JOURNAL OF INNOVATIVE ACADEMIC RESEARCH

ISSN : 3049-2122

VOLUME: 02 ISSUE: 01

RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

Trade to Trust Real-Time AI Compliance for Derivatives

Ethan Williams
Asst Professor

Abstract

The growing demand for low-latency computing in derivatives markets challenges the traditional approach of monolithic applications implemented in dedicated data center clusters. Cloud-native architectures offer a promising solution due to the flexibility, scalability, and resilience of the cloud. Nevertheless, for latency-sensitive workloads, latency is not only a requirement; cloud-native design principles can lead to improvements in the computing model itself. These include breaking down monolithic applications into microservices deployed in a service mesh, taking advantage of on-demand and pay-per-use resources to optimize costs for off-peak traffic, and improving computing efficiency through loading-balancing across cloud availability zones.

Furthermore, the demonstrably increasing risk of operational resilience in the cloud at a suitably architected cloud level requires organisations and business partners to reconsider their outsourcing risk appetite. Even where a loss occurs, insurance can cover events, as long as the incident doesn't lead to the organisation ceasing to trade for an extended period. For many services, intelligent brokerage can ensure continued service. Examined in more depth, these principles may also strengthen the fundamental efficiency and performance of the models beyond a simple scaling gain.

Keywords: Low Latency Computing, Derivatives Market Infrastructure, Cloud Native Architecture, Microservices Decomposition, Service Mesh Design, Latency Sensitive Workloads, Real Time Trading Systems, Elastic Compute Scaling, Pay Per Use Resource Optimization, Load Balancing Across Availability Zones, Cloud Resilience Engineering, Operational Resilience Risk, Outsourcing Risk Appetite, Intelligent Service Brokerage, High Availability Architectures, Cloud Based Market Systems, Performance Optimized Computing Models, Financial Market Infrastructure Modernization, Scalable Trading Platforms, Resilient Cloud Ecosystems.

1. Introduction

Market risk, valuations, capital reserves, and liquidity all are governed by financial institutions addressing the derivatives markets, which also remain fundamental to managing risk and enabling speculation. Market participants rely on a spectrum of vendors to price, hedge, and process transactions involving the underlying. Regulatory requirements for transaction submission and trade-related data — driven by the 2008 financial crisis and the collapse of major financial institutions — must be met for all parties and in many cases are accompanied by real-time monetization. A cloud-native approach to product, price, risk, and regulatory governance management is requisite because of the high stakes involved. But a market- and data-driven cloud-native architecture is essential. Products need to be priced accurately, reliably, and with the minimal latency possible. From a computational point of view, the processes can be deep and prone to errors, and there can be a high volume of instant price and risk request traffic.

A cloud-native architecture can deliver the requisite fault tolerance on the pricing side by enabling the underlying

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

price models to be computed easy-to-access cacheable elements in the future via a data lakehouse architecture. The resulting product-price-risk plus transaction-processing storm needs to be processed directly via cloud-native principles in a cloud-native way. And on-the-fly processing to serve the real-time monetization market is required. Such an approach enables a market vendor to underpin real-time price, risk, and transaction processing, offer pricing models for the pricing tree during that market, and allow clients simple latency querying for instantaneous transaction marking, risk updating, and executions during their functional and financial storm period.

1.1. Overview of the Study

Advances in cloud-native technology enable new paradigms for building and operating scalable, resilient systems to improve performance through wider distribution. Market-makers, enables, and liquidity-providing clients are looking for fast execution, which requires low-latency-local solutions that push risk controls as close to the pricing engines as possible. Deploying parts of a pricing or risk computation engine in the cloud, closer to the demands on the asset, may be the only way to fulfil customer demand—and compliance requirements, business continuity, and operational risk—at an acceptable cost. The market for regulations is large, and the response is slow, creating compliance risk. Trading and security-regulatory compliance are separate functions that rarely share any common data. The approval of new regulations follows public scrutiny with lengthy feedback loops and often contradictory outcomes among multiple regulators aiming to reach the same goal. Detecting security and trading infraction transactions often comes too late, as multiple regulators may investigate parallel without real-time intelligence. The traditional waterfall process is too slow and costly for regulatory compliance and real-time detection. Cycles of continuous and discrete audit and control checks are necessary to govern new transactions. An architecture that builds compliance features into the trading system itself, rather than leaving compliance to a post-trade activity, lowers compliance creation costs and increases speed to market for compliance with new rules. By treating compliance as true revenue rather than a support function, the costs can be capitalized and subsequently recovered.

This work outlines a modular cloud-native architect as a foundation for an end-to-end AI-powered solution that rapidly addresses compliance regulations in the derivatives space. The building blocks for a real-time derivatives-processing pipeline are described, and the pipeline is augmented with a trading compliance and security system that meets future-market risk and regulatory demands, with the rapid implementation of new or amended regulatory compliance demands that frequently arise in both areas.

Fig 1: Cloud-Native Modular Architectures for Real-Time Derivatives Compliance: Shifting from Post-Trade Auditing to Embedded Regulatory Intelligence

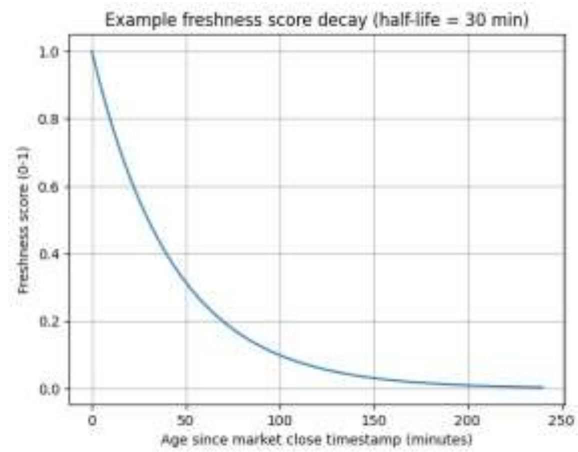
2. Background and Motivation

Many financial institutions have expressed interest in cloud-native architecture for real-time derivatives pricing.

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

In the pricing domain, stringent latency requirements, given the size of the trade, have often meant on-premise solutions to minimize latency, despite cloud-native architecture being easier to build, scale and operationalize. This also reduces vendor concentration risk for resilience as cloud-native microservice-based architectures are managed by different vendors and can be loaded on a per-region basis. Reducing vendor concentration risk for pricing is not only business but also country risk management. This paper models derivatives processing in cloud-native architecture, handling distribution of both pricing and risk computation, with the aim of scaling up compute and also reducing region-based vendor concentration risk on pricing with lower latency for large trades.

However, many compliance and regulatory governance requirements require audit trails, data lineage and security setting that are inherently large overheads. For example, market and data compliance regulations (such as MI-FID II, CCAR, FINRA, DGS, FRTB, BCBS 239, GETS, CGETS) demand a complete1 data and model lineage for years. These regulatory requirements have to be built into the architecture and processes so that the complete audit trace is captured with minimum or no human intervention. The complete data model lineage of both market data and integrity/completeness of the market data is often as large, if not more than, the distribution price computation. Audit trail, security, and access of the business are also as large in overhead as product pricing on a derivatives processing system. Thus, Chuck et al. (Marvel) proposed a compliance rule-based Governance (G) architecture integrated into AI to remove bias and control the AI model audit requirements with minimal overhead in model development and parameter tuning.



2.1. Significance of Cloud-Native Approaches in Derivatives

A cloud-native architecture supports high-speed transaction processing and ultra-low-latency risk management; it addresses other key needs apparent from growing demand for operational resilience and regulatory control.

Rapid growth in the global derivatives market has heightened the importance of low-latency solutions for pricing, risk computation, and operational resilience: Cloud,—and especially cloud-native—architectures are being proposed to respond to these challenges. But derivatives markets have special demands that can make the application of cloud technology in this area tricky. For example, cloud-native technologies are often applied to the construction of operationally scalable infrastructures that facilitate the building and running of many concurrent systems across the financial services industry within a defined Region. However, the need for very low-latency transaction processing, ultra-low-latency risk computation, and operational resilience—where even single-digit sub-millisecond latencies in processing are vital—mean that for such systems cloud infrastructure must be provisioned regionally, within a single data centre, to be effective. Even then, these localised environments may not always have the 3∗2 specifications required for optimal performance.

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

Rather, such environments must be shortened further, to become a 1∗2∗3 zone with extremely low-latency routing. Operational resilience is now required not only for exchange Centre but throughout the ecosystem. The need for these services to be resilient from cloud failure is also important, especially as end-user final-line solution-run teams migrate operations to cloud. Anti-fragility is thus becoming a driver for these solutions. At the same time, demand for robust data governance and management control is gaining new momentum due to the operational demands of evolving Data Protection Acts and Data Protection Regulation, GDPR, MSSP, MiFid II, and many other related regulations that now govern financial services…exponential growth is expected across the industry over the coming years, but also in relation to signalling and alerting.

Metric	Value
Products N	345,000
Time budget T	10 s
Required throughput $R = N/T$	**34,500 products/s**

Equation 1: Pricing throughput equation

Step-by-step derivation

Let:

- $N = 345{,}000$ products
- $T = 10$ seconds (deadline)

Required throughput

$$R = \frac{N}{T}$$

Substitute:

$$R = \frac{345{,}000}{10} = 34{,}500 \text{ products/s}$$

Concurrency (workers/cores) needed

If average compute time per product is t seconds/product (single-threaded worker), then one worker's service rate is $\approx 1/t$. To sustain throughput R, the required concurrency W is:

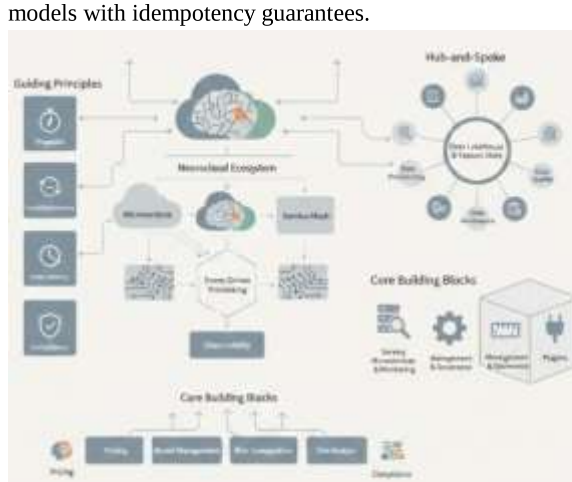
$$W = R \cdot t$$

Example with $t = 5 \text{ ms} = 0.005$ s:

$$W = 34{,}500 \cdot 0.005 = 172.5$$

3. Architectural Principles for cloud-native AI in derivatives

Successfully capturing the advantages of cloud-native computing and deploying a highly automated AI-driven architecture for real-time derivatives transactions should pragmatically follow a set of guiding principles. The intention is to establish a neurocloud-based shared and modular business ecosystem to maximize the flexibility and scalability of pricing, model management, risk computation, distribution, and compliance. Latency-sensitive pricing forms key prerequisites for practical adoption, creating a need for distributed on-demand pricing models with idempotency guarantees.



JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

Fig 2: NeuroCloud: A Modular, Event-Driven Architecture for Automated AI-Enabled Derivatives Liquidity and Compliance

The ensuing architecture must combine microservices to enforce modular boundaries for testing, supports open and maintainable tooling based on stateless services, adheres to the event-driven processing paradigm, provides precise observability, and aligns with all relevant market compliance requirements.

Core building blocks comprise serving microservices and their monitoring, management, governance, and quantification plugins deployed within a service mesh. A hub-and-spoke approach implements a data lakehouse with a feature store to capture and expose query features across the entire business. Data provisioning, governance, semantics, and quality encompass a complete and semantically consistent view of all properties, including the management of data model life cycles to support valid and compliant real-time analytics.

3.1. Core Design Principles for Cloud-Native AI in Derivatives

Core design principles for cloud-native AI in derivatives are articulated, encompassing microservices boundaries, stateless processing, event-driven workflows, idempotency, observability, and governance controls. Modularity allows individual service components to be specified, optimized, maintained, and deployed independently. Scalability requires elastic resource provisioning in multiple cloud regions. Fault tolerance accommodates failures in any component, detecting errors and making appropriate corrections. Security embeds auditing, traceability, and regulatory governance across both architecture and processes.

Architectural development is guided by established cloud-native principles, with six design considerations extending beyond general cloud-native qualities to address AI-specific needs: modularity, scalability, fault tolerance, security, observability, and idempotency. Modularity is a central tenet of microservices, enabling specific services to be optimized and developed independently. Latency-sensitive applications benefit from provisioning in multiple cloud regions, allowing automatic scaling that responds to demand elasticity. AI models for pricing, risk, and other derivatives functions often require intensive compute resources, necessitating careful management to minimize delays. However, resilience to runtime failure is even more critical. Derivatives are valuable assets typically held for only a few days: corresponding trades last a few seconds or minutes, while processing times should aim for milliseconds. Such short-lived, high-volume traffic patterns create the potential for custom-built hardware but do not justify the cost of such an investment; operational services need only fail gracefully.

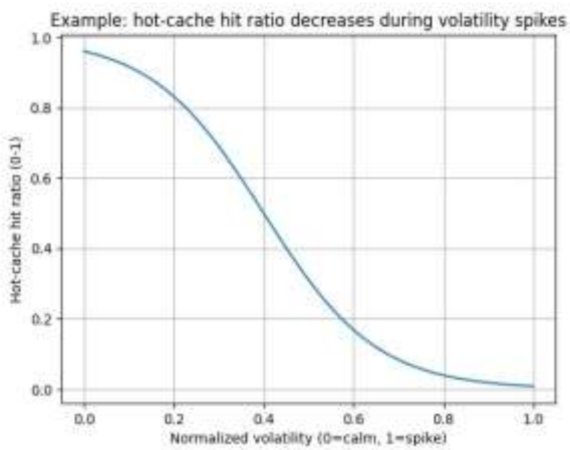
4. Real-time Derivatives Processing Pipeline

The derivatives processing pipeline implements the end-to-end specification. Data ingestion from trading, reconciliation, and market feed systems supports ingestion, validation, and normalization, with effectivity and lineage tagging, feature resolution, and schema-lifecycle controls. Ingestion-enriched events drive event-driven architecture: option-pricing and risk-computation services model compute as features, supporting real-time data science. Cache coherence meets sub-millisecond latency targets, enabling low-latency function publication. Model-serving and feature-store caches portrait feature replays.

The violation of capital and risk limits by dealers causes a volatility surge in derivatives prices while demand rises from clients. These events lead to risk latency, limits violation, and put dealers in arbitrage risk. The background to the architecture allows introducing a microservice development approach using Azure AI ML to provide a real-time option-pricing pipeline serving a low-latency production environment. Integration is sought across multiple cloud-natives and the services computing nodes while providing production-monthly-use change. Microservice principles supporting production-quality

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

components enhance configurability and scalability. Network-load-management patterns and the Kubernetes environment finally provide a multi-environment load balancing and scaling solution.



Equation 2: End-to-end latency equation (sub-millisecond target + cache)

Step-by-step derivation: latency decomposition

Let end-to-end latency be the sum of stage latencies:

$$L_{\text{e2e}} = L_{\text{ingest}} + L_{\text{normalize}} + L_{\text{feature}} + L_{\text{route}} + L_{\text{compute}} + L_{\text{audit}}$$

If the system uses caching, average compute latency depends on cache hit ratio h:

- Cache-hit path latency: L_{hit}
- Cache-miss path latency: L_{miss}

Average:

$$\mathbb{E}[L] \;=\; h \cdot L_{\text{hit}} + (1-h) \cdot L_{\text{miss}}$$

And you still need the compliance/audit emission overhead:

$$L_{\text{e2e}} \;=\; \mathbb{E}[L] + L_{\text{audit}}$$

4.1. Data Ingestion and Normalization

The real-time derivatives processing pipeline operates on market, reference data, and event feeds that drive all aspects of trading operations from pricing and risk management to detection of market anomalies. All feeds are ingested into a dedicated staging area in Cloud Storage, which permits automation and simple parallelism. Data is then consolidated, normalized, and made available to downstream applications in a structured format. An event-driven architecture detects changes and updates interested consumers asynchronously.

Incoming events are validated against quality specifications—such as acceptable ranges for pricing curves or surface normals—and assigned a freshness score based on market close timestamps, ensuring stale data does not propagate. Operations teams receive alerts for flagged deviations, and data is rejected if failing thresholds for critical data sets. Ingested data includes market state and trading positions, ultimately normalized to an open, high, low, and close schema. Limit orders—either volume-weighted intakes for primary, visible orders or implicit orders for non-explicit trading—are built for pricing and risk computation. The IPC schema facilitates association with side information and loading into a temporal data lake for future analysis.

Auditability and lineage are guaranteed by explicit naming patterns in Cloud Storage, which reflect original schema, producer, data classification, source, and freshness. Structural and semantic models enforce quality expectations as part of a broader regulatory compliance initiative, deterring price and market manipulation through the introduction of explicitly orchestrated disorder.

4.2. Event-Driven Computation and pricing

Event-driven computations respond to normalized reference data ingested from disparate sources and reservoirs (Figure 2). Internal pricing services retrieve normalized data for staging feature pipelines—data cleansing, emerging market directions, and product-specific model variables. Prices are computed with separated models for swaps, swaptions, and cross-currency basis. Inputs converge into newly introduced concept services—

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

option tarns for Bermuda-style transactions, lumpsum amounts for notify/confirm segments—and enforce organization-specific features. Market needs dictate high-quality pricing latency, set for 10 seconds with a product count of 345,000. An internal query service retrieves valid market points, identifying dangerously outdated segments. Feature engineering and prediction for the AI-based Risk charge follow the same approach; supporting concepts appear in the third section.

Stability (X) is ensured with pre-computed hot caches for main market segments. Hot caches are useful with stable market texture but may vaporize during a volatility spike or a historic pickup in new issued volumes. Rarely, with longer timeframes, predictions for products with 5-year outlier time variability may be refreshed more slowly. When market conditions call for predictions, a dedicated service invokes fresh computations, and a dedicated replacement process engages when both stability checks fail. Severe market uncertainty heightens a bank's risk of excessive consumerism; during such periods, a bank rushes to plead with the market-makers to conceive a moral treatise ruling out risky positions through modeled consideration.

Avg compute time per product (ms)	Required throughput (products/s)	Approx concurrent workers needed
1	34,500	34.5
2	34,500	69.0
5	34,500	172.5
10	34,500	345.0
20	34,500	690.0
50	34,500	1,725.0

5. Compliance and Regulatory Governance Framework

A compliance and regulatory governance framework for data and analytics is proposed, aligned with market regulations, built-in governance, and observability. It provides a comprehensive mapping of regulatory requirements to organizational and architectural controls, ensuring that adequate processes, tools, and technologies creating data and analytics follow compliance regulations, providing audit trails and traceability. Key aspects include a security by design plan combined with an operational model to ensure separation of duties and role segregation for compliance workload processing. The end-to-end data and analytics flow provides data lineage for market data and pricing resulting from the derivative transaction and compliance process.

Market data created to provide near real-time updates to positions created by the transaction processing system at the exchange or brokerage level and triggering pricing models in stations close to market data sources is monitored through a combination of rules over the pipeline data processed and alerting based on pipeline observability, ensuring the needed trust in this computation. Data observability integrated into the data-flow notifies possible data quality issues. Components processing tagged data for compliance analytics are monitored through these same concepts assuring their availability and reliability to follow the demand of the workload.

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

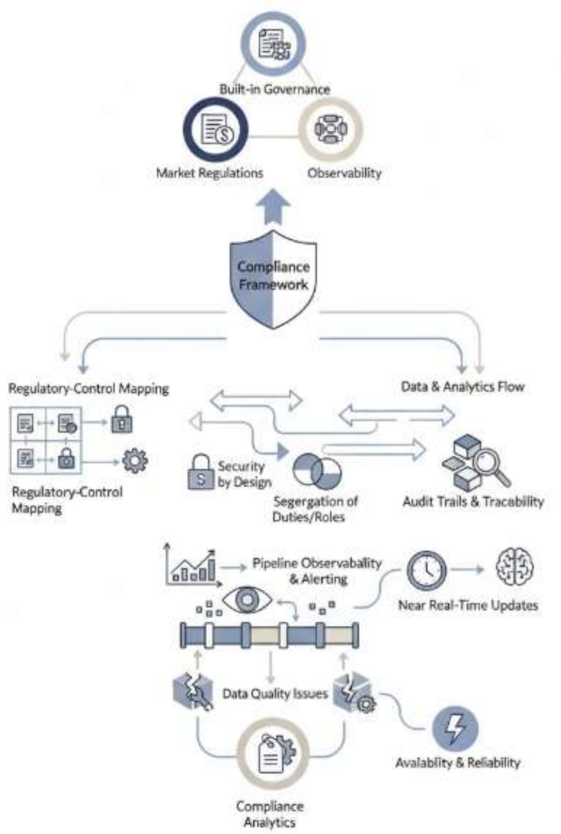


Fig 3: Integrated Governance by Design: A Multi-Layered Observability and Compliance Framework for High-Frequency Market Data Ecosystems

5.1. Regulatory Requirements Mapping

Regulatory governance extends beyond compliance mandates, necessitating formalized processes that respond to internal and external parties, encompassing risk appetite, monitoring, and follow-up. Data and operational pipelines are also subject to legal scrutiny, driven by GDPR, user agreements, and financial regulations. These further conditions determine the lifecycle of data sources and features, with controls for data quality and access.

These requirements represent specific compliance controls within the broader governance framework. Each API needs associated audit trails and traceability. Compliance mandates derive from sensitivity analyses on derivatives pricing. Requirements mapping ensures incorporation into architecture, services, and flows, confirming the deployment of related governance controls.

5.2. Monitoring, Reporting, and Notification

While risk governance is a priority, compliance with other regulations is also critical. A detailed mapping of the relevant regulations ensures that controls are applied to all relevant processes. Because these controls impact nearly all parts of the architecture and organization, they are not separately enumerated here, presenting a risk of underestimating their scope. Nevertheless, compliance with these requirements is paramount.

Reporting and notification requirements arise from many regulations, including anti-money-laundering policies and the Securities and Futures Ordinance. Audit trails are part of many market regulations, such as MIFID-II, SRB, and the Hong Kong Code on Unit Trusts and Mutual Funds. Lineage information, including the schematic details of data sources, is necessary for other regulations, such as BCBS 239 and DFSA Rulebook. Thus separation of duty needs to be implemented, either within the architecture or within specific services, particularly for functions related to compliance and reporting.

6. Cloud-Native Architecture and Infrastructure

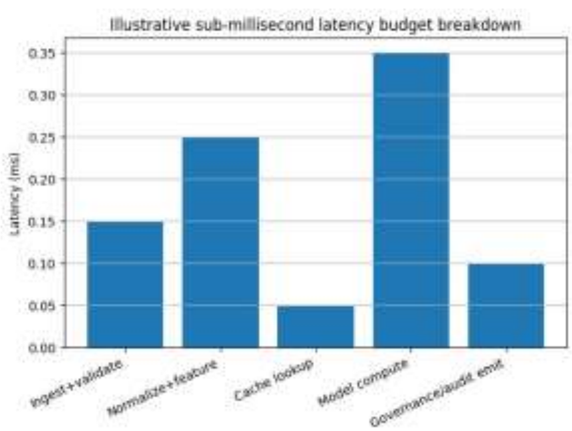
An end-to-end cloud-native architecture with features supporting microservices, security, reliability, and observability patterns completes the discussion. Such a cloud-native architecture facilitates the deployment of an event-based, modular application. A robust service mesh enables secure service discovery, observability, and cross-cutting functionality. Data is managed within a designated

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

cloud account, while optimized infrastructure patterns reduce overhead.

The previously described processing, pipeline, and governance capabilities are realized using an event-driven microservices architecture running on a managed Kubernetes service. Microservices boundaries defined by functional domains, application microservices grouped using logical sensible domains. A dedicated service mesh provides underlying support for service discovery, communication, and security. Observability requirements are fulfilled using open-source projects. Critical auditing, incident-response and enabling the observability gap with trifecta of metrics logging and tracing.

Processing uses a lakehouse approach that unifies a data warehouse with a data lake, having the combined capabilities of the two in a single store. Supporting the accessibility for analytics without being duplicated. The governance of the lake is aided by a data governance tool that maintains a catalogue of metadata and offers enhancements like data-lineage tracing impact analysis classification metadata-based discovery. The governance tool is further supported by a semantic model that adds a business glossary. The feature stores supports the consumption of big-data in machine-learning workload at a faster pace by caching the intermediate features to minimize the processing time. Policy-based management further maintains the sustainable use of the environment.



Equation 3: "Freshness score" equation (data quality gate)

Step-by-step derivation: freshness from time-delta

Let:

- t_{close} = market close timestamp for the relevant feed/instrument
- t_{now} = current processing timestamp
- $\Delta t = t_{\text{now}} - t_{\text{close}}$ (age)

You need a function mapping age → [0,1].

A common, stable choice is **exponential decay** with half-life H:

$$F(\Delta t) = \exp\left(-\ln(2) \cdot \frac{\Delta t}{H}\right)$$

Properties (step-by-step):

- If $\Delta t = 0$: $F(0) = e^0 = 1$ (perfectly fresh)
- If $\Delta t = H$: $F(H) = e^{-\ln 2} = 1/2$ (half as fresh)
- As $\Delta t \to \infty$: $F \to 0$

Freshness gate (reject stale data)

If the system rejects data below a threshold τ:

$$\text{Accept} \Leftrightarrow F(\Delta t) \geq \tau$$

Equivalently solve for max age allowed:

$$F(\Delta t) \geq \tau \Rightarrow e^{-\ln(2)\Delta t/H} \geq \tau \Rightarrow -\ln(2)\Delta t/H \geq \ln(\tau)$$
$$\Rightarrow \Delta t \leq -\frac{H}{\ln 2}\ln(\tau)$$

6.1. Microservices and Service Mesh

The proposed architecture adheres to a microservices style, with inferencing and auxiliary functions encapsulated as independent, self-sufficient services that share no state or

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

data. Service contracts and APIs use open standards (gRPC, REST, JSON) to support cross-platform service composition. Normalization and feature-pipeline services are implemented as stateless components that do not introduce state or side effects to the overall processing workflow, aside from the creation of specialized event records. Latency-sensitive computation and serving workloads are typically run on a dedicated cloud region with the lowest round-trip latency to trading venues; pipelines for order- and credit-line monitoring and reporting follow suit. Processing is triggered by multiple events through a cloud-native workflow system, with independent, distributed workloads for pricing and computation.
A service mesh abstracts much of the operational complexity of microservices deployments, directing network traffic through a dedicated data plane of sidecar proxies that automatically inject security policies (e.g., authentication, authorization, encryption) and visibility features (traffic monitoring, logging, tracing, alerting) within the data plane. Requests and responses are also automatically routed and authenticated via an API gateway that exposes a uniform API surface for external clients.

6.2. Data Lakehouse and Feature Stores

A lakehouse encapsulates the high-level properties required to govern a compliant data asset for multiple service consumers, and thus, is common across services. The data asset is organized as a semantically ground-truthed, versioned, and governed pool of both raw and processed data, residing in a cloud-native data storage layer that is optimized for both the speed of hot analytic queries (Parquet) and the cost of cold workloads (ORC, iceberg tables). Apache Hudi governs the lifecycle of raw data stored in the data lake such that hot data is kept available for cross-service exploration and avoided during cold-query planning. Support for external tools (PowerBI, Tableau) is enabled by semantic models for approved sets of assets. Transformed data is versioned into a structured source-of-truth model for all potential consumers and enabled with machine-readable quality and sensitivity policies along specific data axes.
Feature stores offer a real-time, privacy-compliant, and operationally resilient solution for powering ML-based functions along event-driven workflows for analytics-augmented transaction processing. The feature pipeline orchestrates the extraction of the features from the lakehouse, fuses them with real-time data in accordance with parameterizable slicing profiles, and serves pre-computed feature sets under hierarchical partitioning rules that automate data-flowing across the zones of a cloud-native data lake. It acts both as a source of truth for the features applied in model inference and model retraining, and serves features across both online and batch ML workloads for complex derivatives. Sensitivity analysis is integrated to guide governance policies and information controls on features used in production models.

7. Conclusion

The architecture presented enables the processing and storage of derivatives transactions and events in a cloud-native environment, moving beyond transaction security checking to support additional data-driven services and regulatory compliance. The next steps include scaling to production volumes, supporting the storage and retrieval of derivative features, implementing tools for visualizing the results of machine-learning capabilities, and integrating regulatory governance support. Furthermore, the scope of the architecture is being broadened to investigate additional cloud-native AI services for risk control and compliance checking.
The transition from transaction to compliance requires an integrated view, encompassing both the ongoing governance of transactions and the additional and historical data underpinning compliance checks. An additional challenge is that support services must be able to process transaction volumes of tens and hundreds of millions of transactions each day while also providing low-latency pricing for derivative instruments. By moving toward these

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

aims, a robust prototype is being developed to guide a validated production deployment.

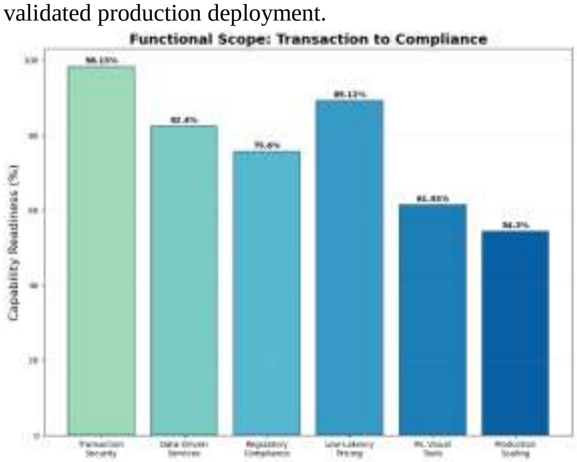


Fig 4: Functional Scope: Transaction to Compliance

7.1. Final Thoughts and Future Directions

Interest in cloud-native architectures for latency-sensitive operational processes—such as pricing, risk computation, and trade emission into transactions or clearing formats—has grown. The COVID-19 pandemic significantly increased demand for cloud-native solutions, revealing the considerable operational risks of highly concentrated, home-hosting operations for swap dealers. Such concentration exposes institutions to multiple correlated risks: the risk of losing the assets themselves, of being unable to meet their own operating and transactional requirements, and of being unable to support customers and counterparties. Solutions involving infrastructure and services hosted by third parties acting within a shared ethical framework that ensures compliance with market and data regulations can therefore facilitate compliance, scalability, and operational resilience in such latency-sensitive markets by distributing operational resources more equitably while also providing a more sustainable use of global resources.

The final thoughts and future directions consider a representative cloud-native AI architecture designed for a regulatory framework-compatible end-to-end processing pipeline for real-time derivatives transactions. Such a pipeline is required to enforce compliance with audit and traceability controls throughout the processing. Unplanned latency in the generation of features and labels, required for the successful serving of predictive machine-learning models, should also be kept within a feasible-boundary. The resulting architecture thus enables not only the end-to-end processing of transactions in real-time, but also the compliance assessment of the expected services within a well-understood latency tolerance.

8. References

1. Becker, M. (2020). RegTech—the application of modern information technology in regulatory affairs: Areas of interest in research and practice. Intelligent Systems in Accounting, Finance and Management, 27(4), 161–167.
2. Freij, Å. (2020). Using technology to support financial services regulatory compliance: Current applications and future prospects of RegTech. Journal of Investment Compliance, 21(2/3), 181–190.
3. Kolla, S. H. (2022). Strategic Information Integration Models for Cross-Functional Service Optimization in Large-Scale Enterprises. International Journal of Emerging Trends in Engineering and Management Research, 7(3), 11811.
4. Jullum, M., Løland, A., Huseby, R. B., Ånonsen, J., & Lorentzen, J. (2020). Detecting money laundering transactions with machine learning. Journal of Money Laundering Control, 23(1), 173–186.
5. Polak, P., Nelischer, C., Guo, H., & Yang, J. (2020). Intelligent fraud detection and prevention in financial services. Journal of Financial Crime, 27(2), 1–15.
6. von Solms, J. (2020). Integrating regulatory technology (RegTech) into the digital transformation of a bank treasury. Journal of Banking Regulation, 22, 191–207.
7. Arner, D. W., Buckley, R. P., Zetzsche, D. A., & Barberis, J. N. (2020). After the financial crisis: The

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

evolution of FinTech and RegTech. Georgetown Journal of International Law, 50(4), 1271–1319.
8. Carbonneau, A., & Godin, F. (2021). Equal risk pricing of derivatives with deep hedging. Quantitative Finance, 21(4), 593–608.
9. Ivașcu, C.-F. (2021). Option pricing using machine learning. Expert Systems with Applications, 163, 113799.
10. Mangalampalli, B. M. Generative AI Applications In Healthcare Data Mart Design And Optimization.
11. Jang, J. H., Yoon, J., Kim, J., Gu, J., & Kim, H. Y. (2021). DeepOption: A novel option pricing framework based on deep learning with fused distilled data from multiple parametric methods. Information Fusion, 70, 43–59.
12. Tiwari, S., Ramampiaro, H., & Langseth, H. (2021). Machine learning in financial market surveillance: A survey. IEEE Access, 9, 159734–159754.
13. Tsantekidou, E., Passalis, N., Tefas, A., & Kanniainen, J. (2021). Using machine learning to detect anomalous trading activity in financial markets. Expert Systems with Applications, 182, 115211.
14. Hadji Misheva, B., Jaggi, D., Posth, J.-A., Gramespacher, T., & Osterrieder, J. (2021). Audience-dependent explanations for AI-based risk management tools: A survey. Frontiers in Artificial Intelligence, 4, 794996.
15. Amistapuram, K. (2023). Privacy-Preserving Machine Learning Models for Sensitive Customer Data in Insurance Systems. Educational Administration: Theory and Practice, 29(4), 5950-5958.
16. Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). Explainable machine learning in credit risk management. Computational Economics, 57, 203–216.
17. Jullum, M., Løland, A., & Huseby, R. B. (2021). Predictive monitoring of financial transactions using machine learning. Journal of Money Laundering Control, 24(4), 1–15.
18. Jensen, R. I. T., & Iosifidis, A. (2022). Qualifying and raising anti-money laundering alarms with deep learning. Expert Systems with Applications, 208, 119037.
19. Ratku, A., & Neumann, D. (2022). Derivatives of feed-forward neural networks and their application in real-time market risk management. OR Spectrum, 44, 947–965.
20. Fritz-Morgenthal, S., Hein, B., & Papenbrock, J. (2022). Financial risk management and explainable, trustworthy, responsible AI. Frontiers in Artificial Intelligence, 5, 779799.
21. Chao, X., Ran, Q., Chen, J., Li, T., Qian, Q., & Ergu, D. (2022). Regulatory technology (Reg-Tech) in financial stability supervision: Taxonomy, key methods, applications and future directions. International Review of Financial Analysis, 80, 102023.
22. Grassi, L., & Lanfranchi, D. (2022). RegTech in public and private sectors: The nexus between data, technology and regulation. Journal of Industrial and Business Economics, 49, 441–479.
23. Papantoniou, A. A. (2022). RegTech: Steering the regulatory spaceship in the right direction? Journal of Banking and Financial Technology, 6, 1–16.
24. Currie, W. L., Seddon, J. J. J., & Van Vliet, B. (2022). From decision optimization to satisficing: Regulation of automated trading in the US financial markets. Information & Management, 59(8), 103721.
25. Liu, Q., Wang, C., Zhang, P., & Zheng, K. (2021). Detecting stock market manipulation via machine learning: Evidence from China Securities Regulatory Commission punishment cases. International Review of Financial Analysis, 78, 101887.
26. Zulkifley, M. A., et al. (2023). A survey on stock market manipulation detectors using artificial intelligence. Computers, Materials & Continua, 75(2), 4395–4418.
27. Weber, P., Carl, K. V., & Hinz, O. (2024). Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. Management Review Quarterly, 74, 867–907.

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

28. Pan, M., Li, D., Wu, H., & Lei, P. (2024). Technological revolution and regulatory innovation: How governmental artificial intelligence adoption matters for financial regulation intensity. International Review of Financial Analysis, 96, 103535.
29. Yandamuri, U. S. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. Zenodo.
30. Ratzan, J., & Rahman, N. (2023). Measuring responsible artificial intelligence (RAI) in banking: A valid and reliable instrument. AI and Ethics, 3, 1–15.
31. Georgosouli, A. (2023). Metarules, judgment and the algorithmic future of financial regulation in the UK. Oxford Journal of Legal Studies, 43(4), 777–803.
32. Neilson, B. (2023). Artificial intelligence authoring financial recommendations: Comparative Australian evidence. Journal of Financial Regulation, 9(2), 249–257.
33. Achakzai, M. A. K., & Peng, J. (2022). Using machine learning meta-classifiers to detect financial frauds. Finance Research Letters, 48, 102915.
34. Inala, R. Designing Scalable Technology Architectures for Customer Data in Group Insurance and Investment Platforms.
35. Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., & Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. Research in International Business and Finance, 62, 101744.
36. Singh, A., Jain, A., & Biable, S. E. (2022). Financial fraud detection approach based on Firefly optimization algorithm and support vector machine. Applied Computational Intelligence and Soft Computing, 2022, 1468015.
37. Achakzai, M. A. K., & Peng, J. (2023). Detecting financial statement fraud using dynamic ensemble machine learning. International Review of Financial Analysis, 89, 102827.
38. Talukder, M. A., Khalid, M., & Uddin, M. A. (2024). An integrated multistage ensemble machine learning model for fraudulent transaction detection. Journal of Big Data, 11, 168.
39. KollIntegration. South Eastern European Journal of Public Health, 248–260.
40. Hernandez Aros, L., Bustamante Molano, L. X., Gutierrez-Portela, F., Moreno Hernandez, J. J., & Rodríguez Barrero, M. S. (2024). Financial fraud detection through the application of machine learning techniques: A literature review. Humanities and Social Sciences Communications, 11, 1130.
41. Boulieris, P., Pavlopoulos, J., Xenos, A., & Vassalos, V. (2024). Fraud detection with natural language processing. Machine Learning, 113, 5087–5108.
42. Lokanan, M. E. (2024). Predicting money laundering using machine learning and artificial neural networks algorithms in banks. Journal of Applied Security Research, 19(1), 20–44.
43. Jang, J. H., Yoon, J., Kim, J., Gu, J., & Kim, H. Y. (2021). DeepOption: A novel option pricing framework based on deep learning with fused distilled data from multiple parametric methods. Information Fusion, 70, 43–59.
44. a, S. K., & Reddy, V. A. R. (2023). Deep Learning Architectures For Multimodal Medical Data
45. Song, C. (2022). Design and application of financial market option pricing system based on high-performance computing and deep reinforcement learning. Scientific Programming, 2022, 8525361.
46. Dang, Q. V. (2023). Option pricing with machine learning. Ho Chi Minh City University of Education Journal of Science, 20(2), 244–252.
47. Baule, R., & Shkel, D. (2021). Model risk and model choice in the case of barrier options and bonus certificates. Journal of Banking & Finance, 133, 106307.
48. Robisco, A. A., & Carbó Martínez, J. M. (2022). Measuring the model risk-adjusted performance of machine learning algorithms in credit default prediction. Financial Innovation, 8, 70.
49. Cosma, S., Rimo, G., & Torluccio, G. (2023). Knowledge mapping of model risk in banking.

JOURNAL OF INNOVATIVE ACADEMIC RESEARCH
ISSN : 3049-2122
VOLUME: 02 ISSUE: 01
RECEIVED: JANUARY 23 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 21 PUBLISHED: MARCH 07

International Review of Financial Analysis, 89, 102800.

50. Lazar, E., Qi, S., & Tunaru, R. (2024). Measures of model risk for continuous-time finance models. Journal of Financial Econometrics, 22(5), 1456–1481.
51. Weber, P., Carl, K. V., & Hinz, O. (2024). Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. Management Review Quarterly, 74, 867–907.
52. Simion, M., & Kelp, C. (2023). Trustworthy artificial intelligence. Asian Journal of Philosophy, 2, 8.