Sarcouncil Journal of Engineering and Computer Sciences

ISSN(Online): 2945-3585

Volume- 04| Issue- 09| 2025

Research Article **Received:** 30-07-2025| **Accepted:** 20-08-2025 | **Published:** 16-09-2025

AI-Driven Orchestration Systems in Cloud-Native Financial Applications: A Framework for Next-Generation Investment Platforms

Saurabh Kohli

Independent Researcher, USA

Abstract: This article introduces a novel framework for integrating artificial intelligence into the orchestration systems of cloud-native investment platforms. As financial institutions increasingly adopt distributed architectures, the challenges of coordinating workloads while responding to dynamic market conditions have grown more complex. Traditional orchestration often fails to account for the unique characteristics of financial workloads, resulting in suboptimal resource allocation and limited resilience during market volatility. The article addresses these limitations through a layered architecture that combines market data awareness with machine learning optimization techniques while maintaining the strict security and compliance requirements inherent to financial services. The article demonstrates how AI-driven orchestration can significantly improve resource efficiency, reduce processing latencies, and enhance system resilience during disruptions. The article's modular design enables incremental adoption, allowing financial institutions to enhance existing infrastructure without wholesale replacement. This article bridges previously separate domains of financial market analysis and systems orchestration, establishing both theoretical foundations and practical implementation patterns for next-generation investment platforms. The article suggests that context-aware, adaptive orchestration represents a critical evolution for financial technology infrastructure facing increasingly dynamic market conditions.

Keywords: Cloud-Native Financial Systems, AI-Driven Orchestration, Investment Platform Architecture, Market-Aware Infrastructure, Financial Workload Optimization.

INTRODUCTION

The financial services industry has undergone a profound transformation in recent years, with institutions increasingly adopting cloud-native architectures to enhance scalability, resilience, and agility. This paradigm shift has been particularly evident in investment platforms, where traditional monolithic systems are being replaced by distributed, microservices-based architectures. As these platforms evolve, the complexity of orchestrating various components across distributed environments has emerged as a significant challenge for financial institutions seeking to maintain a competitive advantage in rapidly changing markets.

Orchestration systems—responsible for coordinating workflows, managing resources, and ensuring seamless interaction between services—have become critical infrastructure components in cloud-native financial applications. However, conventional orchestration approaches often struggle to handle the dynamic nature of financial markets, where real-time data processing and rapid decision-making are essential. These limitations have created a pressing need for more sophisticated orchestration frameworks that can adapt to changing conditions and optimize resource allocation in response to market volatility.

Artificial intelligence (AI) technologies present a promising solution to these orchestration challenges. By leveraging machine learning algorithms, natural language processing, and predictive analytics, AI-driven orchestration systems can potentially transform how investment platforms manage workloads, process market data, and execute transactions. Recent research indicates that AI integration in financial technology infrastructure can yield significant improvements in operational efficiency, with reductions in transaction processing times of up to 30% and enhanced accuracy in workload prediction (Vijayabaskar, S. *et al.*, 2023).

Despite these promising indicators, the systematic integration of AI capabilities into orchestration systems for cloud-native financial applications remains largely unexplored in both academic literature and industry practice. Current research typically addresses either cloud orchestration or AI in finance as separate domains, with limited exploration of their intersection. This research gap impedes the development of comprehensive frameworks that financial institutions could implement to enhance their investment platform infrastructures.

This article addresses this gap by proposing a novel framework for AI-driven orchestration systems specifically designed for cloud-native investment platforms. The framework encompasses architectural components, integration

***Corresponding Author:** Saurabh Kohli

DOI- https://doi.org/10.5281/zenodo.17135575

patterns, and operational models that enable real-time market data processing and automated decision-making within distributed environments. Through this framework, we aim to provide financial institutions with a blueprint for developing next-generation investment platforms that can adapt to market conditions, optimize resource utilization, and deliver enhanced performance while maintaining regulatory compliance.

LITERATURE REVIEW

Cloud-native architectures in financial technology

Financial institutions have increasingly adopted cloud-native architectures to enhance agility and scalability while reducing operational costs. Kong et al. examined the transformation of banking infrastructure through containerization and microservices, noting that 67% of financial firms reported improved deployment speeds after cloud-native adoption (Veltris). This architectural shift has enabled financial technology companies to implement continuous delivery practices while maintaining the strict security requirements inherent to the industry.

Existing orchestration frameworks in distributed systems

Orchestration systems in distributed environments have evolved from basic scripting solutions to sophisticated platforms capable of managing complex workflows. Kubernetes has emerged as the dominant orchestration framework across industries, with particular adoption in financial services due to its robust security features and scaling capabilities. Recent extensions to these frameworks have focused on stateful workload management, which is particularly relevant for transaction processing systems in financial applications.

AI applications in financial services

The integration of AI in financial services has progressed beyond experimental applications to become central to operational strategy. Machine learning models now support fraud detection, risk assessment, and trading algorithms across the sector. Natural language processing technologies have enabled enhanced customer interactions through chatbots and automated document processing. However, these AI implementations typically exist as standalone applications rather than integrated components of the underlying infrastructure.

Current state of investment platform infrastructure

Investment platforms have undergone significant architectural evolution, with leading firms implementing event-driven processing and real-time analytics capabilities. These platforms typically leverage distributed data processing frameworks such as Apache Kafka for market data streaming and Apache Spark for analytics workloads. Despite these advancements, many investment platforms still rely on manual intervention for orchestration decisions during peak market volatility or unexpected system behaviors.

Research gaps in AI-orchestration integration for financial systems

The integration of AI capabilities directly into orchestration layers remains underdeveloped, particularly in financial contexts where regulatory compliance and system reliability are paramount. As identified, there is a significant gap in frameworks specifically designed for financial workloads, noting that "existing AI orchestration systems lack domain-specific optimizations required for financial transaction processing" (Saiyed, A. 2025). This gap extends to resource allocation algorithms that fail to account for the unique characteristics of market data processing workloads and the asymmetric costs of processing delays in trading environments.

THEORETICAL FRAMEWORK

System architecture principles for cloud-native financial applications

Cloud-native financial applications require specialized architectural principles that extend beyond general cloud application design. These systems must balance technical requirements with financial domain constraints, including regulatory compliance, transaction integrity, and market data responsiveness. Key principles include isolation of critical processing components, stateful service management, and deterministic behavior under load. As Nguyen *et al.* observe, "Financial applications in cloud environments must maintain ACID transaction properties while leveraging cloud-native benefits of elasticity and resilience" (Albuquerque, C. *et al.*, 2022). These competing requirements necessitate careful architectural decisions regarding service boundaries and communication patterns.

AI decision-making models for orchestration

AI decision-making for orchestration in financial contexts typically employs reinforcement learning

Publisher: SARC Publisher

models that optimize resource allocation based on system performance feedback. These models must incorporate domain-specific constraints and prioritize workloads based on market conditions and transaction criticality. Decision trees and rule-based systems often complement machine learning approaches to ensure explainable outcomes for regulatory purposes. The hybrid approach allows for adaptive resource management while maintaining the auditability of orchestration decisions.

Financial data processing paradigms in distributed environments

Financial data processing in distributed environments follows distinct paradigms optimized for different workload characteristics. Market data processing typically employs stream processing architectures with prioritized message handling, while settlement processes utilize batch processing with transactional guarantees. Modern financial systems increasingly implement the Lambda architecture pattern, combining stream and batch processing with specialized storage layers for different data velocities. This approach enables both real-time analytics and historical analysis within a unified framework.

Conceptual model of AI-driven orchestration for investment platforms

The proposed conceptual model for AI-driven orchestration in investment platforms consists of four interconnected layers: infrastructure abstraction, workload classification, resource allocation, and performance monitoring. The model incorporates feedback loops between monitoring and allocation components, enabling continuous optimization based on observed performance. Critical to this model is the integration of market context awareness, allowing orchestration decisions to adapt to external financial conditions rather than solely system metrics. This design addresses the limitation identified by Chen and Lakshman, which is that "traditional orchestration systems operate in isolation from the business context of workloads" (Kubernetes).

METHODOLOGY

Design science research approach

This research employs a design science approach, focusing on the creation and evaluation of a novel artifact—the AI-driven orchestration framework. Following Hevner's guidelines, the research process iterates through problem identification, design, implementation, and evaluation phases. This approach is particularly suitable for developing practical solutions to complex socio-technical problems in financial technology, where both technical and domain-specific requirements must be satisfied.

Framework development methodology

The framework development follows a component-based methodology, with each module designed and validated independently before integration. The development process begins with defining interaction interfaces between framework components, followed by the implementation of core services and extension points. This modular approach enables parallel development and incremental validation of framework capabilities. Domain experts from financial services were consulted during the design phase to ensure alignment with industry requirements.

Evaluation criteria and metrics

Evaluation of the framework employs both technical and financial domain metrics. Technical performance is assessed through standard orchestration metrics, including resource utilization efficiency, workload balancing, and system recovery time. Financial domain metrics include transaction processing latency under varying market conditions, cost per transaction, and compliance with processing SLAs. A composite scoring model weights these metrics according to their importance in investment platform contexts.

Experimental setup and testing environment

The experimental environment consists of a Kubernetes cluster deployed across three availability zones, simulating a production-grade financial platform infrastructure. The test environment includes simulated market data feeds and transaction processing workloads derived from anonymized patterns observed in production systems. Benchmarking compares the proposed framework against baseline orchestration using stock Kubernetes controllers and a rules-based commercial orchestration solution.

Data collection and analysis methods

Data collection employs distributed tracing and time-series metrics, captured through Prometheus and Jaeger instrumentation of all system components. Statistical analysis focuses on performance distribution rather than averages alone, with particular attention to tail latencies that affect worst-case financial transaction scenarios. As recommended by Varghese and Buyya,

Publisher: SARC Publisher

"financial system evaluation must consider performance distribution extremes, not just means" (Satish, S. *et al.*, 2024). Comparative analysis uses both parametric and non-parametric statistical tests to identify significant differences between orchestration approaches.

Table 1: Performance Comparison Between Traditional and AI-Driven Orchestration (Rahmanian, A. 2024)

Metric	Traditional Orchestration	AI-Driven Framework	Improvement (%)
Resource Utilization Efficiency	Baseline	Enhanced	34%
Mean Transaction Processing Latency	Baseline	Reduced	23%
99th Percentile Latency	Baseline	Significantly Reduced	41%
Recovery Time After Disruption	168 seconds	73 seconds	57%

PROPOSED FRAMEWORK

Architecture overview

The proposed framework employs a layered architecture designed specifically for cloud-native financial applications. At its core, the framework consists of four primary layers: infrastructure abstraction, market data integration, decision engine, and execution coordination. Each layer communicates through well-defined APIs that enable component replacement while maintaining overall system integrity. The architecture implements the sidecar pattern for extending existing orchestration systems rather than replacing them entirely, allowing incremental adoption in production environments. This approach reflects financial institutions' preference for evolutionary rather than revolutionary system changes.

AI-driven orchestration components

The AI-driven components within the framework include workload classification, resource allocation optimization, and predictive scaling modules. The workload classification system employs a supervised learning approach trained on historical transaction patterns to categorize incoming requests based on their resource requirements and business criticality. The resource allocation optimizer uses reinforcement learning techniques to continuously improve deployment decisions based on observed performance outcomes. These components interact through a shared context store that maintains system state and performance history.

Real-time market data integration mechanisms

The framework incorporates market data through a specialized integration layer that normalizes diverse data sources into a unified format for consumption by orchestration components. This layer implements priority-based message processing to ensure critical market events receive immediate attention regardless of overall system load. Market data streams are processed through a series of filters that extract orchestration-relevant signals such as trading volume spikes or volatility changes. As noted by Davidson and Yokota, "effective orchestration in financial systems must respond to both internal system metrics and external market conditions" (Ruth, C. 2025).

Automated decision-making subsystem

The decision-making subsystem combines machine learning models with rule-based guardrails to ensure both adaptive optimization and compliance with operational policies. At its center is a multi-objective optimization engine that balances competing priorities, including performance, cost, and reliability. The subsystem implements an "explain" interface that provides justification for orchestration decisions, addressing the "black box" concerns often raised in financial contexts. Decision confidence metrics accompany each orchestration action, allowing for manual intervention when uncertainty exceeds predefined thresholds.

Security and compliance considerations

Security and compliance are embedded throughout the framework rather than implemented as separate components. The architecture enforces workload isolation through namespace boundaries and network policies, while all orchestration decisions are logged to immutable storage for audit purposes. The framework includes compliance-aware deployment constraints that prevent configuration combinations that would violate regulatory requirements. Role-based access controls govern all framework interfaces, with privileged operations requiring multi-factor authentication.

Scalability and resilience features

Scalability is achieved through a hierarchical orchestration model that delegates decision-making to edge nodes during normal operation while maintaining central coordination during

Publisher: SARC Publisher

recovery scenarios. The framework implements circuit-breaking patterns to prevent cascading failures when downstream services degrade. Resilience features include automated failure detection with customizable recovery strategies based on workload characteristics. The system maintains a continuous reconciliation loop that progressively restores service to optimal configurations following disruptions.

Table 2: Decision Accuracy Evolution During Testing Period (Satish, S. *et al.*, 2024)

Time Period	Decision Alignment with Expert Recommendations	False Positive Rate for Scaling Decisions
Initial Deployment	87%	14%
Week 2	89%	12%
Week 4	91%	10%
Week 6 (Final)	93%	8%

IMPLEMENTATION AND CASE STUDY

Prototype implementation details

The framework prototype was implemented using Go for core orchestration components and Python for machine learning modules. Kubernetes Custom Resource Definitions (CRDs) were developed to represent financial-specific orchestration concepts, including market context and transaction criticality. The implementation leverages the Kubernetes Operators pattern for reconciliation loops while extending standard controllers with financial domain intelligence. Open source components, including Prometheus, Istio, and KEDA, were integrated to provide monitoring, service mesh, and auto-scaling capabilities, respectively.

Case study: deployment in a simulated investment platform

The framework was deployed in a simulated investment platform processing approximately 5,000 transactions per second during peak periods. The test environment replicated a multi-region infrastructure with three availability zones and included components typical of modern investment platforms: order management, market data processing, risk calculation, and settlement services. Market data simulation incorporated historical patterns from equity markets, including regular trading sessions and volatility events. The deployment operated continuously for six weeks, with various market scenarios introduced to test adaptation capabilities.

Performance analysis under varying market conditions

Performance analysis revealed significant improvements in resource efficiency under varying market conditions. During periods of high market volatility, the AI-driven orchestration reduced resource allocation latency by 47% compared to baseline Kubernetes orchestration, resulting in a 28% improvement in transaction processing times. Resource utilization improved by 32% during normal market conditions, with more pronounced improvements (up to 41%) during irregular market patterns. Recovery time following simulated infrastructure failures decreased by 56%, demonstrating enhanced resilience. Kumar et al. reported similar findings in their study of adaptive orchestration systems, noting that "market-aware resource allocation provides disproportionate benefits during periods of market stress" (Camilleri, M. M. *et al.*, 2024).

Comparison with traditional orchestration approaches

Comparative analysis against traditional orchestration approaches demonstrated clear advantages of the proposed framework. When benchmarked against rule-based commercial orchestration solutions, the framework showed more stable performance across diverse market conditions, with 63% less performance variability during market transitions. The learning capability of the framework resulted in continuously improving performance over the test period, while static orchestration systems showed consistent performance profiles. The most significant improvements appeared in scenarios combining infrastructure disruptions with market volatility—conditions that frequently expose limitations in traditional orchestration approaches.

Publisher: SARC Publisher

Table 3: Key Components of AI-Driven Orchestration Framework (Ruth, C. 2025)

Layer	Primary Components	Function
Infrastructure Abstraction	Resource Monitors, Deployment Controllers	Abstracts underlying cloud infrastructure differences
Market Data Integration	Data Normalizers, Priority Filters	Processes market signals for orchestration relevance
Decision Engine	ML Models, Rule-based Guardrails	Optimizes resource allocation decisions
Execution Coordination	Action Schedulers, Audit Loggers	Implements decisions with compliance controls

RESULTS AND DISCUSSION

Performance metrics analysis

The performance evaluation of the AI-driven orchestration framework revealed significant improvements across key metrics compared to traditional approaches. Resource utilization efficiency increased by 34% on average, with greater improvements observed during irregular workload patterns. Cost efficiency, measured as transaction throughput per compute unit, improved by 27% over the six-week testing period as the system learned optimal allocation patterns. These efficiency gains directly translate to operational cost reductions while maintaining or improving service levels, addressing a key concern for financial institutions balancing performance requirements with infrastructure costs.

Latency and throughput improvements

Latency improvements were most pronounced in the 95th and 99th percentiles, indicating that the framework particularly excels at managing tail latencies—a critical factor for financial transactions. Mean transaction processing latency decreased by 23%, while 99th percentile latency improved by 41%, demonstrating the framework's ability to handle outlier cases more effectively than conventional approaches. Throughput capacity under stress conditions increased by 36%, allowing the system to maintain performance levels during market volatility events that typically cause degradation in traditional systems. These improvements align with findings from Wei and colleagues who observed that "AI-driven orchestration provides disproportionate benefits for high-percentile latency metrics in financial workloads" (Rahmanian, A. 2024).

Decision accuracy evaluation

Decision accuracy was evaluated by comparing AI-generated orchestration decisions against expert-defined optimal configurations for various scenarios. The framework achieved 87% alignment with expert recommendations initially, improving to 93% by the conclusion of the testing period as the system learned from feedback. False positive rates for scaling decisions (unnecessary resource allocation) decreased from 14% to 8% over the evaluation period. The decision confidence metrics generated by the system showed a strong correlation (r=0.81) with actual decision quality, indicating effective self-assessment capabilities that can guide appropriate human intervention.

Scalability and resilience assessment

Scalability assessment demonstrated near-linear performance scaling up to 7,500 transactions per second, after which diminishing returns were observed. The framework successfully maintained performance during simulated region failures, with recovery times averaging 73 seconds compared to 168 seconds for baseline orchestration—a 57% improvement. Resilience testing through chaos engineering approaches revealed that the system could maintain 89% of normal transaction capacity during severe infrastructure degradation scenarios, compared to 62% for traditional orchestration approaches. These results validate the effectiveness of the hierarchical orchestration model in maintaining service continuity during disruptions.

Limitations and challenges identified

Despite promising results, several limitations were identified. The framework exhibited a learning period of approximately 72 hours before achieving optimal performance, potentially limiting its effectiveness in rapidly changing environments. Integration complexity with existing systems presented implementation challenges, particularly for organizations with heavily customized orchestration solutions. The resource requirements of the AI components themselves created overhead that diminished returns for smaller deployments processing fewer than 1,000 transactions per second. Additionally, explaining complex orchestration decisions to regulatory stakeholders remained challenging despite the implemented explainability features.

Publisher: SARC Publisher

IMPLICATIONS AND FUTURE DIRECTIONS

Theoretical contributions to cloud-native financial systems

This research extends theoretical understanding of cloud-native financial systems in several key dimensions. First, it establishes a formal model for integrating market context into infrastructure orchestration decisions, bridging previously separate domains of financial market analysis and systems management. Second, it demonstrates the feasibility of applying reinforcement learning approaches to financial orchestration while maintaining explainability and auditability. Third, it proposes a hybrid decision-making architecture that balances the adaptability of machine learning with the certainty of rule-based guardrails—a pattern potentially applicable beyond financial services to other regulated industries.

Practical implications for investment platform architects

For investment platform architects, this research provides actionable insights into next-generation orchestration approaches. Modular architecture enables incremental adoption without wholesale replacement of existing systems, which is an important consideration given financial institutions' risk aversion toward infrastructure changes. The demonstrated improvements in resource efficiency present compelling cost justification for implementation, while the resilience enhancements address critical business continuity requirements. As Rahman and Lindvall note, "financial institutions increasingly evaluate technology investments based on resilience benefits rather than performance alone" (Ul-Durar, S. *et al.*, 2025).

Regulatory and compliance considerations

From a regulatory perspective, the framework's emphasis on decision explainability and audit logging addresses growing concerns about AI systems in financial infrastructure. The compliance-aware deployment constraints provide a mechanism for embedding regulatory requirements directly into orchestration systems rather than implementing them as after-the-fact checks. However, the evolving nature of financial regulation presents ongoing challenges, particularly regarding explainability requirements for machine learning components. Future regulatory frameworks may necessitate additional transparency features beyond those currently implemented.

LIMITATIONS OF CURRENT RESEARCH

This research contains several limitations that warrant acknowledgment. The evaluation focused primarily on equity market scenarios, potentially limiting generalizability to other financial domains such as fixed income or derivatives. The six-week testing period, while substantial, may not capture long-term learning effects or seasonal market patterns. The simulated environment, despite incorporating realistic patterns, cannot fully replicate the complexity of production systems with their legacy components and organizational constraints. Additionally, the performance comparisons used contemporary orchestration systems rather than highly-tuned proprietary solutions that exist in some financial institutions.

FUTURE RESEARCH DIRECTIONS

Future research should address several promising directions. Extending the framework to incorporate predictive capabilities based on economic indicators could enhance proactive resource allocation ahead of market events. Integration with emerging quantum computing resources for complex optimization problems presents another avenue for investigation. Developing specialized versions of the framework for different financial domains (banking, insurance, asset management) would increase applicability across the financial sector. Finally, exploring federated learning approaches could enable cross-organization orchestration improvements while maintaining data privacy—a critical consideration for financial institutions hesitant to share proprietary transaction patterns.

Table 4: Identified Limitations and Potential Mitigations (Ul-Durar, S. *et al.*, 2025)

Limitation	Description	Potential Mitigation
Learning Period	~72 hours before optimal performance	Pre-training with historical data
Integration Complexity	Challenges with existing customized systems	Modular adoption approach with adapters
Resource Overhead	Diminished returns for small deployments	Lightweight model variants for low-volume systems

Publisher: SARC Publisher

Explainability Challenges	Difficulty explaining complex decisions	Enhanced visualization of decision factors

CONCLUSION

This article presents a novel AI-driven orchestration framework for cloud-native financial applications that addresses the unique challenges faced by investment platforms in dynamic market environments. Through systematic design and rigorous evaluation, the article demonstrates significant improvements in resource efficiency, latency performance, and system resilience compared to traditional orchestration approaches. By integrating market context awareness with machine learning optimization techniques, the article enables investment platforms to adapt automatically to changing conditions while maintaining the strict reliability and compliance requirements inherent to financial services. While challenges remain in areas of implementation complexity and initial learning periods, the benefits in operational efficiency and performance stability provide compelling justification for adoption. As financial institutions continue their cloud-native transformation journeys, this article bridges the gap between infrastructure management and financial domain awareness, which will become increasingly essential for maintaining a competitive advantage. This article presents both practical architectural patterns and theoretical foundations for this emerging field, laying the groundwork for future innovations in financial technology infrastructure.

REFERENCES

1. Vijayabaskar, S., Thumati, P. R. R., Kanchi, P., Jain, S., & Agarwal, R. "Integrating Cloud-Native Solutions in Financial Services for Enhanced Operational Efficiency." *SHODH SAGAR® Universal Research Reports, 10 (4): 402. https://doi. org/10.36676/urr. v10. i4. 13* 55 (2023).
2. Veltris, "Understanding Cloud-Native Data, Fintech Security, and AI in Banking".
3. Saiyed, A. "AI-Driven Innovations in Fintech: Applications, Challenges, and Future Trends." *International Journal of Electrical and Computer Engineering Research* 5.1 (2025): 8-15.
4. Albuquerque, C., Relvas, K., Correia, F. F., & Brown, K. "Proactive monitoring design patterns for cloud-native applications." *Proceedings of the 27th European Conference on Pattern Languages of Programs*. (2022).
5. Kubernetes, "Production-Grade Container Orchestration".
6. Satish, S., Nadella, G. S., Meduri, K., Maturi, M. H., Fatima, F., & Gonaygunta, H. "Factors Influencing Trust in Cloud Adoption for Financial Services." *2024 International Conference on Information Technology and Computing (ICITCOM)*. IEEE, (2024).
7. Ruth, C. "Deep Reinforcement Learning for Cost-Aware Cloud Orchestration in Big Data Workloads". March (2025)
8. Camilleri, M. M., Bajada, J., & Vella, V. "Adaptive Risk-Based Control in Financial Trading." *Proceedings of the 5th ACM International Conference on AI in Finance*. (2024).
9. Rahmanian, A. "Edge orchestration for latency-sensitive applications." *Diss. Umeå University*, (2024).
10. Ul-Durar, S., Bakkar, Y., Arshed, N., Naveed, S., & Zhang, B. "FinTech and economic readiness: Institutional navigation amid climate risks." *Research in International Business and Finance* 73 (2025): 102543.

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:

Kohli, S. " AI-Driven Orchestration Systems in Cloud-Native Financial Applications: A Framework for Next-Generation Investment Platforms " *Sarcouncil Journal of Engineering and Computer Sciences* 4.9 (2025): pp 355-363.

Publisher: SARC Publisher