ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

Generative AI Architectures for Real-Time Transaction Processing

Jens Lehmann
Asst Professor

Abstract

Generative AI (GenAI) is opening new capabilities in software development and operations, reinforcing the broader shift toward DevOps principles and practices. At the same time, organizations across industries are under growing pressure to process data in real time. In financial services, this pressure is most acute in transaction pipelines, which must absorb volume spikes during Black Swan events despite comparatively modest average loads. GenAI-enabled DevOps offers promising ways to accelerate the design, implementation, and monitoring of such pipelines, yet its adoption in this context surfaces unresolved tensions. This paper reviews the existing literature, identifies gaps, draws on Generalized Enterprise Architecture Theory, and formulates research questions whose resolution can produce solutions that reconcile GenAI-enabled DevOps principles with the demands of real-time transaction processing. Toward that end, the paper lays the groundwork by specifying the functional and nonfunctional requirements of such pipelines, surveying GenAI-driven DevOps concepts, architectures, and operational paradigms relevant to financial systems, and outlining suitable design patterns along with their associated challenges and trade-offs.

Keywords: Generative Artificial Intelligence, GenAI Enabled DevOps, Real Time Transaction Processing, Financial Services Pipelines, Black Swan Event Handling, Enterprise Architecture Theory, Cloud Native Architectures, Transaction Pipeline Scalability, DevOps Automation Practices, Real Time Data Processing, Operational Resilience Engineering, Non Functional Requirements Analysis, Intelligent Pipeline Monitoring, Adaptive System Design, Financial Infrastructure Modernization, Event Driven Architectures, High Volume Transaction Systems, AI Assisted Software Operations, Architectural Patterns And Trade Offs, Enterprise Grade Transaction Systems.

1. Introduction

Advancements in Generative Artificial Intelligence (GenAI) and DevOps enable automation across the entire lifecycle of pipelines for online detection of fraudulent activities in real-time financial transaction processing. Historical lessons and challenges in implementing data-driven GenAI models, organizations' needs for accelerated product delivery, and rising cybercrime costs demand automation in operationalization as functionally as possible. GenAI-Enabled DevOps for Real-Time Financial Transaction Pipelines addresses GenAI-enabled DevOps for real-time financial transaction pipelines, characterizing stakeholders, scoping DevOps automation in control, scheduling, and security, and presenting ar-chitecture patterns. Considered deployment scenarios and architectures apply specifically to online detection of fraud-ulent financial transactions, with a broader lens on operationalization-as-a-function of organizations' data pipelines.

Real-time data processing has seen increasingly adoption for multiple usage scenarios, especially by financial institutions, enabling them to detect and respond to events that require operational or business action ex-plaining GenAI-driven DevOps concepts, architectures, and operational paradigms suitable for financial data pipelines. Automation capabilities across pipeline op-erations—model training and deployment, pipeline continuous monitoring, automated rollback, security-governed monitoring, and trade-offs—simultaneously support multiple use-cases, such as anomaly detection for security

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

incidents, fraud detection for financial transactions, and prediction of user behaviours. Numerous architectures for secure, scalable, and performance-optimized implementation of data processing pipelines with controls on data access and governance address these challenges, focusing mainly on data processing.

1.1. Overview of the Study

Automation across the GenAI model lifecycle enables the construction of-DevOps-driven data transaction pipelines for inference and decision-making tasks whose latency, operational knowledge, and expected service horizons correlate. Although it is rare to parallelize model prediction across multiple units with independent data, since the data must be routed through different branches according to its classification, high-frequency decision-making can also be achieved. These models are often known as Single-Task Models. Application in the banking and insurance industries requires special attention to Model Governance since profound consequences from decision errors or a breach in trade-off rules can occur; in those situations where a clear violation of the Business Policy occurs, the reason is usually also evident, so an adequate level of Explainability can be achieved and its adequate monitoring is less critical. The GenAI-DevOps principles generally enable reasonably precise detection of potential errors in a GenAI-driven pipeline.

Implementation in GenAI-DevOps is nevertheless sensitive to several factors. For missions demanding low latency and high throughput and, in particular, for those using recurrent models, the size of the model, the required speed of the inference rate, and the type and frequency of the monitoring checks introduce a complex trade-off. A parallelized model or an Externalized Model must avoid excessive size and complexity, while maintaining low cost and duration of inference, monitoring checks should preferably be executed during non-productive periods and by a less powerful and economical infantry non accessed by GenAI Decision-Making during production hours.

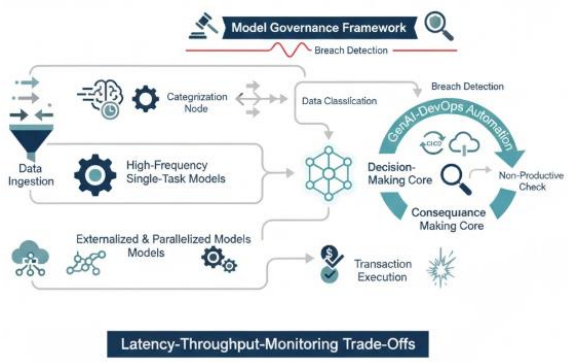


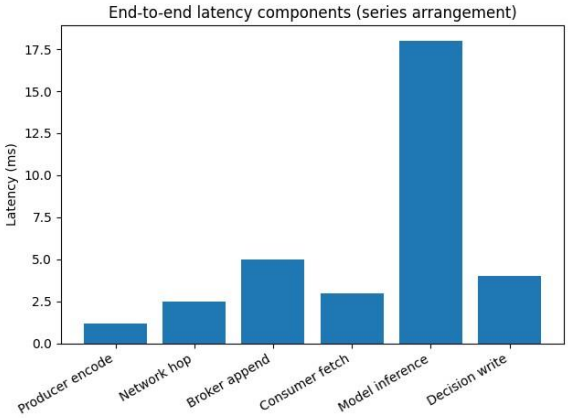
Fig 1: Optimizing the GenAI-DevOps Lifecycle: A Framework for High-Frequency Inference, Latency Trade-offs, and Governance in Financial Decision Systems

2. Background and Related Work

Development and Operations (DevOps) shares with GenAI the ambition of accelerating release cycles while improving quality. Operational characteristics further inspire the adaptation of GenAI to gain efficiency, minimize downtime, and dynamically scale service resources. Integration of open-source GenAI into Hybrid Cloud production environments is viewed as a likely avenue: data ownership, residence, and sovereignty concerns associated with Third-Party Clouds can be ameliorated through a Hybrid Cloud architecture that integrates the Cloud with localized Edge Machines for workloads sensitive to latency. More generally, Data Engineering and Event Sourcing for real-time processing are important domains that witness the emergence of GenAI tools and concepts. Milestones in their evolution into aligned domains are describ-ed, followed by an assessment of architectures, principles, and empirical realizations of DevOps and GenAI, as well as real-time processing. Real-time systems and their pipelines impose strong constraints on availability and response time; major incidents have starkly demonstrated the dire consequences of failure. Misuse of erroneous models can be catastrophic; automatic retraining models with new data and technologies that replace smart

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 04 ISSUE: 03
RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

humans are, indeed, the path to disaster. Wariness of blithe acceptance of GenAI suggestions is warranted, and careful consideration is required to reduce their operation risk.



Equation 1: Latency & Throughput Targets — complete derivations

1.1 Transactions per second (TPS)

Let:

- N = number of transactions processed in time window T seconds

Definition

$$\text{TPS} = \frac{N}{T}$$

Visa up to 65,000 TPS with **average latency 36 ms** .

1.2 Throughput (MB/s) and conversion to TPS

Let:

- B = throughput in MB/s
- S = average message size in MB/transaction

Then transactions per second:

$$\text{TPS} = \frac{B}{S}$$

Step-by-step

1. MB per second divided by MB per transaction gives transactions per second:

$$\frac{\text{MB}}{\text{s}} \div \frac{\text{MB}}{\text{txn}} = \frac{\text{txn}}{\text{s}}$$

2.1. Historical Insights and Key Developments

The maturity of every social, economic, and technological aspect can be seen from the DevOps and GenAI literature is undeniable. The FiTS e-architecture is supported with continuously evolving industrial services. Although there is a lot of innovation happening at various fronts of business verticals, the journey of FiTS is the most appropriate to review for its complexity, integration, security and regulatory requirements. The key financial regulations like PCI-DSS, GDPR, BASEL III, Dodd-Frank, AML, BSA, the list goes on, affects every financial institution around the world. Many services required for these regulations are being offered based on Forms-of-subscribing. Data Processing-as-a-Service (DPaaS), Model-Lifecycle as a Service (MLaaS), Ground-motion-model as a Service (GMMaaS), Model as a Service (MaaS), Log-Monitor-as-a-Service (LMMaaS), Data-collection and Validation-as-Service (DCVaaS), Software as a Service (SaaS), Detect and Display (D&D) as a Service, Detect and Take Action (DTA) as a Service, Detect and Surface (D&S) as a Service, etc., relieve SMEs from any capital investment and allow them to remain competitive with any high-cost subscription for complex services. Standards and tools allow integration with better security and reduced effort. Three architectural styles catering to specific transaction processing speed requirements have matured and have become reference models for Financial-as-a-Service (FaaS). Financial information exchange, the use of ATMs, and online services for transferring money, payments, stock-trading, etc., has integrated the services required to fulfil mandates. These service compositions are essentially integration of existing-monitoring-and-detection-and-display services. Retail a and b evolve from these methods.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

Special channels are also formed between the bank and designated customers based on the high volume of transactions in order to serve specialized needs. Financial institutions are offering relational database services, and point-of-interest authorisation, processing, and settlement services are moving towards Cloud. The SLAs of these services, and real-world patterns of market data motion have enabled the COTS products like CORBA/IIOP/DBPSRL demand-pull-algorithms.

3. Requirements for Real-Time Financial Transaction Pipelines

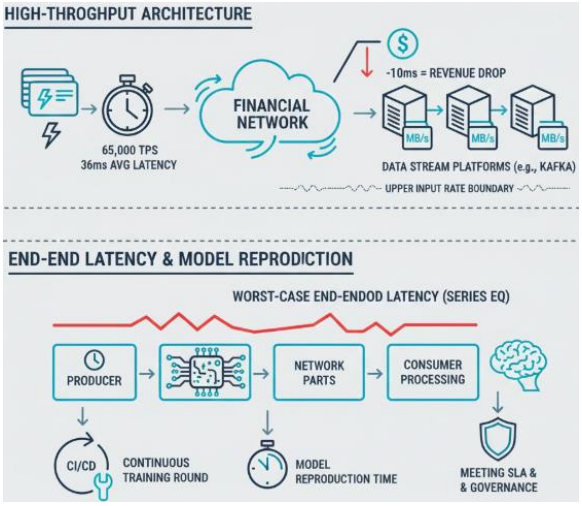
A comprehensive overview of the functional and nonfunctional requirements for real-time financial transaction pipelines is presented. Data correctness, integrity, security, and regulatory compliance are highlighted as pivotal nonfunctional requirements. A clear treatment of latency targets and throughput metrics aligns the discussion with concrete service-level objectives; requirements for data provenance, auditing, and observability establish criteria for external and internal governance. Together, requirements for consistency models, fault tolerance schemes, idempotency, replay protection, and end-to-end reliability guarantees address the preservation of nonfunctional properties during fault recovery.

Functional and nonfunctional requirements for real-time financial transaction pipelines have far-reaching implications yet are rarely articulated with sufficient care. Service-level objectives for latency, throughput, and rollback time influence the choice of architectural, operational, and governance design decisions. While functional and operational design considerations are often partitioned during design, effective implementation of the requirements requires integration awareness from both perspectives. Implementation trade-offs frequently necessitate intra- and cross-discipline compromises between security, data sensitivity, operational complexity, auditability, oversight, and environmental impact.

3.1. Latency and Throughput Targets

Demanding latency and throughput objectives are apparent from the governance policies and service-level agreements of many real-time financial processing environments. For example, Visa's network supports up to 65,000 transactions per second (TPS), with an average latency of 36 milliseconds. An internal study at JPMorgan Chase indicated that a latency of over 10 milliseconds during card authorization leads to a revenue drop. Data stream operators usually define their processing model and workflow based on the anticipated maximum rate of each data source, which serves as the upper input rate boundary for the processing workflow at scale.

Data streaming platforms, such as Apache Kafka and Confluent, define a specific target throughput of MB/s related to each broker. The end-to-end latency of data streams is usually associated with the message-processing delay of the various consumers and the producer acknowledgment time. The different parts of the network have different latencies, which leads to the concept of "worst-case" end-to-end latency that can be computed using the equations of series arrangement in logic circuits. Within the context of model reproduction, a key metric in the DevOps lifecycle is the amount of time it takes to complete a round of continuous training.



ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 04 ISSUE: 03
RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

Fig 2: Quantifying the Latency-Revenue Nexus: Scalable Stream Architectures and Worst-Case Performance Modeling in High-Frequency Financial Networks

3.2. Consistency and Fault Tolerance

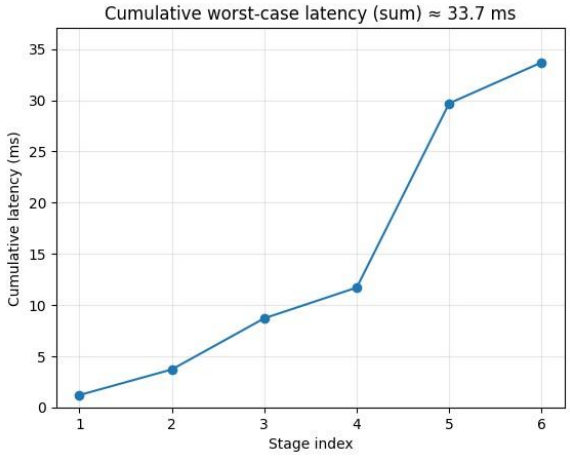
Real-time financial systems are severely affected by node or region failures. In a Service Level Agreement or Service Level Objective–driven environment, it is paramount to minimize recovery windows, and therefore advanced, nontrivial, and often customized solutions are usually adopted. As highlighted in prior work, solutions consist of data duplication, partitioning, and replication across regions and clouds to address region or cloud failures. However, it is also important to focus on the end-to-end implementation guaranteeing data consistency, completing the process without generating duplicates or packet drops, and achieving exactly-once semantics.

Managing in-flight data is critical for its correct management; therefore, mechanisms to identify the data processing path, such as sequence identifiers, and capabilities to replay in-flight data during a failure period must be implemented. Also, for proper consumption on the receiving side, it may be required that each message contains the complete state of a certain element (for instance, the last balance held by an account) or that messages define a sequence that permits reconstructing the information on the other side.

4. GenAI-Driven DevOps: Concepts and Paradigms

This section explains concepts, architectural patterns, and operational paradigms of GenAI-enabled DevOps that adapt to real-time financial transaction pipelines. These elements support automation throughout the pipeline lifecycle, from model training and deployment to monitoring, rollback, and governance. GenAI capabilities enable a range of automation solutions for GenAI and non-GenAI models. Most tasks—except model training and some aspects of deployment—are executed in production environments; Model monitoring techniques are integrated with standard performance, security, and resiliency monitoring procedures.

GenAI- and non-GenAI-based business-specific models are trained and validated before being deployed in their runtime environments. Model inputs are continuously assessed by standard data pipelines. Underperformance triggers dedicated retraining pipelines to create new model versions. DevOps returns these models to production through automated deployment processes that perform the necessary rollout and rollback actions. Model validation is similarly automated; the processes detect functional segmentation and quality discrepancies between new and incumbent models and govern making them effective. These procedures facilitate continuous model validation with low overhead and support online learning when appropriate.



Equation 2: Queueing latency under peak load (Black Swan) — step-by-step

A standard minimal queueing model to show "why latency explodes near capacity" is **M/M/1**.

Let:

- arrival rate λ (TPS)
- service rate μ (TPS), where $\mu > \lambda$

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 04 ISSUE: 03
RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

Expected time in system

$$W = \frac{1}{\mu - \lambda}$$

Derivation sketch (stepwise)

2. Utilization:

$$\rho = \frac{\lambda}{\mu}$$

2. Expected number in system for M/M/1:

$$L = \frac{\rho}{1 - \rho}$$

3. Little's Law $L = \lambda W$ gives:

$$W = \frac{L}{\lambda} = \frac{\rho}{\lambda(1 - \rho)}$$

4. Substitute $\rho = \lambda/\mu$:

$$W = \frac{\lambda/\mu}{\lambda(1 - \lambda/\mu)} = \frac{1}{\mu - \lambda}$$

4.1. Automation in Pipeline Lifecycle

Broad automation—covering the entire financial transaction pipeline lifecycle—is an indispensable element of GenAI-enabled DevOps. It encompasses the development, deployment, monitoring, rollback, and governance of the GenAI models that serve the pipeline. Automation of model training and deployment is relatively straightforward, with platforms like Amazon SageMaker Stream Processing facilitating both activities. Monitoring, however, is more complex; a monitoring solution should enable not only observability into model predictions, but also generation of alerts when production predictions begin to diverge from production labels or when confidence in predictions falls below a specified threshold. Sufficient confidence is critical, as inference costs usually increase with model size—and delayed attention to a failing model may compound operational risk. Where possible, such monitoring should also support early warning for other pipeline components, enabling corrective measures (like alternative processing paths) that minimize short-term disruption.

Automation of model rollback may also be of interest, especially in scenarios where a failed prediction causes higher downstream costs than simply waiting for a return to service. Additionally, prediction monitoring may yield sufficient lead time for supervised model retraining; any such retraining should incorporate the labels accompanying the production predictions. Governance automation facilitates quality-assured model updates by operationalizing traditional data science governance, testing, and monitoring procedures. A pivotal aspect here is the dimensionality of the training data: larger data volumes lead to larger models, which add cost and risk to inference.

5. Architectural Patterns for GenAI-Enabled Pipelines

A set of architectural patterns emerges from the preceding analysis and can accommodate GenAI-driven automation of the full pipeline lifecycle: (i) Hybrid Cloud and Edge Architecture Patterns define the integration of Hybrid and Edge-Cloud infrastructures with the application workload, (ii) Streaming Ingestion and Processing Patterns outline the design aspects of streaming data ingestion and processing for the pipeline.

Hybrid Cloud and Edge Integration Patterns establish the topology and placement of workload execution. Depending on the traffic flows, processing workloads and data regulatory and sovereignty requirements, ingestion and processing may be executed using an edge-local solution or engineered in collaboration with the Hybrid Cloud, using the edge as an accelerator that moves data close to the Cloud resources that actually process it. The Streaming Ingestion Pattern governs the real-time ingestion of streaming data flows, focusing on the definition of the input event schema, control and coordination mechanisms for event windowing, any required event enrichment, stateful processing with updates to internal state stores, idempotent writes to external systems, and requirements for exactly-once processing guarantees.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

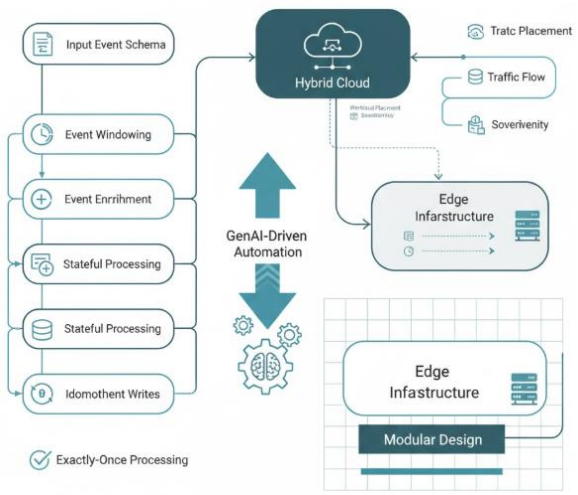
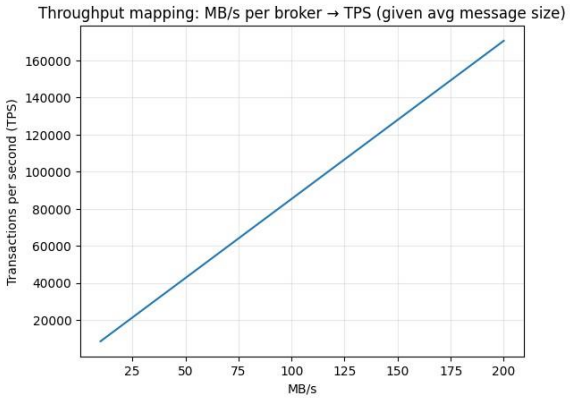


Fig 3: Integrated Architectural Patterns for GenAI Lifecycle Management: Synergizing Hybrid Edge-Cloud Topologies with Stateful Stream Orchestration

5.1. Hybrid Cloud and Edge Integration

Hybrid Cloud infrastructures enable the best of both worlds by combining the benefits of Public Clouds (unlimited scalability, low cost, diversity of services) with the advantages of Private Clouds (cost control, exclusivity, Security, and Compliance). Many financial institutions leverage a Hybrid Cloud setup to store non-sensitive data on the Public Cloud or leverage services such as Natural Language Processing, Optical Character Recognition, or SenseMaking that have a high potential cloud cost due to large Resource Management Management requirements. Workloads are offloaded to the Public Cloud when it is cheaper than using the on-Premise Infrastructure. Placing the Transaction Pipeline either fully in the Private Cloud or Partitioned with parts on the Public Cloud makes sense from a Cost Efficiency and Risk Compliance point of View. Certain Data Paths, such as Data Classification, Money Laundering Monitoring, Fraud Detection, are however Priority Cloud Paths for these Financial Institutions and are required to execute locally to Safeguard Sensitive Data or to Ensure Low Latency Operation. Executing on Private Edge Sites leverages the low-latency and low-continuous cost Benefits of Edge Processing while maintaining the Integrity of the Data within the Organization. Thus, a Subset of Workload is Specified to run on the Edge.



5.2. Streaming Data Ingestion and Processing

A typical architectural pattern for continuously ingesting streaming data and performing real-time analysis and processing is shown in . Such architectures form the foundation of most GenAI-enabled application scenarios that are not one-shot transactions but require continuous interaction with external users or systems and, consequently, have a large number of transactions not all requiring GenAI model inference capabilities. Data is continuously ingested from one or multiple sources, analyzed, transformed, and possibly enriched using model inference along the way. The results from these analyses, transformations, and enrichments are streamed to one or more systems. While the simplest instantiation of this pattern treats all incoming data as atomic transactions, the pattern becomes far more useful when support for evolving schemas and throwing away, modifying, or adding some of the ingested messages over time is incorporated. Streaming ingestion can be driven by a variety of events and trigger different types of processing. Very common schema types for ingestion are messages from message buses, traffic and sensor data, logs, and clickstreams. Data can also be ingested in a batched manner while still being processed with millisecond resolutions. Every ingesting message has a schema associated with it. The processing

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 04 ISSUE: 03
RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

component can be internally merged into one component or distributed over many for both performance and scaling reasons. It can process different windows of the data, re-ingest messages into a message bus once processed, or route them to different processing components. Different processing components can operate on the same data using different windows or separated amounts of data. Windows can be generated using sliding time windows, tumbling time windows, or session-driven windows. Windows processing with complete and update models is also commonplace.

6. Challenges and Trade-offs

Concrete challenges and trade-offs surface throughout pipeline design and operation. Different design aspects must be carefully balanced to avoid misguided decision-making. For instance, supporting strict latency and high throughput in an automated high-volume fraud detection application using 10 challenger models covered twenty slices of the prediction phase, adding significant overhead. Motivation for the additional latency and cost stemmed from the fact that a single successful fraud attempt, typically demanding at least in the low five-part sum, could have high impact on the business.

User acceptance, maintenance, monitoring, and continuous assessment of model quality become increasingly challenging at scale, and the need for governance, security, and explainability increases with the size of the generative models. Manpower-demanding complex models, such as text-to-code or text-to-image systems, enable business transformation. However, negligence of all three aspects of MLOps could result in high operational risk or worse business outcomes than with simpler, well-governed, and comprehensively monitored solution models.

Equation 3: Consistency, replay protection, exactly-once — equations/logic

3.1 Idempotent processing condition

Let:

- message has unique sequence id s
- state store keeps processed ids set D

A processing function is idempotent under duplicates if:

$$\forall s: \quad \text{Apply}(s) \text{ multiple times} \equiv \text{Apply}(s) \text{ once}$$

Implementation rule:

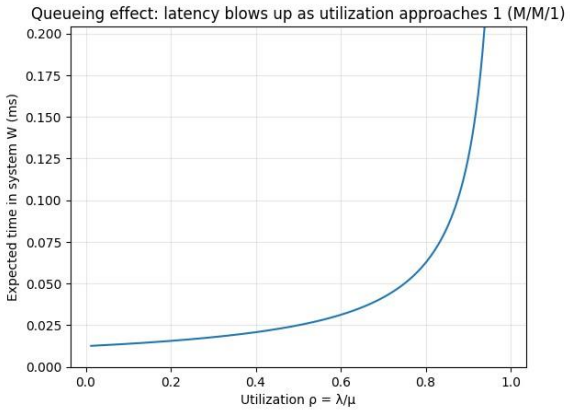
$$\text{if } s \in D\text{: skip output} \quad \text{else: process and add } s \to D$$

This is the mathematical basis for “no duplicates” even under replay.

3.2 Replay correctness

If failure causes messages $s_a, \ldots, s_b$ to be replayed, correctness requires:

$$\text{Final state after replay} = \text{Final state if no failure occurred}$$



6.1. Performance vs. Model Complexity

Automating the development and deployment of GenAI models can reduce risks associated with human intervention but may also increase it, especially for low-latency environments. Long-running Pipeline

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 04 ISSUE: 03
RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

Enhancements deliver pre-trained models that potentially lower operational risk, but they accelerate the flywheel in the opposite direction, since unMonitored or poorly placed models can exert significant operational pressure. As with all changes in open systems, care must be taken to keep entry standards high. Human review and patience are just as important as the automation of mundane operational tasks. Although unMonitored models can relieve pressure from the backlog, they place additional pressure on operations. The addition of an unMonitored model should surface operational pressure in the monitoring. The accepted complexity of a model is not a simple median: model performance and operational pressure are often highly correlated. Balancing model size and parameter count with latency requirements, supported data, and monitoring is important to strike the right operational complexity.

The tension between latency and throughput is multi-dimensional. A single low-latency model placed at ingress can often be replaced with several larger models with lower per-inference latency and more complex processing pipelines with larger windowing bursts. Reducing the frequency that GenAI calls are made can reduce the pressure on monitoring and repoussé systems that track and redirect the model traffic and populate the training data needed for long-running pipelines.

7. Conclusion

GenAI, banks, and other financial market participants are keenly exploring the potential of engines like OpenAI's ChatGPT for a variety of use cases. This paper narrows the focus to the area of DevOps applied to Financial Pipelines needing a real-time capability. The paper synthesizes the active and important area of DevOps with GenAI techniques and their ramifications for architecture and pipelines in real-time financial transaction processing systems. GenAI-enabled DevOps–a formalization of both concepts in this application domain–offers the promise of lessening the burden of these, often manual and random, activities in the operational lifecycle and governance of such systems, enabling pipeline owners to operate their systems with a greater degree of confidence.

GenAI, banks, and other financial market participants are keenly exploring the potential of engines like OpenAI's ChatGPT for a variety of use cases. This paper narrows the focus to the area of DevOps applied to Financial Pipelines needing a real-time capability. The paper synthesizes the active and important area of DevOps with GenAI techniques and their ramifications for architecture and pipelines in real-time financial transaction processing systems. GenAI-enabled DevOps–a formalization of both concepts in this application domain–offers the promise of lessening the burden of these, often manual and random, activities in the operational lifecycle and governance of such systems, enabling pipeline owners to operate their systems with a greater degree of confidence.

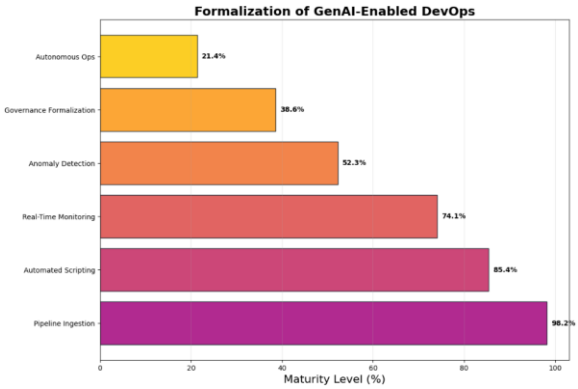


Fig 4: Formalization of GenAI-Enabled DevOps

7.1. Final Thoughts and Future Directions

Despite the operational challenges and risks, the potential rewards warrant continued experimentation with GenAI-enabled DevOps in real-time financial transaction pipelines. These systems automate the management of the pipeline model lifecycle, from training and deployment to monitoring, rollback, and governance. Such automation allows organizations to decisively reap the benefits of real-time systems—reduced latency and increased business value—while offloading continuous operations to a controlled, automated process that requires limited manual intervention.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

As GenAI improves and stabilizes, areas to monitor and explore include support for dispersed Hybrid Cloud-Edge pipelines with strict data sovereignty constraints; reliable streaming data ingestion and processing; and continuously supervised, multi-objective, long-term training of models for multiple and evolving objectives. The need for oversight and management of the auto-generation, in-context training, and execution risk fosters a second axis for monitoring and experimentation—establishing and controlling an appropriate balance between simplicity and transparency of GenAI usage and the associated risk of blind trust in the output. A third direction for focus is the definition of a suitable standard operation generator and the exploration of the trade-offs between operational complexity, optimal operation, and exploitation of operation explainability.

8. References

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.
2. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.
3. Mashetty, S., Malempati, M., Paleti, S., Adusupalli, B., & Singireddy, J. (2025). A Multidisciplinary Framework for AI and Data-Driven Transformation in Taxation, Insurance, Mortgage Financing, and Financial Advisory: Integrating Cloud Computing, Deep Learning, and Agentic AI for Community-Centric Economic Development. Insurance, Mortgage Financing, and Financial Advisory: Integrating Cloud Computing, Deep Learning, and Agentic AI for Community-Centric Economic Development.
4. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1–67.
5. Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P.-S., ... Hassabis, D. (2021). Scaling language models: Methods, analysis & insights from training Gopher. arXiv preprint arXiv:2112.11446.
6. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
7. Kummari, D. N., Singireddy, J., Sheelam, G. K., Nandan, B. P., Pandiri, L., Lakkarasu, P., & Dwaraka. (2025, August). Generative AI Models for Process Optimization in Semiconductor Wafer Design and Yield Prediction. In International Conference on Artificial Intelligence: Theory and Applications (pp. 220-233). Cham: Springer Nature Switzerland.
8. Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120), 1–39.
9. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvetkov, Y., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., ... Fiedel, N. (2022). PaLM: Scaling language modeling with pathways. Journal of Machine Learning Research, 24(240), 1–113.
10. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

in large language models. Advances in Neural Information Processing Systems, 35, 24824–24837.

11. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730–27744.
12. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. In International Conference on Learning Representations.
13. Mashetty, S. (2025). LEVERAGING DEEP LEARNING, NEURAL NETWORKS, AND DATA ENGINEERING FOR INTELLIGENT MORTGAGE LOAN VALIDATION. INTERNATIONAL JOURNAL OF SOCIAL SCIENCE & INTERDISCIPLINARY RESEARCH ISSN: 2277-3630 Impact factor: 8.036, 14(04), 51-65.
14. Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. Advances in Neural Information Processing Systems, 35, 16344–16359.
15. Aminabadi, R. Y., Rajbhandari, S., Zhang, M., Awan, A. A., Li, C., Li, D., Zheng, E., Rasley, J., Smith, S., Ruwase, O., & He, Y. (2022). DeepSpeed inference: Enabling efficient inference of transformer models at unprecedented scale. Microsoft Research Technical Report MSR-TR-2022-21.
16. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36, 68539–68551.
17. Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Brown, A., Bubeck, S., Cao, M., Caswell, I., Cecchi, M., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., ... Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
18. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
19. OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., ... Zoph, B. (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774.
20. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles, 611–626.
21. Dao, T. (2023). FlashAttention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691.
22. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. Advances in Neural Information Processing Systems, 36.
23. Adusupalli, B., Malempati, M., Paleti, S., Mashetty, S., & Singireddy, J. (2025). Integrated financial ecosystems: AI-driven innovations in taxation, insurance, mortgage analytics, and community investment through cloud, big data, and advanced data engineering. Journal of Information Systems Engineering and Management, 10, 1103-1117.
24. Zhang, S., Zeng, X., Wu, Y., & Yang, Z. (2023). Harnessing scalable transactional stream processing for managing large language models. arXiv preprint arXiv:2307.08225.
25. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. L., Bressand, F., Lengyel, G., Bour, G., Lample, G., ... Scao, T. L. (2024). Mixtral of experts. arXiv preprint arXiv:2401.04088.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 04 ISSUE: 03

RECEIVED: JULY 17 REVISED: JULY 26 ACCEPTED: AUGUST 04 PUBLISHED: AUGUST 13

26. Gemini Team. (2024). Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.
27. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., ... Zhang, J. (2024). The Llama 3 herd of models. arXiv preprint arXiv:2407.21783.
28. Hammami, H., Baligand, L., & Petrovski, B. (2024). Fighting crime with Transformers: Empirical analysis of address parsing methods in payment data. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 201–212.
29. Bernardi, M. L., Casciani, A., Cimitile, M., & Marrella, A. (2024). Conversing with business process-aware large language models: The BPLLM framework. Journal of Intelligent Information Systems, 62, 1607–1629.
30. Karst, F. S., Chong, S.-Y., Antenor, A. A., Lin, E., Li, M. M., & Leimeister, J. M. (2024). Generative AI for banks: Benchmarks and algorithms for synthetic financial transaction data. arXiv preprint arXiv:2412.14730.
31. Recharla, M. (2024). Antioxidants, Biological Markers, Catalase, Glutathione Peroxidase, Chronic Periodontitis, Saliva, Smokeless tobacco, Smoker. Frontiers in Health Informatics, 13(8), 4999.
32. Murtaza, S. S., Nie, Y., Avan, E., Soni, U., Liao, W., Carnegie, A., Mathias, C. J., Jiang, J., & Wen, E. (2025). Implementing retrieval augmented generation technique on unstructured and structured data sources in a call center of a large financial institution. In Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 598–606.
33. Bhupathi, S. (2025). Role of databases in GenAI applications. arXiv preprint arXiv:2503.04847.
34. Ouafiq, E. M., & Saadane, R. (2025). Retrieval-augmented OLAP: Generative AI architecture for smart systems & equipment. Proceedings of the AAAI Symposium Series, 6(1), 304–312.
35. Ghali, M.-K., Farrag, A., Won, D., & Jin, Y. (2025). Enhancing knowledge retrieval with in-context learning and semantic search through generative AI. Knowledge-Based Systems, 311, 113047.