

Scalable AI Infrastructure for Large-Scale Financial Data Processing Using Cloud Computing

George Emmanuel

Independent Researcher

Midwestern, United States

Email ID: George_emmohio3102@yahoo.com

DOI: <https://doi.org/10.66592/jcstm.01.04.53>

Abstract

Financial institutions process data at an extraordinary scale and speed. Market data arrives in millions of messages per second, payments must be scored for fraud in milliseconds, and regulatory stress tests require vast computations within fixed deadlines. Artificial intelligence (AI) adds further demands: training models on years of high-frequency data, serving predictions with strict latency, and processing unstructured documents. On-premises infrastructure sized for peak load is costly and slow to change. Cloud computing offers elasticity, but using it well for regulated financial AI requires deliberate architecture. This article proposes a reference infrastructure for scalable AI-driven financial data processing. It first characterizes six financial workload types by latency, volume, and computing profile. It then sets out a layered cloud infrastructure: workload placement across deployment models, container orchestration, lakehouse storage, stream processing, distributed training, and model serving. Operations are handled by bounded autonomous agents working through configurable change workflows, with resilience requirements drawn from financial regulation. Using a design-oriented approach grounded in ten studies published between 2016 and 2025, the article illustrates the infrastructure with a bank running real-time fraud scoring, daily market risk, and a quarterly stress test. It argues that scalable financial AI depends on matching infrastructure patterns to workload characteristics and on operating that infrastructure autonomously within governed limits.

Keywords: cloud computing, AI infrastructure, financial data processing, scalability, container orchestration, lakehouse, stream processing, distributed training, model serving, autonomous operations, operational resilience

1. Introduction

Finance has always been a data business, but the scale has changed. Equity markets produce order book updates in the billions per day. Card networks and instant payment systems handle thousands of transactions per second, each needing a fraud decision before authorization. Risk departments run simulations across millions of positions and thousands of scenarios. Compliance teams process large volumes of documents and communications. AI now touches almost all of these activities.

These workloads differ enormously. Some need responses in milliseconds, others must finish overnight, and some run only a few times a year but then need enormous capacity. Traditional on-premises infrastructure handles this variety poorly. It must be sized for the largest peak, so it sits partly idle most of the time and still takes months to expand. Cloud computing changes this equation by providing capacity on demand. Sannidhanam (2020) presents a comparative analysis of cloud computing architectures for enterprise applications, showing that architectures differ in scalability, performance, cost, security, and manageability, and that the right choice depends on the workload.

Scaling infrastructure for financial AI raises further challenges. Large clusters of services must be orchestrated reliably. Storage must serve both regulatory reporting and ML from consistent data. Streams must be processed with exactly-once guarantees, because counting a payment twice is not acceptable. Models must be trained across many machines and served with predictable latency. The infrastructure must be operated around the clock, increasingly by autonomous agents (Sannidhanam, 2025), under change controls that satisfy auditors (Basireddy, 2022). All of this must meet regulatory expectations for operational resilience, including the management of cloud provider risk (European Parliament & Council of the European Union, 2022).

This article asks: How should cloud infrastructure be designed and operated to process large-scale financial data for AI workloads in a scalable, efficient, and resilient way? It contributes (1) a characterization of financial AI workloads, (2) a layered reference infrastructure, (3) operating principles for autonomous, governed infrastructure management, and (4) an illustrative application.

2. Related Work

2.1 Cloud architectures and workload placement

Sannidhanam (2020) compares cloud architectures for enterprise applications and shows that none is best for every workload. Public cloud offers elasticity and managed services. Private environments offer control. Hybrid combinations balance the two at the cost of integration complexity. For financial institutions, where data sensitivity and regulation vary by workload, this supports placing each workload deliberately rather than moving everything to one environment.

2.2 Container orchestration

Burns et al. (2016) describe how Google's Borg and Omega cluster managers led to Kubernetes, an open system for scheduling containers across machines, restarting failed workloads, and scaling services. Kubernetes has become the common foundation for running data processing, model training, and model serving on shared cloud clusters. It gives financial institutions one control plane across public and private environments.

2.3 Lakehouse storage

Armbrust et al. (2021) propose the lakehouse, which adds transactions, schema enforcement, versioning, and performance optimizations to open file formats in low-cost cloud object storage. For financial institutions, this means regulatory reporting, analytics, and ML can use the same governed copy of data. Versioning also allows a report or model to be reproduced from the exact data used at the time, which auditors and model validators require.

2.4 Stream processing

Armbrust et al. (2018) describe Structured Streaming in Apache Spark, which expresses streaming computations through the same high-level interface as batch queries. It updates results incrementally and provides end-to-end exactly-once guarantees. For finance, exactly-once processing prevents duplicated or lost transactions in real-time aggregates. Using the same interface for batch and streaming reduces the risk of inconsistent logic between real-time and end-of-day calculations.

2.5 Distributed training on financial data

Financial models increasingly learn from very large datasets. Sirignano and Cont (2019) trained

deep learning models on billions of high-frequency order book records across a large universe of stocks. They found universal features of price formation, meaning a single model trained across all stocks outperformed models trained on individual stocks. Research at this scale needs distributed training. Sergeev and Del Balso (2018) introduced Horovod, which uses ring-allreduce communication to average gradients efficiently across many GPUs. It lets training code written for one machine scale to large clusters with few changes.

2.6 Model serving

Olston et al. (2017) describe TensorFlow Serving, a system for serving ML models in production with high performance. It supports multiple model versions and can switch between them without interruption. Serving is a distinct infrastructure concern with its own scaling needs. In finance, where fraud models may be updated frequently and must never stop scoring, version management and zero-interruption updates are essential.

2.7 Autonomous infrastructure operations

Sannidhanam (2025) examines autonomous AI agents for cloud infrastructure operations: agents that monitor infrastructure, diagnose issues, and carry out corrective actions such as scaling, restarting, or rerouting within defined limits and with human oversight. Large financial AI infrastructure produces more operational signals than human teams can follow, so such agents can greatly improve response times. Their authority must be carefully bounded.

2.8 Governed change

Basireddy (2022) proposes configurable enterprise workflow architecture for digitizing risk management, in which identification, assessment, approval, escalation, and remediation are defined as configurable workflows with audit trails. Infrastructure changes in regulated institutions are subject to change management controls. Configurable workflows provide a way to route changes proposed by agents or engineers through risk-based approvals, and to record them.

2.9 Operational resilience regulation

The European Union's Digital Operational Resilience Act (European Parliament & Council of the European Union, 2022) sets requirements for financial entities on ICT risk management,

incident reporting, resilience testing, and management of third-party ICT providers, including cloud providers. It includes expectations for exit strategies and concentration risk. It makes infrastructure resilience a regulatory obligation, not only an engineering goal.

2.10 Research gap

The studies above cover architecture selection, orchestration, storage, streaming, training, serving, autonomous operations, governed change, and resilience regulation. There is little integrated guidance on how these fit together as infrastructure for financial AI workloads specifically. This article provides that integration.

3. Research Methodology

The study follows a design-oriented research approach. The problem, the difficulty of providing infrastructure for highly varied, regulated financial AI workloads, was identified from the literature and from the characteristics of financial data processing. Requirements were derived by characterizing the main workload types (Section 4). A layered reference infrastructure and a set of operating principles were designed from the ten studies reviewed and demonstrated through an illustrative bank case. The demonstration shows feasibility and internal consistency. Empirical evaluation is left to future work.

4. Characterizing Financial AI Workloads

Infrastructure should follow workload characteristics. Table 1 characterizes six common financial AI workloads.

The table shows why no single infrastructure configuration fits all financial AI workloads. Transaction scoring needs steady, low-latency serving that can absorb peaks. Stress testing needs very large capacity for a short time. Training needs GPU clusters with fast interconnects. The reference infrastructure is therefore designed to support several patterns on a common foundation.

Table 1. Characteristics of financial AI workloads and matching infrastructure patterns

Workload	Latency need	Data volume and pattern	Compute profile	Infrastructure pattern
Real-time transaction scoring	Milliseconds	Continuous, high-rate events	Many small inferences	Autoscaled serving, stream processing
Market data analytics	Sub-second to seconds	Very high-rate streams, bursts at market open and close	Continuous stream computation	Stream processing on dedicated capacity
Daily risk calculation	Hours, fixed overnight window	Large batch of positions and scenarios	Parallel batch computation	Elastic batch clusters
Regulatory stress testing	Days, fixed deadline	Very large, occasional	Massive parallel simulation	Temporary burst capacity
Model training	Hours to days	Years of historical data	GPU-intensive, communication-heavy	Distributed training clusters
Document and text processing	Seconds to hours	Large unstructured corpora	Language model inference	Batch and on-demand inference services

5. Reference Infrastructure

The reference infrastructure has eight layers (Table 2).

Table 2. Reference cloud infrastructure for large-scale financial AI

Layer	Function	Key technologies and practices	Supporting studies
1. Workload placement	Decide where each workload runs	Public, private, and hybrid placement by sensitivity and elasticity	Sannidhanam (2020)
2. Orchestration	Schedule and scale containerized workloads	Kubernetes across environments	Burns et al. (2016)
3. Storage	Hold governed, versioned financial data	Lakehouse on cloud object storage	Armbrust et al. (2021)
4. Stream processing	Process events in real time with exact counts	Structured streaming, exactly-once guarantees	Armbrust et al. (2018)
5. Distributed training	Train models on very large datasets	Data-parallel GPU training	Sergeev & Del Balso (2018); Sirignano & Cont (2019)
6. Model	Deliver predictions	Versioned serving,	Olston et al.

serving	with low latency	autoscaling	(2017)
7. Autonomous operations	Monitor, diagnose, and correct infrastructure issues	Bounded AI agents	Sannidhanam (2025)
8. Governance and resilience	Control change and meet resilience obligations	Configurable change workflows, exit and continuity plans	Basireddy (2022); European Parliament & Council (2022)

Workload placement. Following Sannidhanam (2020), each workload is placed according to its data sensitivity, latency, and elasticity needs. Customer-identifiable transaction data and core scoring may run in private or dedicated environments. Stress testing and model training, which need large temporary capacity and can use pseudonymized data, run in public cloud.

Orchestration. Kubernetes (Burns et al., 2016) provides one way to deploy and scale all workloads across environments. Separate resource pools isolate latency-critical scoring from batch and training jobs so that they do not compete for capacity.

Storage. A lakehouse (Armbrust et al., 2021) holds transactions, positions, market data, and features. Versioning allows any regulatory report or model training run to be reproduced from the exact data used at the time.

Stream processing. Payments and market data are processed as streams with exactly-once guarantees (Armbrust et al., 2018). The same logic can be applied in batch for end-of-day reconciliation, which keeps real-time and batch results consistent.

Distributed training. Large models are trained across GPU clusters using efficient data-parallel training (Sergeev & Del Balso, 2018). The scale of research such as Sirignano and Cont (2019) shows why this capability matters for models learned from high-frequency financial data.

Model serving. Models are served through versioned serving systems (Olston et al., 2017). New fraud model versions can be deployed alongside existing ones and receive traffic gradually, with no interruption to scoring.

Autonomous operations. AI agents monitor metrics, logs, and alerts across the infrastructure (Sannidhanam, 2025). Within predefined limits, they scale services, restart failed components, and reroute traffic. They diagnose more complex problems and propose actions for human approval.

Governance and resilience. Changes proposed by agents or engineers pass through configurable change workflows with approval requirements set by risk (Basireddy, 2022). Resilience

measures, including multi-region deployment, tested failover, and exit plans for critical cloud services, meet operational resilience requirements (European Parliament & Council, 2022).

6. Operating Principles

Four principles govern how the infrastructure is operated.

Match elasticity to workload. Latency-critical workloads keep a guaranteed baseline of capacity and scale out ahead of known peaks. Batch workloads scale up to finish within their windows and then release capacity. Occasional workloads such as stress tests use temporary burst capacity. This lets the institution pay for peak capacity only when it is needed.

Bound agent authority by impact. Autonomous agents (Sannidhanam, 2025) act freely on low-impact, reversible actions within approved limits, such as adding serving replicas during a traffic spike. Medium-impact actions, such as changing resource pool sizes, require one approval through the change workflow. High-impact actions, such as failing over a region or changing network rules, require senior approval (Basireddy, 2022). Every action is logged.

Design for reproducibility. Versioned data (Armbrust et al., 2021), versioned models (Olston et al., 2017), and recorded infrastructure changes (Basireddy, 2022) together allow any past result to be reproduced. This is essential for model validation, audit, and regulatory inquiry.

Treat providers as a risk. Following operational resilience requirements (European Parliament & Council, 2022), critical workloads are designed so that they can fail over across regions and, where feasible, be moved to alternative environments. The portability of containers and orchestration across environments (Burns et al., 2016; Sannidhanam, 2020) supports this.

7. Illustrative Case: A Bank's AI Infrastructure

Consider a mid-sized bank that operates real-time card fraud scoring, daily market risk calculations for its trading book, and a quarterly regulatory stress test on shared cloud infrastructure.

Normal operations. Card transactions stream through the stream processing layer with exactly-once guarantees (Armbrust et al., 2018). The fraud model, served through a versioned serving system (Olston et al., 2017), scores each one in milliseconds. It runs in a dedicated resource pool in a private environment because it handles customer-identifiable data (Sannidhanam,

2020). Each night, an elastic batch cluster computes market risk across the trading book from lakehouse data (Armbrust et al., 2021) and shuts down when the calculation finishes.

Seasonal peak. On a major retail shopping day, transaction volumes rise to four times normal. The operations agent detects rising scoring latency and, within its approved limits, adds serving replicas in the fraud resource pool (Sannidhanam, 2025). Latency returns to target within minutes, and the agent logs its actions. When volume continues to rise and the pool nears its approved maximum, the agent proposes raising the limit. The proposal routes through the change workflow to the on-call infrastructure lead, who approves it (Basireddy, 2022).

Model update. The fraud team trains a new model on two years of transaction data using distributed GPU training in the public cloud (Sergeev & Del Balso, 2018) on pseudonymized data. After validation, the new version is deployed alongside the current one and receives 5% of traffic, then more as its performance is confirmed (Olston et al., 2017). The change workflow records the validation evidence and approvals.

Stress test. At quarter end, the bank must run a regulatory stress test covering thousands of scenarios. Burst capacity is provisioned in the public cloud for three days, the simulation runs across it, and the capacity is released. Because the lakehouse data is versioned, the bank can reproduce the exact inputs if the regulator asks.

Resilience. During the quarter, the bank tests failover of fraud scoring to a second region, as required by its resilience program (European Parliament & Council, 2022), and documents the result. Its exit plan for the stress-testing platform is reviewed and updated.

The case shows how one infrastructure serves very different workloads by matching patterns to their characteristics, and how autonomous operation within governed limits keeps it responsive, efficient, and compliant.

8. Discussion

8.1 Implications

For research, the article connects work on cloud architecture, data systems, distributed ML, and autonomous operations with the specific needs of regulated financial workloads. Its workload characterization offers a basis for comparing infrastructure patterns. Its principle of bounding agent authority by impact extends research on autonomous operations to regulated settings.

For practitioners, the main lesson is to characterize workloads before choosing infrastructure,

standardize on common orchestration and versioned storage, and introduce autonomous agents for low-impact actions first, with operational resilience treated as a design requirement from the start.

8.2 Limitations and future work

The infrastructure was demonstrated through an illustrative case and draws on a focused set of ten studies. Future research should evaluate it empirically, measuring latency under peak load, cost relative to on-premises baselines, time to recover from incidents, and the accuracy and safety of agent actions. Further work is needed on scheduling mixed financial workloads on shared clusters, on setting agent authority limits from evidence, on energy efficiency of large-scale financial model training, and on practical methods for meeting exit-strategy requirements for cloud services.

9. Conclusion

Large-scale financial data processing for AI requires infrastructure that is elastic, fast, consistent, reproducible, and resilient. This article characterized six financial AI workloads and proposed an eight-layer cloud infrastructure. It covers deliberate workload placement, container orchestration, lakehouse storage, exactly-once stream processing, distributed training, versioned model serving, autonomous operations, and governance and resilience. It also set out operating principles that match elasticity to workloads, bound agent authority by impact, ensure reproducibility, and treat cloud providers as a managed risk. The bank case showed the infrastructure handling real-time scoring, peak demand, model updates, and stress testing on one foundation. The central conclusion is that scalable financial AI depends on infrastructure designed around its workloads and operated autonomously within governed limits.

References

1. Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I., & Zaharia, M. (2018). Structured streaming: A declarative API for real-time applications in Apache Spark. In Proceedings of the 2018 International Conference on Management of Data (SIGMOD) (pp. 601–613).

2. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In Proceedings of the 11th Conference on Innovative Data Systems Research (CIDR).
3. Basireddy, S. R. (2022). Digitizing risk management through configurable enterprise workflow architecture. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 14(4), 231–239.
4. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. Communications of the ACM, 59(5), 50–57.
5. European Parliament & Council of the European Union. (2022). Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (Digital Operational Resilience Act). Official Journal of the European Union, L 333, 1–79.
6. Olston, C., Fiedel, N., Gorovoy, K., Harmsen, J., Lao, L., Li, F., Rajashekhar, V., Ramesh, S., & Soyke, J. (2017). TensorFlow-Serving: Flexible, high-performance ML serving (arXiv:1712.06139). arXiv.
7. Sannidhanam, A. H. (2020). Comparative analysis of cloud computing architectures for enterprise applications. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 12(02), 169–180. <https://doi.org/10.18090/samriddhi.v12i02.16>
8. Sannidhanam, A. H. (2025). Autonomous AI agents for cloud infrastructure operations. Journal of Contemporary Science and Technology Management, 1(01), 83–100. <https://jcstm.info/index.php/jcstm/article/view/20>
9. Sergeev, A., & Del Balso, M. (2018). Horovod: Fast and easy distributed deep learning in TensorFlow (arXiv:1802.05799). arXiv.
10. Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. Quantitative Finance, 19(9), 1449–1459.