



Adaptive Multi-Cloud Lakehouses for Real-Time Analytics

Luca Bianchi

Asst Professor

Abstract

Enterprise analytic frameworks increasingly rely on multi-cloud architectures, yet these systems are rarely described in a systematic, evidence-based way. This paper presents and evaluates a representative lakehouse-based architecture designed to deliver near-real-time analysis of user- and system-generated data across cloud providers. The architecture addresses three critical multi-cloud processing requirements: adaptive resource scaling, cross-cloud data consistency, and support for data governance and privacy policies. We survey the supporting technologies that underpin these requirements, including dynamic resource orchestration, cross-cloud data synchronisation, integration with data governance solutions, and streaming ingestion, querying, and processing. Together, these findings inform a flexible, adaptive architecture for near-real-time analytics in multi-cloud data ecosystems.

Large-scale, multi-user information systems deployed as public cloud services continuously generate events, alerts, and notifications. Combined with user-generated inputs, this telemetry can be exploited within a multi-cloud architecture to enrich the enterprise information model, enforce privacy policies, and monitor service-level agreements. Building the cloud-native data pipelines needed to support real-time analytics across such multi-cloud ecosystems raises significant challenges, particularly in meeting strict latency constraints while upholding safety and privacy requirements. Meeting these demands imposes specific technological constraints on data processing, data lake/data warehouse integration, and the provision of cross-cloud data consistency and synchronisation.

Keywords: Cloud computing; data integration; data management; data warehousing; enterprise architecture; enterprise information systems; data analytics; analytics; data governance; policy-based access control.

1. Introduction

The growth of multi-cloud environments is driven by enterprises' urgency to ensure resiliency, reduce costs, and increase flexibility. Nevertheless, the emergence of distributed data ecosystems aggravates the difficulty in maintaining synchronized datasets across different data clouds, resulting in higher engineering costs and data discrepancies that greatly hinder enterprise analytics. Additionally, the full potential of a multi-cloud data ecosystem is not currently exploited. Resource skews and predictably high data processing workloads lead to the significantly correlated and unexpected spike of processing workloads in one cloud, raising the importance of resource orchestration and task scheduling across clouds to fit workload changes. Though lakehouse-driven architectures are becoming mainstream for enterprise analytics, important requirements—such as cross-cloud data consistency and governance and real-time data processing using streaming ingestion and processing—have not yet been adequately addressed.

To this end, an adaptive multi-cloud lakehouse architecture is proposed, targeting real-time enterprise analytics by integrating streaming data processing technologies, dynamic cross-cloud resource orchestration, and policy-based access control into lakehouse paradigms. These additions form an architecture named adaptive lakehouse in a multi-cloud context. The solution thus



enables sophisticated enterprise analytics on the vast data silos scattered across multiple clouds while supporting mechanisms to fulfill data synchronization and governance requirements.

Table 1. Core Components of Adaptive Multi-Cloud Lakehouse Architecture

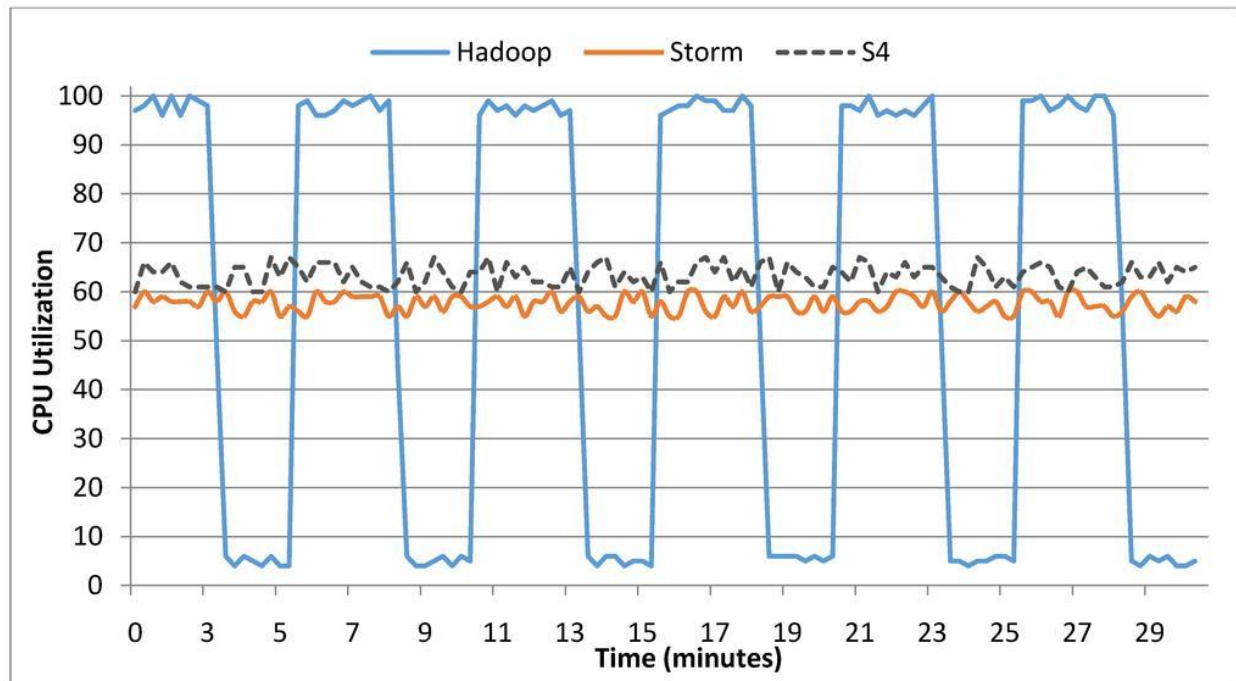
Component	Function	Key Technologies	Benefits
Data Lake	Stores raw and semi-structured data	Delta Lake, Object Storage	Scalability, flexibility
Data Warehouse	Stores curated analytical data	Snowflake, BigQuery, Redshift	Fast analytics
Streaming Layer	Real-time data ingestion and processing	Kafka, Flink, Kinesis	Low-latency analytics
Metadata Catalog	Maintains schemas and lineage	Apache Atlas, Data Catalog	Data discoverability
Governance Layer	Access control and compliance	ABAC, PBAC	Security and compliance
Orchestration Engine	Dynamic resource allocation	Kubernetes, Cloud Controllers	Cost optimization

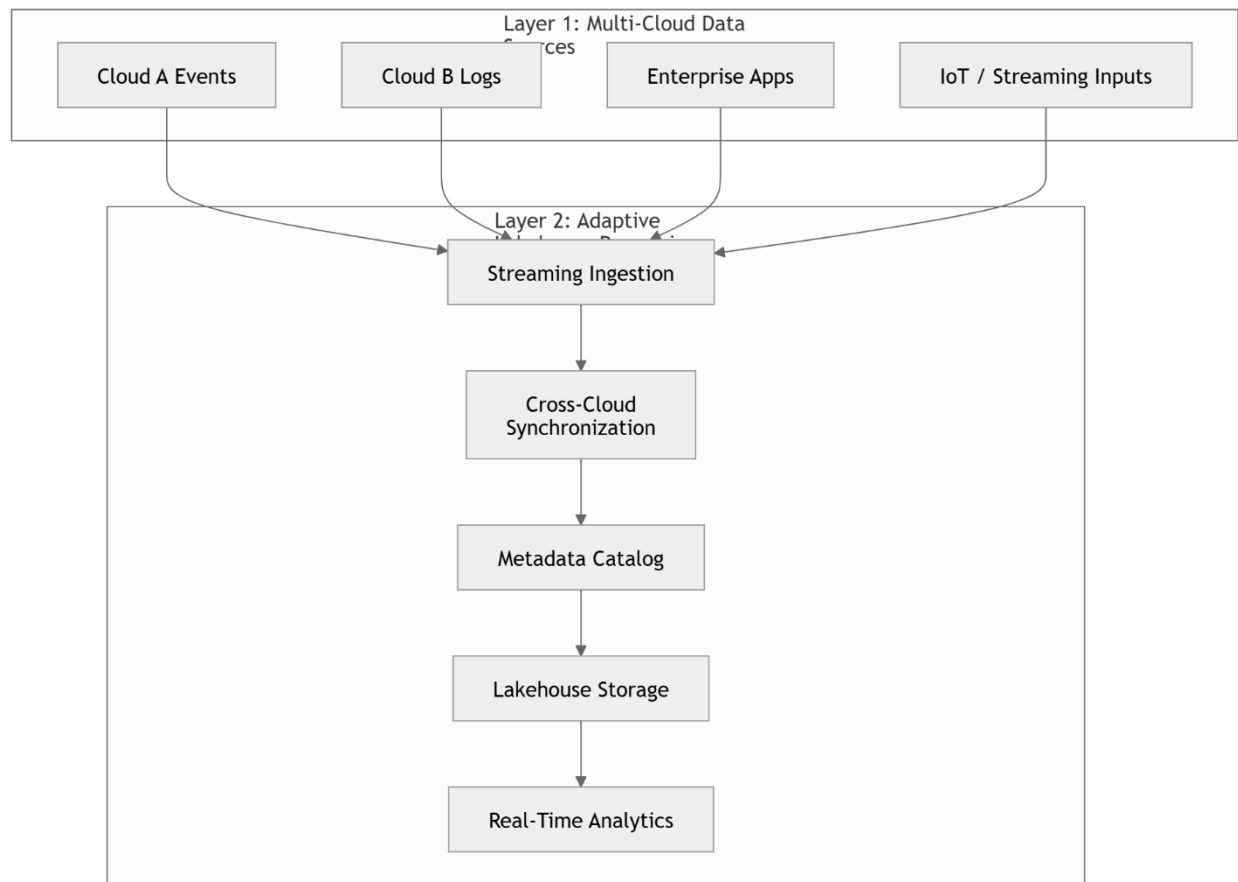
1.1. Cross-Cloud Data Consistency and Synchronization

Managing data consistency among Cloud Service Providers (CSPs) requires a paradigm shift within a traditional Centralizing Lakehouse model into a Synchronizing Lakehouse model that is capable of synchronizing data that is updated across clouds for both storage and analytical purposes. External systems or sources could act as publishers and subscribers when it comes to Lakehouse Synchronization, thereby motivating a consistent data state and synergy across clouds. Support is also needed for practical use cases, such as the ability to integrate multiple analytical environments, or to operate data in a more cost-efficient manner without performance trade-offs.

The set of prepared data synchronization patterns between heterogeneous storage systems, which is leveraged by the Synchronizing Lakehouse model, aims to provide a wider end-to-end data latency guarantee from data production to data consumption in cross-cloud applications. Depending on the internal setups, separation of duties, and cloud-to-on-premises relations of Analytical Service Providers (ASPs), the consistency requirement of the system can be changed by externalizing the responsibilities through these defined patterns. With the Commercial-Off-The-Shelf (COTS) alternatives provided by the Synchronizing Lakehouse model, customers are able to satisfy their predefined consistency requirements without breaking the bank. Such patterns can enrich data even if the agreeability among data in multiple clouds is no longer held. Be it a Centralizing Lakehouse or Synchronizing Lakehouse, both finally lead to move frequent data and take insights immediately.

Data are essential assets for organizations since they enable better and more targeted decision-making. Having the right data at the right time, in the right place, for the right people becomes critical. The ability to provide a common and integrated view over data from different locations with diverse storage technologies has gained considerable attention in recent years from academia, as well as from commercial environments. CSPs now offer highly scalable and easily usable analytical services that are specifically developed to meet the growing requirements of enterprises.





Adaptive Multi-Cloud Lakehouse Architecture

2. Background and Motivation

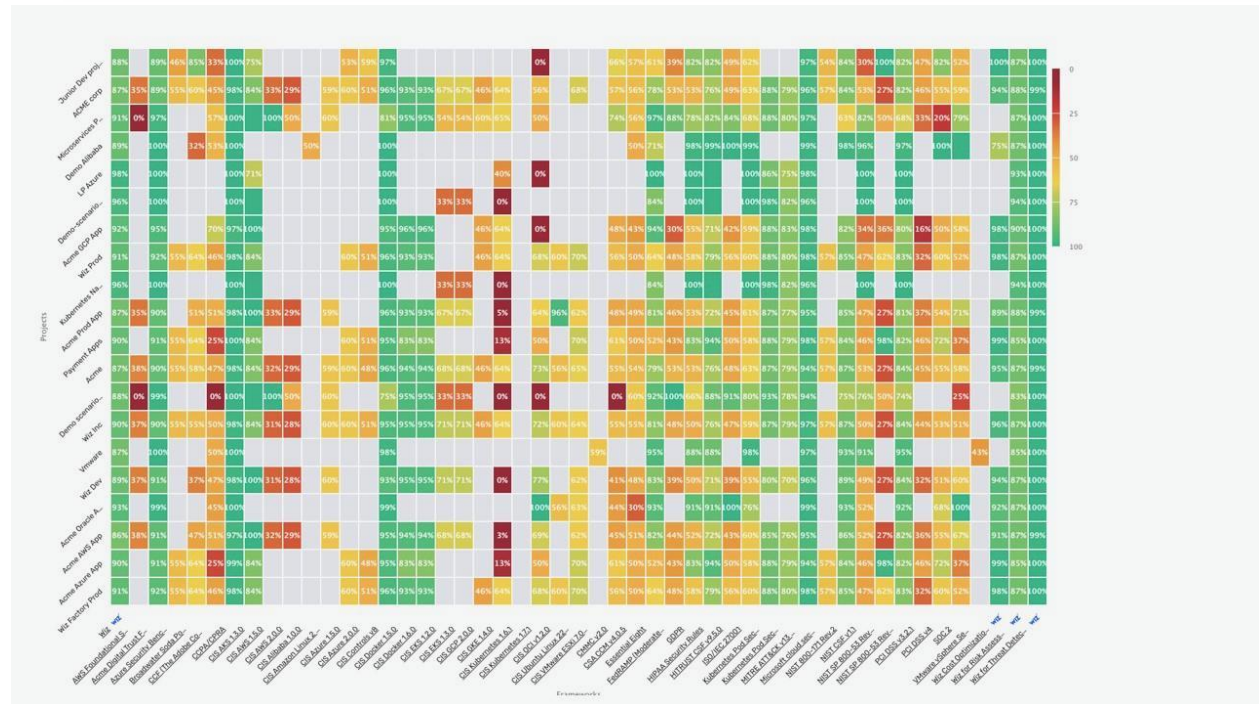
Organizations are actively engaged in the adoption of multi-cloud data ecosystems supported by various data systems to manage operational data. These multi-cloud data ecosystems enable to store, ingest, analyze, and process huge amount of generated data in a cost-efficient manner over cloud infrastructure sourced from multiple cloud service providers. Data solutions residing on heterogeneous infrastructure must guarantee consistent data access across cloud environments. As a result, a wide variety of data integration solutions and approaches have emerged, such as policy-based data interaction frameworks enabling cost-effective data transfer in multi-cloud ecosystems, or a cloud-specific ingestion service to allow transferring data into others cloud storage optimized on costs.



Various data management and analytical architectures, like data warehouses or data lakes, often hosted on different cloud clouds, are combined to allow users to run cross-cloud data analytics. However, policy-based access control across multiple clouds has not been incorporated directly in the data management solutions yet—these policy-based mechanisms consider nevertheless the data access control at the infrastructure layer only. As the availability of low-code or no-code solutions rises, interest in real-time data analytics is on the rise. In this sense, users can create flows to support data processing tasks and derive predefined events with real-time tech like edge computing with the ability to store them in the central enterprise data lake for further analysis. Being cloud agnostic tends to be a strategic disadvantage in real-time enterprise lakehouse networks. Therefore, real-time dynamic ingestion and processing of data using technologies like Apache Flink or Kafka are key requirements for a flow-based enterprise data lakehouse.

Table 2. Multi-Cloud Data Synchronization Strategies

Strategy	Description	Consistency Level	Use Case
Centralized Synchronization	Single master repository	Strong	Financial reporting
Event-Driven Replication	Data propagated via events	Eventual	Enterprise monitoring
Publisher-Subscriber Model	Cross-cloud data exchange	Moderate	Real-time dashboards
Incremental Synchronization	Transfers only changes	Strong	Operational analytics
Hybrid Synchronization	Combines multiple approaches	Configurable	Large enterprises



2.1. Metadata Management and Data Catalogs

Cross-cloud analytics dialects—destructive reads with complex data coordination—need high-stability and available-cloud-oriented execution. Metadata management with data catalogs adapts deployments to such requirements. Towards an all-encompassing and cost-optimized solution for cross-cloud-agnostic operations, a/B data pacification/electrification dilemma catalyzes resource orchestration tuned towards the given execution plan model. Moreover, surface-cache proxies guide queries towards the current cloud acting as a multi-cloud data ecosystem control plane.

Data catalogs abstract technicalities of cloud compatibility, consistency, security, and synchronization across clouds. They provide a network-interface-like view for orchestrating queries, rationalizing data-surface latencies, and managing the cost incurred by cross-cloud data transfers. Furthermore, through a representation of the cross-cloud multi-cloud data ecosystem, they control the resources involved in handling and serving queries. Such resource-usage pattern is akin to the adaptive policies of data pipelines, rationalizing the high operational costs associated with cross-cloud data exchanges. Handling those transfers as a (sliding-window) streaming mini-batch process, surface caches minimize latencies using replica placement similarity that knows that, through a dedicated event-processing service, data joined in a (expected-temporal) big join/merge (or in a sequence of small ones) are almost always reconstructed in a single cloud.

Mathematical Formulas:



Eq. (1) Data Throughput

$$T = \frac{D}{t}$$

Eq. (2) Analytics Latency

$$L = t_p - t_i$$

Eq. (3) Resource Utilization

$$U = \frac{R_u}{R_t}$$

Eq. (4) Scaling Factor

$$S = \frac{R_n}{R_c}$$

Eq. (5) Data Consistency Score

$$C = \frac{D_s}{D_t}$$

Eq. (6) Synchronization Delay

$$D_{sync} = t_r - t_s$$

Eq. (7) Stream Processing Efficiency

$$E = \frac{O}{I}$$

Eq. (8) Query Response Time

$$Q_t = t_e - t_q$$

Eq. (9) Governance Compliance Rate

$$G = \frac{P_c}{P_t}$$



Eq. (10) Access Control Accuracy

$$A = \frac{A_v}{A_t}$$

Eq. (11) Metadata Coverage

$$M = \frac{M_d}{D}$$

Eq. (12) Cloud Cost Efficiency

$$CE = \frac{V}{C}$$

Eq. (13) Event Processing Rate

$$E_r = \frac{N_e}{t}$$

Eq. (14) Workload Prediction Error

$$W_e = |W_a - W_p|$$

Eq. (15) Lakehouse Performance Index

$$LPI = \frac{T \times C}{L}$$

3. Architectural Foundations of Lakehouse Paradigms

The foundational architecture of lakehouse paradigms integrates a data warehouse that also serves as an operational data store (ODS) with a cloud storage service used as a data lake. The ODS is populated through periodic batch ingestion from external systems and sinks of continuous queries over streaming source data ingested into the data lake. The advantage of this architecture stems from the separation of analytical workloads and operational processing. Adaptive lakehouses enhance this architecture by dynamically scaling resources according to demand.

The rationale for integrating data warehouses and data lakes closely aligns with the operational model of multi-cloud data ecosystems. In this scenario, access latency and data consistency requirements are orthogonal dimensions that influence the architecture of real-time analytics services. For relatively low-latency workloads with only moderate data freshness requirements, a data lake is sufficient. Multi-cloud data catalog services enable global schemas to be applied at query time, eliminating the need



for denormalization, while optimization and caching strategies are employed to minimize read-access costs. When data freshness is paramount, analytic workloads are constantly aligned with incoming transactions through continuously running queries that materialize the results in a data warehouse. Consistency is then addressed by ensuring proper synchronization of the two storage layers.

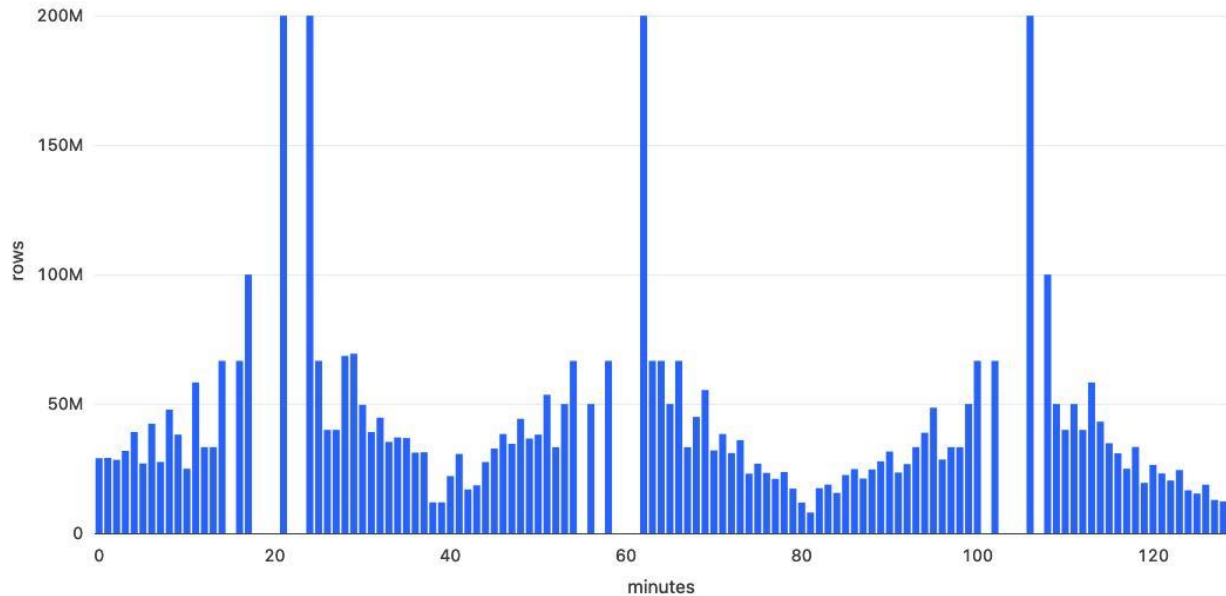
Table 3. Real-Time Analytics Pipeline Layers

Layer	Input	Processing Activity	Output
Data Sources	Applications, IoT, Logs	Event generation	Raw events
Ingestion Layer	Raw events	Stream collection	Event streams
Processing Layer	Event streams	ETL, enrichment	Processed data
Storage Layer	Processed data	Persistence	Lakehouse datasets
Analytics Layer	Curated datasets	BI, ML, AI analysis	Insights
Visualization Layer	Analytical outputs	Dashboard rendering	Business reports

3.1. Data Warehouse and Data Lake Integration

Proposed lakehouse architectures provide integrated environments for business intelligence, analytics, and machine learning while enabling transactional workloads on data stored in cloud object storage. Although merging data warehouse and data lake capabilities into a shared system helps eliminate data silos, it does not resolve data consistency and synchronization issues in multi-cloud ecosystems. Accordingly, several research initiatives focus on providing support for data consistency and synchronization across cloud data ecosystems.

Despite being seen as a logical evolution of the data lake and its associated paradigms, the lakehouse is not as adoption-ready as the latter. First, many cloud vendors may offer generic data processing and computation platforms and services but are not yet positioned with cloud-agnostic services. Second, standardized interfaces for data ingestion, metadata storage, and data processing across vendors are still needed. Consequently, although a cloud data ecosystem based on the managed services of multiple cloud vendors is possible, the establishment of a refined or semi-structured evolution of the data lake is a more practical immediate step. With the broader availability of cloud-agnostic services, a shift toward an integrated lakehouse paradigm across open, cloud-agnostic data ecosystems should occur.



4. Real-Time Analytics in Multi-Cloud Environments

Real-time enterprise analytics necessitate the development of specific ingestion and processing mechanisms capable of handling continuous data streams. Unlike traditional warehouses, which usually accommodate periodic batch updates from data lakes, streaming ingestion frameworks should be built into analytic architectures, together with support for continuous processing. Thus, these frameworks should minimize, and ideally eliminate, the latency between the arrival of data at a source system and its availability to analytic services, accounting for the capture and processing of data at sources both within and external to an enterprise. To achieve low-latency ingestion and processing, these components must operate in a streaming mode, relying on one or more internal or external execution engines.

One of the most established open-source solutions for streaming ingestion and processing is Apache Kafka, an ecosystem comprising multiple components (e.g., Kafka, Kafka Connect and Kafka Streams) supporting the ingestion and storage of continuous streams of data produced by a wide variety of sources. Data produced by Kafka Connect can be ingested directly into an enterprise lakehouse or processed at an interim stage by Kafka Streams to address analytic and operational workloads that require quality-of-service guarantees. Kafka and its associated components can be deployed locally or in external cloud environments, possibly even in multiple locations simultaneously.

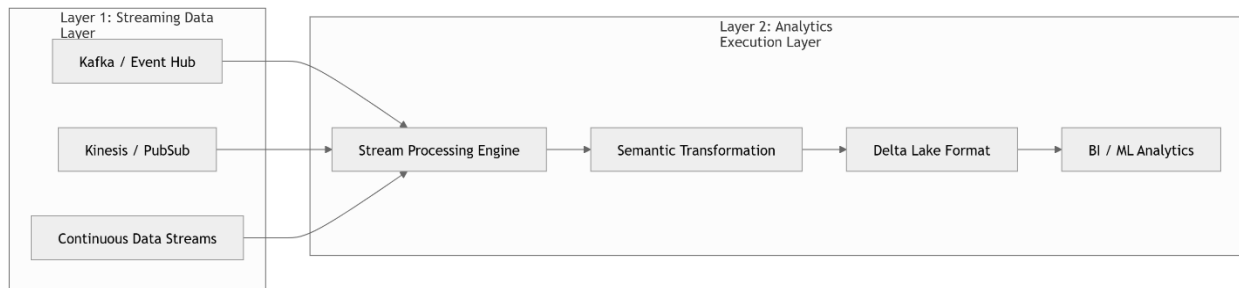


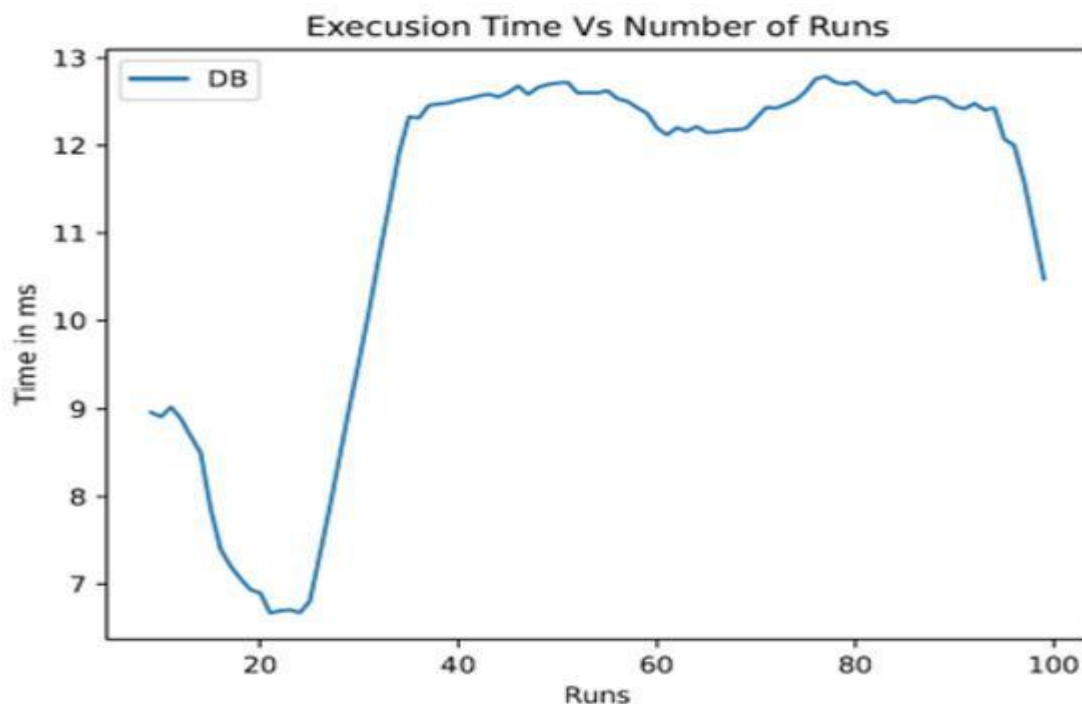
Table 4. Cloud-Native Streaming Technologies Comparison

Technology	Primary Purpose	Latency	Scalability	Multi-Cloud Support
Apache Kafka	Event streaming	Low	High	High
Apache Flink	Stream processing	Very Low	High	High
Amazon Kinesis	Managed streaming	Low	High	Medium
Azure Event Hub	Event ingestion	Low	High	Medium
Google Pub/Sub	Messaging service	Low	High	Medium

4.1. Streaming Ingestion and Processing

Real-time analytics capability requires multilayered stream ingestion and processing pipelines that ensure correct and performant ETL flows. The streaming ingestion layer at the bottom utilizes cloud-native managed **streaming** services such as Azure Event Hub, Amazon Kinesis Data Streams, and Google Cloud Pub/Sub. Streamed data often has nonconsumable sources such as GPS devices or brain-computer interface EEG sensor that these services absorb before finally loading back to cloud services that the upper layer—cloud-native managed functions responding to ingestion event messages—transforms in-place process to Delta Lake format. Semantic data transformation is achieved by streaming processing engines, which take incoming data and continuously apply rules onto them, delaying the outbound event message until a data-event congruency is met. Such processing enables complex data queries, externaling data into multi-cloud data ecosystems into producer subcomponents before loading back into consumer subcomponents. Dynamic Stream Computation, which uses a differential, redundant-computation-free technique and checks data events through data-use queries before publication, allows the Delta Lake to have fresh data and support real-time analysis. Real-time analytics in adaptive lakehouse environments relies on multilayered stream ingestion and processing pipelines that maintain efficient and accurate ETL workflows. Cloud-native streaming services such as Azure Event Hubs, Amazon Kinesis Data Streams, and Google Cloud Pub/Sub ingest high-velocity data from heterogeneous sources, including IoT sensors and specialized devices such as EEG systems. Event-driven serverless functions transform these streams into Delta Lake formats, while streaming

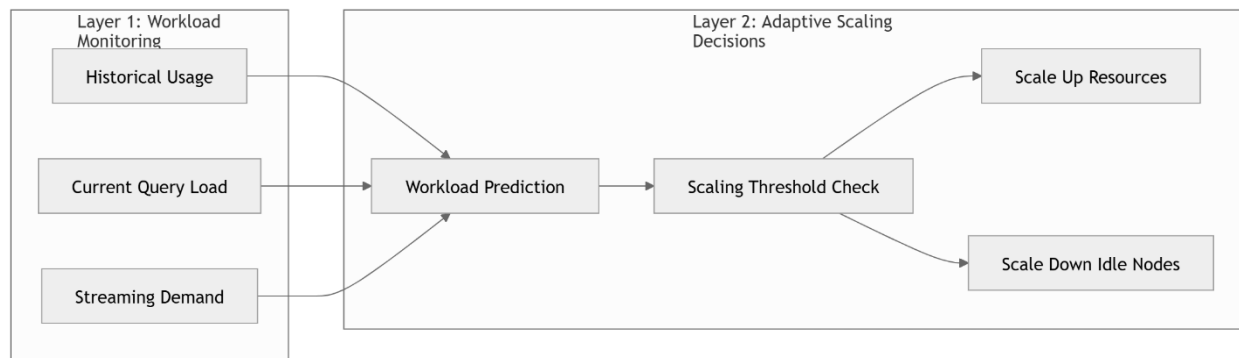
processing engines continuously apply semantic validation and business rules to ensure data consistency before publication. Dynamic stream computation further minimizes redundant processing by evaluating data events against active query requirements, enabling fresh, synchronized datasets that support scalable real-time analytics across multi-cloud producer and consumer ecosystems.



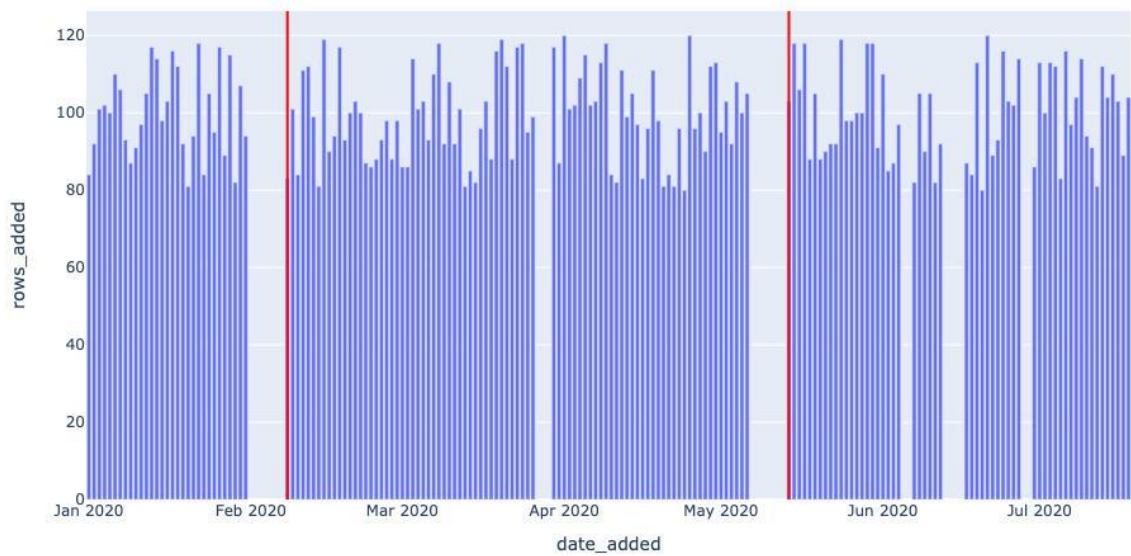
5. Adaptive Lakehouse Mechanisms

Adaptive Lakehouse Mechanisms are designed to simplify the overall architecture and minimize overhead while fulfilling multiple, often conflicting, stakeholder requirements of multi-cloud enterprise data ecosystems. Such requirements are fulfilled by dynamic resource orchestration, which helps accommodate varying resource needs at different times.

The nature and scale of resource consumption in data-processing workloads exhibit considerable variations over time, matching to a larger extent with those of the resource provisioning demand in public clouds compared to on-premises infrastructures. A cross-cloud lakehouse can exploit this asymmetry by shifting forward-demand resource provisioning—and the associated cost—to a cost-subsidizing, on-premises infrastructure, while employing cloud resources only in the presence of peak forward-donation demands that exceed the on-premises infrastructure’s processing capacity. Providing this datacenter elasticity requires additional provisioning logic and automation to orchestrate the provisioning of read-only copies of the data warehouse and data lake in the respective cloud datacenters and associated streaming data-replication hetero-ERO overheads.



Dynamic Resource Orchestration



5.1. Dynamic Resource Orchestration

Real-time analytics, across-cloud data exchange and high resource demands (for on-the-fly data preparation and transformation) contribute to the need for resources that can dynamically scale up/down via an automated orchestration mechanism. Such a mechanism should focus on the resources underlying the more frequently used lakehouses, rather than redundantly configuring all three infrastructures of the multi-cloud data ecosystem. Key enablers of workload prediction within the considered ecosystem are



machine learning models based on historical usage. The projected future workload within each cloud should inform the scaling policies (i.e., scaling thresholds) for the resources of the corresponding lakehouses. In particular, when the load is expected to exceed a defined threshold, more resources are allocated; when a given idle threshold is reached, the resources are removed. Furthermore, due to operator costs, the increase and decrease of resources should occur on multiple nodes (chunks) to improve prediction accuracy and save time (i.e., allow more time between rescaling – shrinking or expanding the resources).

Table 5. Dynamic Resource Orchestration Policies

Policy Type	Trigger Condition	Action Taken	Expected Outcome
Scale-Up Policy	CPU > 80%	Add compute nodes	Higher throughput
Scale-Down Policy	CPU < 30%	Remove idle nodes	Cost reduction
Predictive Scaling	Forecasted workload spike	Provision resources	Reduced latency
Data Locality Policy	Cross-cloud transfer detected	Move computation closer	Lower transfer cost
Workload Balancing	Uneven utilization	Redistribute jobs	Improved efficiency

6. Data Governance in Multi-Cloud Lakehouses

Data governance is a key concern of every enterprise using data analytics, especially in a multi-cloud environment. To reduce the risk of compartmentalized insights and even more the risk of retaining incorrect data, a process for detecting and removing data that is obsolete or incorrect must be established. Cloud services provide various kinds of indicators of data quality for this purpose, which can be retained in a separate cloud database that stores the status table and triggers the removal process. Besides the price for data storage, it is also important to restrict user access to sensitive data. A flexible policy for managing access to data is essential, especially in a multi-cloud environment where data governance and risk management are critical for security and compliance with regulations, such as the GDPR.

A policy model for access control can be created by combining the Attribute-Based Access Control model with the Policy-Based Access Control model in a hierarchical structure. The first level can define the authorization for a group of users with specific roles, and the next layer can further limit specific administrators by location, date and time. This model also allows the definition of external data access policies, according to the conditions established by the cloud services. For example, a cloud service can allow external access to well-established databases throughout the year with no restrictions, and data only during business hours with a special authorization. This model allows to protect sensitive information even when data is being shared between multiple cloud services, since the encryption key can also be defined according to the access policy.

Table 6. Metadata Management Functions



Function	Description	Business Impact
Schema Registry	Maintains data structures	Consistent analytics
Data Lineage	Tracks data movement	Auditability
Data Cataloging	Organizes datasets	Faster discovery
Quality Monitoring	Detects anomalies	Improved trust
Policy Metadata	Stores governance rules	Compliance assurance

7. Conclusion

The field of real-time enterprise analytics has developed from the convergence of cloud data warehouses and data lakes into a single architecture, the lakehouse. A lakehouse tightly integrates these underlying components, thereby supporting the analytics capabilities of both paradigms. Additionally, dynamic systems resource allocation is a first-order requirement on enterprise architectures, especially in the context of the tight coupling between cross-cloud data-consistency and data-analytics requirements.

Various construction and operation patterns address these issues in the context of enterprises that span multiple clouds. The mechanisms can evolve using analytics to inform historical decisions and support remediation of data-and-analytical sensors in other clouds. Enterprise-wide data-governance requirements necessitate extending the enterprise metadata-management system with a policy-based access-control facility.

References

1. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. CIDR.
2. Sawadogo, P. N., & Darmont, J. (2021). On data lake architectures and metadata management. *Journal of Intelligent Information Systems*, 56(1), 97–120.
3. Reddy, R. (2023). Predictive Health Insights: Ai And Ml's Frontier In Disease Prevention And Patient Management. Available at SSRN.
4. Begoli, E., Goethert, I., & Knight, K. (2021). A lakehouse architecture for the management and analysis of heterogeneous data for biomedical research and mega-biobanks. In *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data 2021)* (pp. 4643–4651).
5. Paruchuri, J. K. (2021). Lakehouse architecture: Unifying data lakes and data warehouses. *International Journal of Computer Techniques*, 8(1).
6. Kaur, K., Garg, S., & Kaur, K. (2021). Multi-cloud adoption challenges for the cloud-native era. *International Journal of Cloud Applications and Computing*, 11(2).



7. Anwarbasha, H., Sasi Kumar, S., & Dhanasekaran, D. (2021). An efficient and secure protocol for checking remote data integrity in multi-cloud environment. *Scientific Reports*, 11, 13755.
8. Srinivas Kalisetty. (2023). Big Data–Driven Cloud Collaboration Models for Enhancing Supplier–Retailer Synchronization in Modern Manufacturing Supply Chains. *Journal of Computational Analysis and Applications (JoCAAA)*, 31(4), 2188–2205. Retrieved from <https://eudoxuspress.com/index.php/pub/article/view/4232>
9. Rau, A., & Suri, P. K. (2021). Towards securing cloud data in the multi-cloud scenario. *Periodicals of Engineering and Natural Sciences*, 9(2), 868–872.
10. Bice, V., & Hesse, W. (2021). Modeling metadata in data lakes—A generic model. *Data & Knowledge Engineering*, 136, 101931.
11. Machado, I. A., Costa, C., & Santos, M. Y. (2022). Data mesh: Concepts and principles of a paradigm shift in data architectures. *Procedia Computer Science*, 196, 263–271.
12. Pamisetty, V. (2023). Transforming Community Engagement with Generative AI: Harnessing Machine Learning and Neural Networks for Hunger Alleviation and Global Food Security. *Journal for Re Attach Therapy and Developmental Diversities*.
13. Wieder, P., & Nolte, H. (2022). Toward data lakes as central building blocks for data management and analysis. *Frontiers in Big Data*, 5, 945720.
14. Zichichi, M., Ferretti, S., D'Angelo, G., & Rodríguez-Doncel, V. (2022). Data governance through a multi-DLT architecture in view of the GDPR. *Cluster Computing*, 25(6), 4515–4542.
15. Nadal, S., Jovanovic, P., Bilalli, B., & Romero, O. (2022). Operationalizing and automating data governance. *Journal of Big Data*, 9, 117.
16. Armenatzoglou, N., Basu, S., Bhanoori, N., Cai, M., Chainani, N., Chinta, K., Govindaraju, V., Green, T. J., Gupta, M., Hilling, S., Hotinger, E., Leshinsky, Y., Liang, J., McCreedy, M., Nagel, F., Pandis, I., Parchas, P., Pathak, R., Polychroniou, O., Rahman, F., Saxena, G., Soundararajan, G., Subramanian, S., & Terry, D. (2022). Amazon Redshift re-invented. In *Proceedings of the 2022 International Conference on Management of Data* (pp. 2205–2217).
17. Behm, A., Palkar, S., Agarwal, U., Armstrong, T., Cashman, D., Dave, A., Greenstein, T., Hovsepian, S., Johnson, R., Krishnan, A. S., Leventis, P., Luszczak, A., Menon, P., Mokhtar, M., Pang, G., Paranjpye, S., Rahn, G., Samwel, B., van Bussel, T., van Hovell, H., Xue, M., Xin, R., & Zaharia, M. (2022). Photon: A fast query engine for lakehouse systems. In *Proceedings of the 2022 International Conference on Management of Data* (pp. 2326–2339).
18. Meda, R., & Pamisetty, A. (2023). Intelligent Infrastructure for Real-Time Inventory and Logistics in Retail Supply Chains. *Educational Administration: Theory and Practice*, 29 (4), 5215–5233.
19. Tosun, E., Machado, I. A., Costa, C., & Santos, M. Y. (2022). Data mesh: Concepts and principles of a paradigm shift in data architectures. *Procedia Computer Science*, 196, 263–271.
20. Shahane, R. (2022). Enhancing data governance in multi-cloud environments: A focused evaluation of Microsoft Azure's capabilities and integration strategies. *Journal of Computational Analysis and Applications*, 30(2), 536–548.
21. Arul, K. (2022). Data engineering challenges in multi-cloud environments: Strategies for efficient big data integration and analytics. *International Journal of Scientific Research and Management*, 10(6).
22. Dubey, S., Diwivedi, P., Shukla, A., Kushwaha, R., Shukla, A., & Gupta, S. (2022). A review on data storage in multi-cloud environment for privacy preserving. *International Journal of Food and Nutritional Sciences*, 11(Special Issue 6), 458–464.
23. Pandugula, C., & Nampalli, R. C. R. Optimizing Retail Performance: Cloud-Enabled Big Data Strategies for Enhanced Consumer Insights.



24. Liu, Y., Zhang, H., & Wang, J. (2022). Observations and expectations on recent developments of data lakes. *Procedia Computer Science*, 214, 405–411.
25. Sun, Y., Meehan, T., Xie, W., Basmanova, M., Erling, O., Zhong, R., Wang, K., Agarwal, S., Gupta, A., Logothetis, D., & others. (2023). Presto: A decade of SQL analytics at Meta. *Proceedings of the VLDB Endowment*.
26. Ait Errami, S., Hajji, H., Ait El Kadi, K., & Badir, H. (2023). Spatial big data architecture: From data warehouses and data lakes to the LakeHouse. *Journal of Parallel and Distributed Computing*, 176, 70–79.
27. Goedegebuure, A., Kumara, I., Driessen, S., Monsieur, G., Di Nucci, D., van den Heuvel, W.-J., & Tamburri, D. A. (2023). Data mesh: A systematic gray literature review. *ACM Computing Surveys*.
28. Mazumdar, D., Hughes, J., & Onofre, J. B. (2023). The data lakehouse: Data warehousing and more. *arXiv*.
29. Thamma, S. R. (2023). Lakehouse architecture: Bridging on-prem big data and modern data platforms. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(4).
30. Soon, K. Y. (2023). Data lakehouse architecture for self-service data analytics. *Universiti Malaysia Pahang Al-Sultan Abdullah*.