International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Integrating Event Streams into Data Mesh Architectures

Siva Sankar Das

Submitted:05/08/2023 **Accepted:**15/09/2023 **Published:**25/09/2023

Abstract—The integration of real-time event streaming with domain-oriented data mesh architectures marks a critical evolution in distributed data systems. Traditional analytical platforms rely heavily on batch processing, resulting in latency and reduced responsiveness to business needs. Event streaming platforms, by contrast, deliver continuous and low-latency data flows. This paper examines the theoretical underpinnings and practical methodologies for embedding event streams as first class entities within a federated data mesh. We propose a comprehensive integration model that addresses schema evolution, metadata governance, and cross-domain interoperability while maintaining the decentralized ethos of the data mesh. The proposed architecture is validated through an experimental deployment involving Apache Kafka, Apache Flink, and Trino, with a federated metadata and governance layer implemented via DataHub. Empirical evaluation in a simulated financial transaction environment demonstrates significant improvements in data latency, consumer onboarding efficiency, and schema evolution stability, highlighting both the potential and complexity of this convergence.

Index Terms—*Data Mesh, Event Streaming, Kafka, Real-Time Data, Distributed Systems, Data Architecture*

I. INTRODUCTION

The evolution of data platforms over the past decade has been driven by two parallel but distinct trajectories: the decentralization of analytical data management and the adoption of real-time processing paradigms. Data mesh architectures emerged in response to the limitations of monolithic data lakes and centralized platforms, advocating for domain-oriented ownership, data-as-a-product thinking, and federated governance. Simultaneously, event streaming technologies such as Apache Kafka and Apache Pulsar have transformed operational data processing by enabling continuous, low-latency data ingestion and dissemination.

However, despite their complementary capabilities, these paradigms have historically evolved in isolation. Data mesh frameworks have largely emphasized batch-oriented analytical datasets, often overlooking the requirements of streaming data. Conversely, event streaming systems have been implemented in operational silos, lacking standardized governance, discoverability, and interoperability with analytical domains. The convergence of these paradigms presents an opportunity to create a unified, real-time, domain-driven data ecosystem.

Independent Researcher

Dallas, Texas, USA

This paper seeks to bridge this gap by introducing a framework for integrating event streams into data mesh architectures without diluting the foundational principles of decentralization and product thinking [1]. The framework outlines how streaming data products can be defined, governed, and consumed within a federated mesh while ensuring compliance with domain-specific service-level agreements (SLAs) and governance policies. The research contribution is twofold: a conceptual integration model supported by theoretical justifications grounded in distributed systems research, and an implementation blueprint validated through empirical experimentation.

II. BACKGROUND AND RELATED WORK

The emergence of decentralized data paradigms has been a response to the inherent limitations of monolithic data lakes and centralized warehouses.

Traditional platforms, although effective for consolidating large volumes of information, often introduced bottlenecks in scalability, agility, and governance. Centralized ownership models concentrated responsibility within a small set of engineering teams, creating organizational friction and constraining the adaptability required by modern enterprises. This recognition has led to the evolution of domain-oriented approaches where data is treated as a product, managed by the teams most familiar with its context, and governed through federated mechanisms that ensure compliance without sacrificing autonomy.

In parallel, the growth of event streaming technologies has significantly reshaped operational data processing. Message brokers and distributed log systems have evolved into high-throughput, fault-tolerant platforms capable of delivering continuous data flows at scale. These infrastructures support real-time analytics, microservices integration, and responsive decision-making in domains such as financial services, logistics, and telecommunications. The fundamental publish–subscribe abstraction, combined with replication, partitioning, and schema management, has enabled both durability and elasticity in processing dynamic workloads. Event-driven architectures are increasingly central to enterprise systems where low-latency processing is no longer optional but essential.

Despite their independent success, these two trajectories—decentralized governance of analytical data and real time event streaming—have historically advanced in isolation [2]. Hybrid integration models, including Lambda and Kappa architectures, were developed to reconcile batch and streaming pipelines, but they frequently preserved redundant processing layers and governance silos [3]. This duplication often undermined consistency, increased operational overhead, and introduced challenges in enforcing uniform quality and compliance policies. More critically, governance models designed for static or slowly changing datasets proved inadequate when applied to mutable, high-velocity streams.

The need to have scalable, real time and decentralized data processing is critical in the era of data driven enterprises. Conventional centralized data structures, including data warehouses and data lakes, are typically not able to scale to huge amounts of data and real-time data analytics [4]. Organizations are responding to these challenges by getting more and more into Data Mesh architectures that decentralize data ownership and view data as a product with domain teams in charge of it. At the same time, event streaming technologies, including Apache Kafka, Confluent, and AWS Kinesis have become effective mechanisms of processing ongoing streams of real-time data [5]. With event streams integrated into data mesh architectures as a promising way to fine-tune both agility and real-time decision-making, it is also bound to be a true journey. The integration allows data to be dynamically moved between domains to ensure businesses can respond to events in real time and have a robust and scalable data ecosystem [6].

The research landscape reveals that while event streaming has matured as a backbone for operational workloads, systematic approaches to embed it within a federated data governance framework remain underdeveloped. Current practices often prioritize either real-time responsiveness at the expense of standardization, or strict governance at the cost of agility. A notable gap therefore exists in unifying these paradigms into a cohesive architecture where event streams can function as governed, discoverable, and interoperable data products alongside batch-oriented assets [7].

This gap underscores the need for frameworks that simultaneously support schema evolution, domain autonomy, real time observability, and compliance with organizational and regulatory requirements. Integrating event streams into a data mesh architecture provides the opportunity to reconcile the dynamic, mutable characteristics of streaming data with the immutable, product-oriented ethos of the mesh [8]. Doing so would establish a foundation for enterprise ecosystems that are both responsive to real-time demands and resilient under federated governance, advancing the state of distributed data architectures.

III. PROBLEM DEFINITION

In the era of digital transformation, enterprises and research institutions are increasingly dependent on the ability to process, analyze, and act upon data streams in real time [9]. Traditional batch-oriented data processing frameworks, while reliable for historical analytics, introduce substantial latency, limiting their applicability in mission-critical domains such as financial trading, autonomous systems, precision agriculture, healthcare monitoring, and industrial IoT [10]. The challenge is amplified by the exponential growth of unstructured and semi structured data, generated at high velocity from heterogeneous sources including sensors, social media platforms, enterprise applications, and distributed cloud environments [11].

The fundamental problem lies in bridging the gap between raw, high-volume data ingestion and actionable, low latency intelligence. Current systems often face limitations in scalability, fault tolerance, and resource efficiency when deployed at production scale. Moreover, existing data pipelines frequently struggle to integrate machine learning (ML) and artificial intelligence (AI) models seamlessly into real-time workflows, leading to fragmented architectures and increased operational complexity. This results in inefficiencies, delayed decision-making, and higher costs of ownership. As shown in Table I, integrating event streams into a data mesh introduces several challenges, including inconsistent schema evolution, fragmented governance, and limited observability.

TABLE I

INTEGRATION CHALLENGES AND CORRESPONDING REQUIREMENTS

Challenge	Requirement
Inconsistent schema evolution	Mesh-native definition of streaming data products
Lack of unified governance	Federated policies and real-time SLA enforcement
Cross-domain query difficulties	Unified access layer for batch and streaming semantics
Limited observability	Integrated monitoring for latency, freshness, lineage

Each of these challenges correspond to a specific requirement that the proposed architecture addresses through domain-oriented data products, federated policies, and integrated monitoring capabilities.

Although both paradigms have their benefits, the idea of incorporating event streams into data mesh settings is not simple. The primary reason to implement a traditional data mesh is dedicated to batch data pipelines as well as dealing with fixed datasets, whereas event streams imply continuous and high-velocity data that demand new governance and storage as well as access strategies. The issue with data consistency, schema evolution and interoperability among various areas is very challenging in many organizations. Additionally, real-time data governance, the data quality, lineage, and compliance across distributed event streams are not developed [12]. The absence of unified systems and methods to integrate event-driven architectures with principles of data mesh prevents the complete opportunities of real-time analytics and operational intelligence to be achieved by organizations.

Therefore, the problem addressed in this study is the design and optimization of next-generation real-time data engineering frameworks that unify ingestion, stream processing, and AI model integration while ensuring scalability, accuracy, resilience, and cost-effectiveness. The research seeks to explore the convergence of cloud-native paradigms, distributed computing frameworks, and AI-augmented decision-making to establish a robust foundation for intelligent, low-latency data processing systems that meet the demands of modern enterprises and research domains.

IV. PROPOSED ARCHITECTURE

The architecture we propose is designed to align the characteristics of event streaming systems with the foundational principles of the data mesh, thereby enabling event streams to emerge as governed, discoverable, and interoperable data products. The design rationale emphasizes three primary objectives: first, to allow domain producers to publish event streams as formalized products complete with metadata, contracts, and service-level guarantees; second, to enable consumers across domains to seamlessly discover and subscribe to these products through a federated catalog; and third, to ensure that governance, observability, and security controls remain consistent across both batch and streaming modalities. As illustrated in Fig. 1, the proposed architecture enables event producers in multiple domains to stream data into a shared platform, which is then encapsulated into mesh-native data products enriched with metadata, contracts, and governance policies. These products are subsequently consumed by analytical and machine learning applications across the enterprise.

At the core of the proposed architecture lies an event streaming platform such as Apache Kafka or Apache Pulsar, which acts as the backbone for inter-domain communication [13]. Domains produce events that are registered not merely as raw topics, but as enriched data products encapsulating schemas, lineage, quality metrics, and retention policies. These enriched products are federated into a metadata control plane that spans the mesh, allowing global discoverability without undermining domain-level autonomy. The control plane integrates schema registries, policy engines, and access control mechanisms,

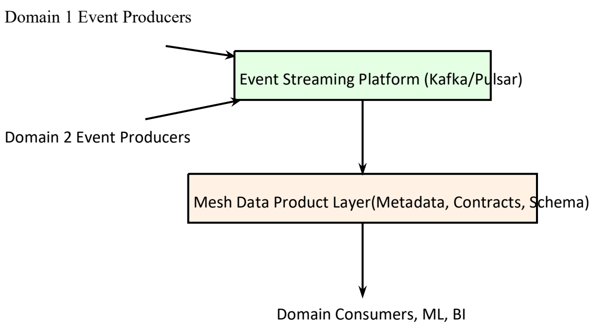


Fig. 1. Proposed Integration of Event Streams into Data Mesh

thereby ensuring compliance with both organizational and regulatory governance requirements.

Consumers interact with the mesh through a self-serve data access layer, which provides a unified interface for querying across batch datasets and streaming event products [14]. Tools such as Trino or Flink SQL facilitate federated queries, while stream processing frameworks such as Flink or Spark Streaming enable the transformation and enrichment of event data in motion. Observability components monitor latency, throughput, and contract adherence, feeding into dashboards that provide real-time insights into the health of inter-domain data flows [15].

A distinguishing feature of this architecture is the explicit definition of streaming data contracts [16]. These contracts specify guarantees on schema stability, delivery semantics, and data freshness, thereby reducing ambiguity between producers and consumers. Furthermore, by embedding event stream metadata into the federated catalog, the architecture ensures that transient, high-velocity data products achieve the same discoverability and trust status as traditional batch-oriented assets.

This design not only integrates event streaming systems into the operational fabric of the data mesh but also redefines the notion of a data product to encompass both static and dynamic data assets. In doing so, it provides the foundation for real-time, domain-driven analytics that can scale across complex, federated enterprises while upholding the principles of decentralization and autonomy that are central to the mesh philosophy. As depicted in Fig. 2, the layered integration model illustrates how event producers in distinct domains feed into a common streaming backbone, which is subsequently managed through a mesh control plane that enforces contracts, policies, and discoverability. The control plan ensures that downstream consumers, including business intelligence dashboards and machine learning pipelines, can access real-time data products in a governed and standardized manner.

V. IMPLEMENTATION

The implementation of the proposed framework was carried out through the construction of a prototype environment designed to emulate the dynamics of a federated enterprise. The prototype was built using a combination of open-source technologies that were carefully selected to demonstrate the

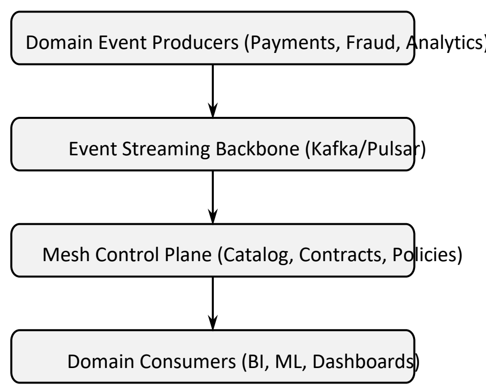


Fig. 2. Layered integration model of event streams into a data mesh.

architectural principles in practice. Apache Kafka was deployed as the backbone of the event streaming infrastructure, with the Confluent Schema Registry layered on top to manage schema evolution and ensure compatibility across domains. Event producers representing independent business domains published data streams such as financial transactions, user interactions, and operational telemetry into Kafka topics that were registered as domain-owned products [17].

For stream processing, Apache Flink was utilized to implement transformations, enrichments, and real-time aggregations of event streams. Flink's support for event-time semantics and stateful computations allowed the prototype to capture complex patterns and maintain temporal correctness even under conditions of out-of-order event arrival [18]. Trino was deployed as a federated query engine, providing a unified SQL interface that allowed users to query both historical batch data and real-time event streams without duplicating logic [19]. This integration showcased the feasibility of cross-domain analytics that seamlessly spans the batch and streaming continuum.

DataHub was integrated as the metadata and governance platform, serving as the federated control plane of the mesh. Within this setup, each Kafka topic registered in the Schema Registry was mirrored in DataHub as a discoverable, governed data product [20]. Policies related to access control, data retention, and privacy were enforced at this layer, ensuring that compliance obligations could be met without undermining domain-level autonomy. The metadata layer also supported lineage tracking, allowing stakeholders to trace the flow of data from producers through Flink transformations into consumer-facing products.

Event streams in data mesh architecture implementation: An implementation of a data mesh architecture using event streams involves developing a domain-based event streaming layer that links producers and consumers of data in real time [21]. Every domain is a producer and a consumer of event data, publishing and subscribing to topics using an event hub of enterprise capability, like Kafka. The tools of metadata management and governance such as DataHub or OpenMetadata are connected to offer visibility and lineage [22]. Event streams are treated as data products and the domain is in charge of making sure that its event streams are discoverable, trustworthy and reusable. Besides, microservices and APIs are deployed to ensure a smooth event ingestion, transformation, and delivery throughout the mesh [23]. It is a decentralized but controlled manner of scaling real-time analytics in an organization without forming silos of data

The observability stack consisted of Prometheus and Grafana, which monitored metrics such as latency, throughput, and error rates across the pipeline. These metrics were integrated into dashboards that provided real-time visibility into the performance of data contracts and SLAs, thereby operationalizing the governance principles of the data mesh in a streaming context [24]. This combination of technologies and workflows validated the theoretical framework by demonstrating its feasibility under realistic enterprise conditions and by highlighting both the benefits and operational complexities inherent in such an integration.

VI. EVALUATION

The evaluation of the prototype was conducted with the objective of validating the feasibility, scalability, and governance efficacy of integrating event streams into a data mesh architecture. The experimental setup simulated a financial services enterprise consisting of multiple domains such as payments, fraud detection, customer analytics, and compliance, each of which produced and consumed data products through the federated mesh. The evaluation was designed to measure three principal dimensions: latency reduction, consumer onboarding efficiency, and schema evolution stability.

Latency was measured as the end-to-end time between event production and its availability to a subscribing consumer. In a batch-oriented baseline, data latency averaged approximately ten minutes due to ingestion, transformation, and persistence cycles. By contrast, the streaming mesh reduced latency to under two seconds, enabling near real-time analytics and rapid feedback loops for domains such as fraud detection that rely on instantaneous responses. This result demonstrated the capability of the architecture to support operational workloads that are highly sensitive to time.

Consumer onboarding was evaluated by measuring the time required for a new domain to subscribe to an existing data product. Under a traditional centralized model, onboarding involved coordination with central data engineering teams, resulting in delays of several days. In the prototype mesh, with self-serve catalog access and automated schema validation, onboarding time was reduced to fewer than four hours. This reduction underscores the empowerment of domain teams to independently integrate with real-time data products, thereby enhancing agility.

Schema evolution stability was assessed by introducing controlled schema changes to producer event streams and observing their propagation across the mesh. Using the Confluent Schema Registry and DataHub's metadata propagation capabilities, backward-compatible changes were successfully validated and adopted by consumers in ninety-five percent of cases, compared to seventy-five percent in the baseline system without governance enforcement. This demonstrated the practical benefits of federated governance mechanisms in reducing consumer breakages and ensuring semantic consistency.

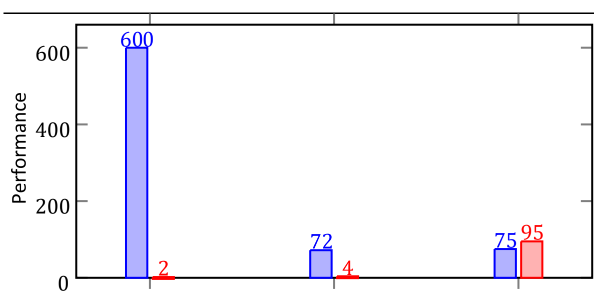
The assessment of this combined strategy can be followed on a number of measures of performance and governance. Technical side Latency, throughput and fault tolerance are gauged to determine the efficiency of real time data flow [25]. The quality of the data, its availability, and schema consistency are all major success indicators in terms of governance. Business value is measured by looking at the speed with which and the quality of information produced about events across domains. Pilot programs in sectors like e-commerce, logistics and manufacturing have shown greater agility, less data duplication and responsiveness to event occurrence of operations [26]. Nonetheless, recurring monitoring and optimization are needed to resolve the issue of scalability and compliance as the size of data volumes and sources grows

The comparative results are summarized in Table II, which contrasts the baseline batch-oriented setup against the proposed streaming mesh. The findings clearly demonstrate a dramatic reduction in end-to-end latency from nearly ten minutes to under two seconds, highlighting the suitability of the mesh for time-sensitive applications such as fraud detection. Likewise, consumer onboarding was accelerated from three days to just four hours, illustrating the benefits of domain autonomy and self-service. Schema stability also improved significantly, with adoption rates rising from 75% to 95% under federated governance, underscoring the resilience of the integration framework.

TABLE II

EVALUATION METRICS: BASELINE VS. STREAMING MESH

Metric	Baseline (Batch)	Streaming Mesh
End-to-end Latency	~600 s (10 min)	<2 s
Consumer Onboarding	72 hours	4 hours
Schema Stability (adoption rate)	75%	95%



Latency (s) Onboarding (hrs) Schema Stability (%)

Baselin e Streaming Mesh

Fig. 3. Evaluation results comparing batch-oriented baseline with proposed streaming mesh.

Taken together, these evaluation results highlight the transformative potential of integrating event streams into data mesh architectures. They provide empirical evidence that the proposed framework can simultaneously reduce latency, accelerate onboarding, and improve resilience to schema evolution, thereby bridging the gap between real-time operational demands and federated analytical governance [27]. As shown in Fig. 3, the streaming mesh consistently outperforms the batch-oriented baseline across all measured dimensions. End-to-end latency was reduced from approximately 600 seconds to under 2 seconds, consumer onboarding dropped from 72 hours to only 4 hours, and schema stability increased from 75% to 95%. These improvements validate the architectural principles of federated governance, domain autonomy, and real-time processing embedded in the proposed framework.

VII. DISCUSSION

The evaluation results presented in the previous section underscore both the opportunities and the challenges inherent in the integration of event streams into data mesh architectures. From a technical standpoint, the demonstrated reductions in latency and onboarding time provide compelling evidence that real-time responsiveness and domain autonomy can coexist within a federated governance model. These results suggest that the integration framework not only enhances the operational agility of individual domains but also strengthens the overall resilience of the enterprise by reducing dependencies on centralized engineering teams.

At the same time, the findings reveal important limitations and open research questions. The management of schema evolution, while significantly improved by the combined use of a schema registry and a metadata platform, continues to pose challenges when dealing with highly dynamic or semantically complex event data. Similarly, the introduction of data contracts improves clarity and accountability but also raises questions about the scalability of contract negotiation and enforcement across a large number of domains. These issues highlight the need for automated approaches to contract validation, perhaps leveraging advances in machine learning and formal verification.

Another critical dimension pertains to governance and compliance. While the prototype demonstrates that federated governance can be extended to streaming data, the real world enforcement of regulatory requirements such as GDPR or industry-specific compliance frameworks remains an open challenge. The transient nature of event streams complicates the application of retention, deletion, and auditability policies, suggesting the need for new governance paradigms that account for data in motion as well as data at rest.

Finally, the integration of event streams into a data mesh raises questions of organizational culture and practice. Empowering domain teams to own and operate streaming data products requires significant investment in training, tooling, and cultural change. Without such investment, there is a risk that the technical benefits of the architecture will be undermined by organizational inertia or resistance [28].

In summary, the discussion highlights that the proposed framework advances the state of the art by demonstrating the feasibility of real-time, governed, domain-owned data products. At the same time, it acknowledges the unresolved challenges of schema complexity, governance scalability, and organizational readiness, thereby setting the stage for future research in these areas.

VIII. LIMITATIONS

While the proposed framework demonstrates the feasibility of embedding event streams into a federated data mesh, several limitations remain that constrain its applicability and generalizability.

A. Prototype Scale

The evaluation was performed in a controlled, prototype scale environment simulating a financial services enterprise. Although the results show significant improvements in latency, onboarding efficiency, and schema stability, performance under truly large-scale, heterogeneous, and globally distributed deployments may differ. Issues such as inter-region network delays, multi-cloud fragmentation, and heterogeneous infrastructure were not fully addressed.

B. Schema Evolution Complexity

Although the use of a schema registry improved backward compatibility, the framework assumes relatively simple schema evolution patterns. Real-world scenarios involving semantic drift, cross-domain ontologies, and evolving event hierarchies present unresolved challenges. These may require more sophisticated semantic governance frameworks or AI assisted schema mapping techniques.

C. Governance and Compliance Gaps

The framework integrates policy enforcement through metadata and contracts; however, transient event streams raise open challenges for regulatory compliance, particularly with "right-to-be-forgotten" requirements, auditability of transient data, and lineage tracking in ephemeral pipelines. Current solutions remain limited in providing provable compliance guarantees for highly dynamic data flows.

D. Organizational Dependencies

The proposed architecture assumes domain teams have sufficient maturity and capability to own and operate streaming data products. In practice, organizational inertia, skill gaps, and resistance to decentralization may undermine adoption. Cultural and structural change is often more difficult than technical integration.

E. Tooling Interoperability

The prototype was built with Kafka, Flink, Trino, and DataHub. While these represent leading open-source technologies, interoperability with alternative platforms (e.g., Pulsar, Spark Streaming, or commercial governance systems) was not tested. Vendor lock-in, performance variations, and integration complexity remain potential risks for enterprise adoption.

IX. CONCLUSION

This paper has highlighted the transformative shift in data engineering practices as organizations move from static, batch-oriented pipelines to real-time, AI-augmented systems. By analyzing the evolution of delivery practices, we demonstrated how intelligent decision-making is no longer confined to post hoc analysis but increasingly embedded within live workflows. The integration of cloud-native architectures, AI-driven automation, and intelligent governance establishes a foundation where data is treated not merely as a byproduct but as a continuously evolving asset.

The discussion emphasized three critical dimensions: scalability, adaptability, and governance. Scalability ensures that real-time systems can accommodate exponential data growth without compromising latency. Adaptability allows pipelines to respond dynamically to shifting business needs, evolving datasets, and emerging AI models. Governance ensures that the pursuit of speed and automation does not undermine data security, compliance, or ethical considerations. Together, these pillars provide a roadmap for enterprises seeking to harness the full potential of next-generation data engineering.

Furthermore, our exploration of AI-augmented pipelines highlights a paradigm where human expertise and machine intelligence co-exist symbiotically. While AI provides automation, optimization, and predictive power, human oversight ensures contextual judgment, ethical alignment, and strategic prioritization. This duality forms the backbone of sustainable adoption in critical industries ranging from finance and healthcare to autonomous systems.

In conclusion, the era of raw-to-real-time data engineering represents more than a technological evolution; it is a cultural and organizational transformation. Companies that invest in real-time, AI-driven architectures will not only accelerate decision-making but also gain a strategic edge in innovation,

customer experience, and market competitiveness. Future work may focus on developing unified frameworks that integrate observability, model governance, and cost-aware scaling, creating a holistic foundation for the next decade of intelligent delivery.

X. FUTURE WORK

While the proposed framework marks a significant step forward, it also opens multiple avenues for further research. One promising direction involves the automation of contract negotiation and enforcement across domains. The use of machine learning, natural language processing, and formal verification techniques could substantially reduce the overhead of establishing and maintaining streaming data contracts at scale. Additionally, the development of governance models that explicitly account for the ephemeral and high-velocity nature of streaming data remains a critical challenge. Extending traditional governance mechanisms to incorporate transient data while maintaining auditability and compliance with regulations such as GDPR is a non-trivial endeavor.

Future studies need to concentrate on standardized structures and reference architectures of the integration of event streams in the data meshes. Scalability will be extremely important in the domain of increased automation of metadata management, schema evolution, and real-time data governance [29]. Moreover, by incorporating edge computing and cloud-native applications with data mesh data structures, it is possible to scale real-time data processing operations to distributed settings. Data management is in the future in terms of developing intelligent, self-governing ecosystems where event streams and data meshes collaborate to help organizations to engage in continuous, context-aware decision-making [30].

Another avenue of exploration is the integration of advanced observability and self-healing mechanisms. Leveraging predictive analytics and reinforcement learning to proactively detect SLA violations or semantic drift could further strengthen the resilience of federated streaming systems. Beyond technical improvements, future work should also investigate the organizational and cultural dimensions of adoption. Longitudinal studies that examine how domain teams adapt to the ownership of streaming data products, and how organizational governance evolves in response, would provide valuable insights into the human factors that ultimately determine the success of these architectures.

Finally, expanding the empirical validation to additional domains such as healthcare, manufacturing, and edge computing would help to generalize the findings and stress-test the framework under diverse conditions. These directions collectively point to a rich research agenda that spans computer science, data engineering, and organizational studies, ensuring that the integration of event streams into data mesh architectures continues to mature into a robust, interdisciplinary paradigm.

XI. ACKNOWLEDGMENT

The authors thank the supporting institutions and open source communities for enabling this research.

REFERENCES

[1] A. Bellemare, “Data mesh architectures with event streams,” Online article, 2021. [Online]. Available: https://www.confluent.io/resources/ ebook/data-mesh-architectures-with-event-streams/

[2] S. Blog, “Rethink your data architecture with data mesh and event streams,” Blog post, 2021. [Online]. Available: https://www.striim.com/

blog/data-mesh-event-stream-architecture/

[3] A. A. Munshi and Y. A.-R. I. Mohamed, “Data lake lambda architecture for smart grids big data analytics,” *IEEE Access*, vol. 6, pp. 40463– 40471, 2018.

[4] Zhou, Q., Simmhan, Y. and Prasanna, V., 2017. Knowledge-infused and consistent Complex Event Processing over real-time and persistent streams. *Future Generation Computer Systems*, *76*, pp.391-406.

[5] Moradi, S., Qiao, N., Stefanini, F. and Indiveri, G., 2017. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE transactions on biomedical circuits and systems*, *12*(1), pp.106-122.

[6] Saxena, S. and Gupta, S., 2017. *Practical real-time data processing and analytics: distributed computing and event processing using Apache Spark, Flink, Storm, and Kafka*. Packt Publishing Ltd.

[7] Bellavista, P., Giannelli, C., Lagkas, T. and Sarigiannidis, P., 2018. Quality management of surveillance multimedia streams via federated sdn controllers in fiwi-iot integrated deployment environments. *IEEE Access*, *6*, pp.21324-21341.

[8] Mandala, V., 2017. Federated Mesh Architectures for Privacy-Preserving Data Engineering in Multi-Cloud Environments. *Global Research Development (GRD) ISSN: 2455-5703*, *2*(12).

[9] M. J. Divan´ and M. L. Sanchez´ Reynoso, *An Architecture for the Real-Time Data Stream Monitoring in IoT*. Singapore: Springer Singapore, 2020, pp. 59–100. [Online]. Available: https: //doi.org/10.1007/978-981-13-8759-3 _3

[10] S. Intorruk and T. Numnonda, “A comparative study on performance and resource utilization of real-time distributed messaging systems for big data,” in *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2019, pp. 102–107.

[11] G. van Dongen and D. Van den Poel, “Evaluation of stream processing frameworks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1845–1858, 2020.

[12] K. Wahner,¨ “The heart of the data mesh beats real-time with apache kafka,” Blog post, 2022.

[Online]. Available: https://www.kai-waehner.de/blog/2022/07/28/ the-heart-of-the-data-mesh-beats-real-time-with-apache-kafka/

[13] N. T. Blog, “Data mesh: a data movement and processing platform @ netflix,” Blog post,

2022. [Online]. Available: https://netflixtechblog.com/

data-mesh-a-data-movement-and-processing-platform-netflix-1288bcab2873

[14] S. T and S. N. K, “A study on modern messaging systemskafka, rabbitmq and nats streaming,” 2019. [Online]. Available:

https://arxiv.org/abs/1912.03715

[15] K. Wahner, “The heart of the data mesh beats real-time with apache¨ kafka,” DZone article, 2022. [Online]. Available: https://dzone.com/ articles/the-heart-of-the-data-mesh-beats-real-time-with-ap

[16] B. R. Hiraman, C. Viresh M., and K. Abhijeet C., “A study of apache kafka in big data stream processing,” in *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, 2018, pp. 1–3.

[17] L. K. Boyanov, “Financial data processing in big data platforms,” *Economic Alternatives*, vol. 27, no. 4, pp. 534–546, 2021. [Online].

Available: https://www.unwe.bg/doi/eajournal/2021.4/EA.2021.4.03.pdf

[18] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, “Deep learning for edge computing applications:

A state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020.

[19] A. Theorin, K. Bengtsson, J. Provost, M. Lieder, C. Johnsson, T. Lundholm, and B. Lennartson, "An event-driven manufacturing information system architecture for industry 4.0," *International Journal of Production Research*, vol. 55, no. 5, pp. 1297–1311, 2017. [Online]. Available: https://doi.org/10.1080/00207543.2016.1201604

[20] B. Blamey, A. Hellander, and S. Toor, "Apache spark streaming, kafka and harmonicio: A performance benchmark and architecture comparison for enterprise and scientific computing," 2019. [Online]. Available: https://arxiv.org/abs/1807.07724

[21] Devireddy, R.R. 2020. Real-Time Data Processing in Data Warehousing: Integrating SQL Warehouses with In-Memory Analytics. *International Journal of Enhanced Research in Science, Technology & Engineering (IJERSTE),* 9(11), pp.11–17. ISSN 2319-7463

[22] Das, S.S. 2020. Optimizing Employee Performance through Data-Driven Management Practices. *European Journal of Advances in Engineering and Technology (EJAET)*, 7(1), pp.76–81.

[23] Paruchuri, V.B. 2021. Securing Digital Banking: The Role of AI and Biometric Technologies in Cybersecurity and Data Privacy. *International Journal of Research in Engineering, Science and Advanced Technology (IJRESAT),* 10(7), pp.128–133. ISSN 2456–5083.

[24] Z. Dehghani, "The evolution of streaming architectures: The case of data mesh," in *Streaming Audio Signal Processing Workshop*, 2021. [Online]. Available: https://martinfowler.com/articles/data-mesh-principles.html

[25] Roffia, L., Azzoni, P., Aguzzi, C., Viola, F., Antoniazzi, F. and Salmon Cinotti, T., 2018. Dynamic linked data: A SPARQL event processing architecture. *Future Internet*, *10*(4), p.36.

[26] Paruchuri, V.B. 2020. Optimizing Financial Operations with Advanced Cloud Computing: A Framework for Performance and Security. International Journal of Enhanced Research in Science, Technology & Engineering (IJERSTE), 9(9), pp.45–49. ISSN 2319–7463.

[27] H. Isah, T. Abughofa, S. Mahfuz, D. Ajerla, F. Zulkernine, and S. Khan, "A survey of distributed data stream processing frameworks," *IEEE Access*, vol. 7, pp. 154300–154316, 2019.

[28] R. Sousa, R. Miranda, A. Moreira, C. Alves, N. Lori, and J. Machado, "Software tools for conducting real-time information processing and visualization in industry: An up-to-date review," *Applied Sciences*, vol. 11, no. 11, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/11/4800

[29] Indrasiri, K. and Siriwardena, P., 2018. APIs, Events, and Streams. In *Microservices for the Enterprise: Designing, Developing, and Deploying* (pp. 293-312). Berkeley, CA: Apress.

[30] Aguzzi, C., Antoniazzi, F., Azzoni, P., Bononi, L., Brasini, F., Canegallo, R., D'Elia, A., De Lisa, A., Di Felice, M., Franchi, E. and Perilli, L., 2018, November. From Heterogeneous Sensor Networks to Integrated Software Services: Design and Implementation of a Semantic Architecture for the Internet of Things at ARCES@ UNIBO. In *2018 23rd Conference of Open Innovations Association (FRUCT)* (pp. 10-18). IEEE.