COVENANT AEGIS REVIEW OF FINANCIAL TECHNOLOGY VOL. 3 · NO. 2 · 2026 · PP. 4–23

Operational Big Data in Finance: Streaming Architectures, Lakehouses and Real-Time Decisioning

P. Vasquez[1], E. Rossi[1], F. Adeyemi[2], H. Said[3]

[1]*Covenant Aegis Finance Division, Data Platform Laboratory*

[2]*Department of Financial Data Systems, Partner Research Network*

[3]*Center for Real-Time Analytics*

Received 16 Jan 2026 · Revised 3 Mar 2026 · Accepted 11 Mar 2026 · REVIEW ARTICLE · DOI 10.48-CA/fintech.2026.0321 ·
Corresponding author: p.vasquez@covenantaegis.com

ABSTRACT

Background. Financial institutions operate on high-velocity, high-volume data that must inform decisions in milliseconds while remaining governable and auditable. **Objective.** To review the architectural evolution enabling operational big data—distributed batch, in-memory and stream engines, the log-centric Kappa pattern and the lakehouse—and the governance demands specific to regulated finance. **Methods.** Narrative review of the large-scale data-systems literature and real-time-decisioning practice. **Results.** Durable logs and stream processing with exactly-once semantics enable sub-second decisioning (e.g., fraud scoring) at scale; lakehouses unify data-lake economics with warehouse reliability; class imbalance and concept drift dominate fraud-model design. **Conclusion.** Real-time decisioning at scale is feasible, but only disciplined architecture and governance—durable logs, exactly-once processing, governed lakehouse—make decisions trustworthy in regulated finance.

Keywords: big data; stream processing; lakehouse; real-time decisioning; fraud detection; exactly-once; data governance

1. Introduction

Finance is data-intensive, and latency, accuracy and auditability all carry direct economic and regulatory weight. Detecting a fraudulent payment, pricing risk or honouring a market obligation may require decisions within milliseconds, on data arriving continuously and at scale [1].

2. Architectural Evolution

Distributed batch processing via MapReduce made petabyte-scale computation tractable [2], and in-memory engines such as Apache Spark accelerated iterative and interactive analytics [3]. Stream-processing systems then enabled low-latency computation over unbounded data; Apache Kafka established the distributed log as a durable backbone [4]. The Lambda architecture combined batch and speed layers, while the simpler Kappa architecture processes everything as a stream, reducing duplicated logic [5] (Figure 1).

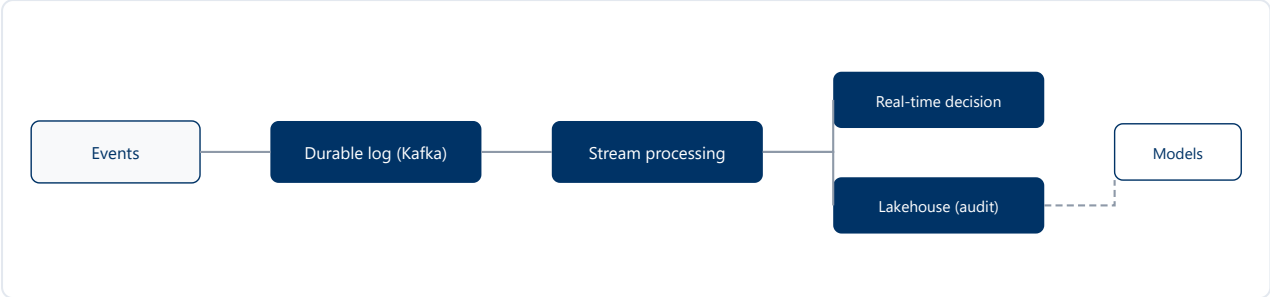


Figure 1. Kappa-style architecture: a durable log feeds stream processing that drives real-time decisions and persists governed tables in a lakehouse; the same features serve model training [4,5]. Schematic.

3. The Lakehouse

Historically, data lakes offered cheap, flexible storage but weak guarantees, while warehouses offered reliability at higher cost. The lakehouse adds transactional tables, schema enforcement and governance over low-cost storage, unifying analytics and machine learning—an attractive fit for finance's mix of regulatory reporting and real-time analytics [6].

Table 1. Storage paradigms and fit for regulated finance.

Paradigm	Strength	Weakness
Data warehouse	Reliability, SQL	Cost, rigidity
Data lake	Cheap, flexible	Weak guarantees
Lakehouse	Both + transactions	Operational maturity

4. Real-Time Decisioning

Operational analytics turns these platforms into decision engines. Fraud detection scores transactions in flight against behavioural baselines, balancing detection against false declines; reviews document the methods and the central challenge of class imbalance and concept drift [7]. Feature stores supply consistent features to both training and low-latency inference, avoiding training-serving skew.

5. Governance for Regulated Finance

Throughput is insufficient without control. Data lineage, quality monitoring, access control and reproducibility are mandatory, as is consistency: exactly-once processing prevents double-counting of financial events. Streaming and lakehouse platforms now provide transactional and exactly-once guarantees that make them defensible for financial records [4,6].

6. The Covenant Aegis Platform

Our platform ingests payment and market events through a durable log, computes features and risk scores in-stream with exactly-once semantics, and persists governed, transactional tables in a lakehouse for analytics and audit. Lineage and quality controls are built in, and the same features serve real-time decisioning and model training.

7. Discussion and Conclusion

The architecture's value is realised only with governance; performance without

lineage and exactly-once semantics is indefensible in regulated finance. Figures and tables here are illustrative. Operational big data lets institutions decide in real time at scale, but disciplined architecture makes those decisions trustworthy.

References

1. Provost F, Fawcett T. Data science and its relationship to big data and data-driven decision making. *Big Data*. 2013;1(1):51–59.
2. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. *Commun ACM*. 2008;51(1):107–113.
3. Zaharia M, et al. Apache Spark: a unified engine for big data processing. *Commun ACM*. 2016;59(11):56–65.
4. Kreps J, Narkhede N, Rao J. Kafka: a distributed messaging system for log processing. *NetDB*. 2011.
5. Marz N, Warren J. *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems*. Manning; 2015.
6. Armbrust M, et al. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. *CIDR*. 2021.
7. Abdallah A, Maarof MA, Zainal A. Fraud detection system: a survey. *J Netw Comput Appl*. 2016;68:90–113.

9. Distributed Storage and Processing Foundations

Operational big data rests on distributed storage and compute. The Google File System and MapReduce established fault-tolerant storage and batch processing over commodity clusters [A1]; Hadoop popularised the model; and Spark's resilient distributed datasets enabled fast iterative and interactive computation in memory [A2]. These foundations make petabyte-scale analytics tractable, but batch latency motivated the stream-processing systems that enable real-time decisioning.

Table A1. Processing paradigms and latency profile.

Paradigm	Latency	Use
Batch (MapReduce)	Minutes–hours	Reporting, ETL
In-memory (Spark)	Seconds–minutes	Iterative analytics
Stream (Kafka/Flink)	Milliseconds	Real-time decisions

10. Stream Processing and Exactly-Once Semantics

Stream processors compute over unbounded event streams with state, windowing and event-time handling. The distributed log (Kafka) provides durable, ordered, replayable event storage that decouples producers and consumers [A3]. Correctness in finance demands exactly-once processing—each event affects state once despite failures and retries—achieved via idempotent writes, transactional sinks and checkpointed state. Without it, replays double-count financial events.

11. The Lakehouse and Transactional Tables

The lakehouse adds ACID transactions, schema enforcement, time travel and governance to low-cost object storage via open table formats, unifying batch and streaming on one platform [A4]. This reconciles the historical tension between cheap, flexible lakes and reliable, governed warehouses—well suited to finance's mix of regulatory reporting and real-time

analytics. Figure A1 contrasts decision latency across architectural generations.

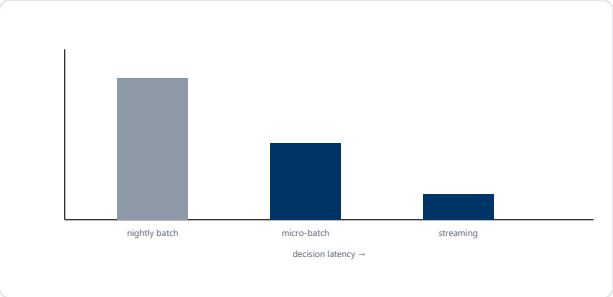


Figure A1. Schematic decision-latency reduction from nightly batch to micro-batch to true streaming, enabling in-flight decisions [A2,A3]. Conceptual.

12. Real-Time Fraud Detection

Card and payment fraud detection is the archetypal operational use: transactions are scored in milliseconds against behavioural baselines, balancing fraud capture against false declines that harm customers. The defining challenges are extreme class imbalance, adversarial concept drift as fraud tactics change, and the need for sub-100-ms latency [A5]. Feature stores serve identical features to training and inference, preventing training-serving skew, and models are retrained frequently against drift.

13. Data Governance and Lineage

In regulated finance, governance is inseparable from performance. Lineage traces every figure to its sources and transformations; data-quality monitoring, access control, retention and privacy controls are mandatory; and reproducibility supports audit and model risk management (e.g., SR 11-7-style model governance) [A6]. The platform must answer, for any decision, what data and code produced it.

14. Reliability and Cost Engineering

Operational platforms require high availability, disaster recovery and predictable cost. Techniques include multi-zone replication, backpressure handling, schema evolution without downtime, tiered storage and workload isolation. Capacity and cost are engineered explicitly, since unbounded retention and reprocessing can dominate spend at financial-data scale.

15. Future Directions

Streaming-first lakehouses, unified batch/stream semantics, real-time feature platforms, and privacy-preserving analytics (tokenisation, differential privacy, secure enclaves) are advancing. The objective remains millisecond decisions at scale that are simultaneously governed, auditable and reproducible.

Extended References

8. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. *Commun ACM*. 2008;51(1):107–113.
9. Zaharia M, et al. Apache Spark: a unified engine for big data processing. *Commun ACM*. 2016;59(11):56–65.
10. Kreps J, Narkhede N, Rao J. Kafka: a distributed messaging system for log processing. *NetDB*. 2011.
11. Armbrust M, et al. Lakehouse: a new generation of open platforms. *CIDR*. 2021.
12. Abdallah A, Maarof MA, Zainal A. Fraud detection system: a survey. *J Netw Comput Appl*. 2016;68:90–113.
13. Board of Governors of the Federal Reserve System. SR 11-7: Guidance on Model Risk Management. 2011.

Author contributions: PV led the review; ER and FA contributed finance and governance; HS reviewed streaming systems. All authors revised. **Funding:** Covenant Aegis Finance Division. **Competing interests:** Authors are affiliated with Covenant Aegis.

© 2026 Covenant Aegis Finance Division. This review synthesises publicly available peer-reviewed literature; figures are illustrative. Nothing herein is financial advice. For professional and educational use.