPLEMENTATION CHALLENGE.

Organizational Resistance

Transitioning to composable architectures often encounters organizational resistance stemming from established practices, skill investments, and governance concerns. This resistance typically manifests as:

- Preference for familiar, integrated solutions from established vendors
- Concerns about fragmentation and increased operational complexity
- Resistance from teams with deep expertise in legacy platforms
- Hesitation from governance stakeholders regarding distributed control

Successful implementations address these concerns through incremental approaches, demonstrated quick wins, and comprehensive education about the benefits of composability.

Skills and Capability Gaps

Composable architectures require different skill sets than traditional data platforms, creating capability gaps during transition periods. Organizations frequently encounter challenges in:

- API design and management across heterogeneous components
- Metadata modeling and orchestration across diverse technologies
- Cloud-native deployment and operations for containerized components
- Integration of specialized analytical tools and data science frameworks

Leading organizations address these gaps through targeted hiring, internal training programs, and strategic partnerships with service providers during transition periods (Wells, D. 2018).

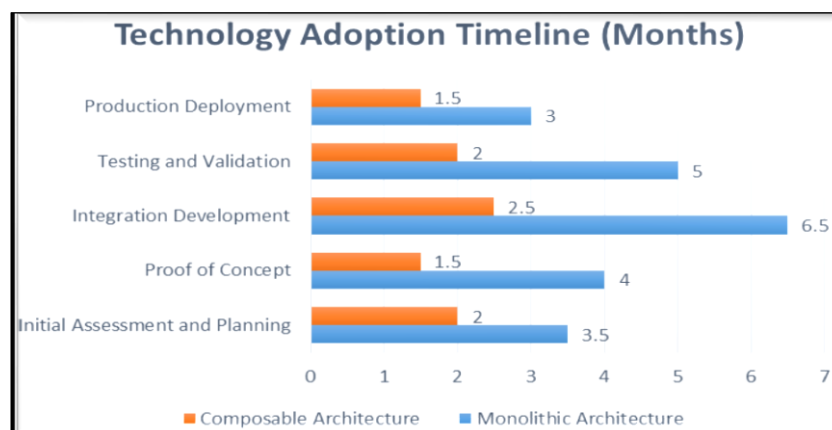


Fig 1: Illustrative Technology Adoption Timeline (Months) - Monolithic vs. Composable Architectures, Not Derived from any Production Systems

Integration Complexity

While composable architectures reduce long-term complexity, they can introduce integration challenges during implementation phases. Common difficulties include:

- Managing schema evolution across component boundaries
- Ensuring consistent security controls across heterogeneous technologies

- Maintaining performance across distributed processing components
- Debugging issues that span multiple independent components

Successful organizations mitigate these challenges through robust contract management, comprehensive testing frameworks, and investment in observability capabilities that provide visibility across component boundaries.

Table 2: Implementation Outcomes of Composable Data Architectures Across Industries (Patrick Connolly, 2023)

Industry	Primary Challenge	Implementation Focus	Key Benefits Realized
Financial Services	Regulatory compliance while enabling innovation	Decoupled risk reporting platform with consistent lineage	Reduced regulatory change implementation time; Enhanced auditability
Retail	Legacy modernization without business disruption	Data virtualization layer; Domain-specific data products	Reduced operating costs; Improved time-to-market for analytics
Healthcare	Balancing centralized compliance with distributed innovation	Central policy registry with distributed enforcement	Consistent HIPAA compliance across the network; Specialized analysis capabilities
Manufacturing	Complex supply chain visibility with legacy systems	Integration layer with metadata-driven orchestration	End-to-end process visibility; Reduced technical debt
Telecommunications	Real-time analytics across diverse data sources	Event-driven architecture with metadata standardization	Improved customer experience insights; Enhanced data quality

FUTURE DIRECTIONS

Emerging Technologies and Their Impact

1. Several emerging technologies promise to enhance the effectiveness of composable data architectures:
2. **Data contracts as code:** Formalized approaches to defining and enforcing contracts between data producers and consumers, enabling more robust decoupling.
3. **Knowledge graphs:** Providing richer semantic context for metadata, enhancing interoperability between components.
4. **Declarative configuration:** Advancing beyond infrastructure-as-code to comprehensive platform-as-code approaches that define entire data ecosystems declaratively.
5. **Policy-as-code frameworks:** Enabling automated governance enforcement across heterogeneous components through machine-readable policies.

These technologies will likely accelerate the adoption of composable approaches by reducing integration friction and enhancing governance capabilities.

AI/ML Integration in Composable Architectures

Artificial intelligence and machine learning workloads present both challenges and opportunities for composable data architectures. Future developments will likely focus on:

1. **Feature stores:** Specialized components for managing ML features across training and inference workflows.
2. **Model governance integration:** Extending data governance to include model lineage, bias monitoring, and regulatory compliance.
3. **Automated metadata generation:** Leveraging AI to enhance metadata completeness and quality across component boundaries.
4. **Intelligent orchestration:** Using ML to optimize workflow execution across distributed components (Kim Hazelwood. *et al.*).

Organizations implementing composable architectures today should establish patterns that accommodate these emerging capabilities to maximize long-term value.

Research Opportunities

Several research areas promise to advance the theoretical foundations and practical implementation of composable data architectures:

1. **Formal methods for component contracts:** Developing mathematical frameworks for specifying and verifying component interactions.
2. **Quantitative models of architectural flexibility:** Creating metrics to assess and compare the adaptability of different architectural approaches.

3. **Automated architecture synthesis:** Developing tools that can recommend optimal component configurations based on workload characteristics and business requirements.
4. **Cross-domain metadata standardization:** Establishing consistent metadata models that span traditional boundaries between operational, analytical, and AI systems.

Academic-industry partnerships in these areas could significantly accelerate the maturation of composable approaches and expand their applicability across domains.

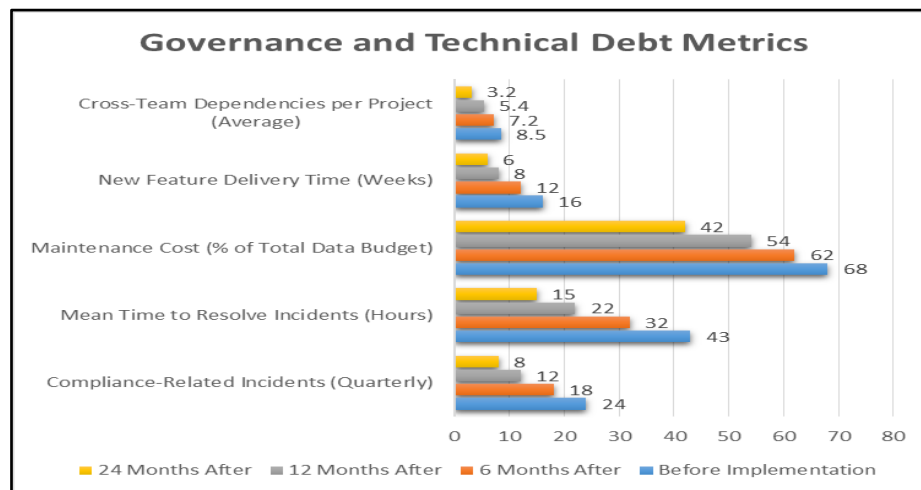


Fig 2: Illustrative Governance and Technical Debt Metrics Before and After Composable Architecture Implementation, Not Derived from any Production Systems

CONCLUSION

Composable data architectures represent a transformative approach to enterprise data management, offering a compelling alternative to traditional monolithic platforms that have dominated the landscape for decades. The article has demonstrated that composability delivers tangible benefits in platform agility, technical debt reduction, and governance effectiveness. The case studies across financial services, retail, and healthcare sectors illustrate that these benefits are achievable across diverse industries and regulatory environments. While challenges remain, particularly in organizational change management, skills development, and integration complexity, the trajectory of both technology and organizational practices favors composable approaches. As emerging technologies like data contracts, knowledge graphs, and AI-driven metadata management continue to mature, the article anticipates further acceleration in composable architecture adoption. Organizations that embrace these principles today position themselves for sustainable competitive advantage through data infrastructure that can evolve incrementally, adapt

to changing business requirements, and incorporate innovations without disruptive rewrites. The future of enterprise data management lies not in monolithic platforms but in thoughtfully composed ecosystems of specialized capabilities united through shared metadata, consistent governance, and clear contracts between components.

REFERENCES

1. Simoni G. D. , Beyer M , "Market Guide for Active Metadata Management," *Gartner, Inc.*, 14 November 2022. <https://assets.ctfassets.net/7p3vnbznfiw/1Re7aoxkTekQ3CDpZaihtZ/c79f7f5fb8ca143ac56821b585518412/market-guide-for-active-metadata-management.pdf>
2. Dehghani,Z "Data Mesh: Delivering Data-Driven Value at Scale," *O'Reilly Media, Inc.*, 2021. https://biconsult.ru/files/Data_warehouse/Data%20Mesh%20Delivering%20Data-Driven%20Value%20at%20Scale.pdf
3. Stonebraker, M., & Çetintemel, U. "One size fits all" an idea whose time has come and

- gone." *Making databases work: the pragmatic wisdom of Michael Stonebraker*. 2018. 441-462.
4. Strengholt,P "Data Management at Scale," *Medium*, Jul 11, 2023. <https://piethein.medium.com/data-management-at-scale-91118a1a7d83>
 5. Fowler,M. "Strangler Fig," 22 August 2024. <https://martinfowler.com/bliki/StranglerFigApplication.html>
 6. Assur, N., & Rowshankish, K. "The data-driven enterprise of 2025." *McKinsey & Company* 28 (2022).
 7. Patrick Connolly, "DataOps.live celebrates a breakout FY23 with 400 %+ growth, as demand for DataOps Products accelerates and market awareness grows," *DataOps.live*, Apr 01, 2023. <https://www.dataops.live/blog/dataops.live-celebrates-a-breakout-fy23-with-400-growth-as-demand-for-dataops-products-accelerates-and-market-awareness-grows>
 8. Wells, D. "Data Engineering Coming of Age," *Eckerson Group*, Apr 01, 2018. <https://www.eckerson.com/articles/data-engineering-coming-of-age>
 9. Hazelwood, K., Bird, S., Brooks, D., Chintala, S., Diril, U., Dzhulgakov, D., ... & Wang, X. "Applied machine learning at facebook: A datacenter infrastructure perspective." *2018 IEEE international symposium on high performance computer architecture (HPCA)*. IEEE, 2018.
 10. Deloitte, "2023 banking regulatory outlook", <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/regulatory/us-banking-regulatory-outlook-2023.pdf>
 11. Witcher, B. Oct 26 2022, Forrester, "Predictions 2023: Bold Shifts In Retail Models Will Be The Salve For Uncertain Times".<https://www.forrester.com/blogs/predictions-2023-shift-in-retail-models/>
 12. HIMSS "Understanding National Interoperability Frameworks and Networks " :https://www.himss.org/sites/hde/files/media/file/2024/08/02/himss_interopframeworks_ebook-2.pdf
 13. Gartner, " How to Attract and Retain Talent for Your Data and Analytics Team". <https://www.gartner.com/en/articles/how-to-attract-and-retain-talent-for-your-data-and-analytics-team>

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

Cite this article as:

Joshi. R" Composable Data Architectures: Moving Beyond the Monolithic Platform. " *Sarcouncil Journal of Engineering and Computer Sciences* 4.6 (2025): pp 232-240.