Sarcouncil Journal of Engineering and Computer Sciences

ISSN(Online): 2945-3585

Volume- 04| Issue- 12| 2025

Research Article **Received:** 20-10-2025| **Accepted:** 25-11-2025 | **Published:** 04-12-2025

API-Level Fraud Detection in Financial Systems: Real-Time AI and Behavioral Analytics Integration

Krishna Seemanapalli

University Of North Texas, USA

Abstract: Financial organizations are using cloud-native API designs more and more to meet evolving customer demands while maintaining demanding safety and regulatory standards. This transformation helps banks, fintech firms, and insurers improve their operational scalability and resilience through the use of serverless computer models, microservices, and containerization. Hardware security module integration, API gateway security, and OAuth2 authentication, among other levels of security methods to safeguard sensitive transaction data, are used in contemporary financial service APIs. Strong consent management and audit capabilities are required in open banking systems where third-party integrations exist; hence, regulatory frameworks such as PCI DSS, GDPR, and PSD2 drive architectural choices. Real-time behavioral analysis and AI-driven automated decision-making systems change fraud detection at the API level. These technical innovations demonstrate how financial institutions may achieve operational excellence by means of wise API integration, balancing innovation with legal compliance and security demands in an ever more digital financial environment.

Keywords: Cloud-native APIs, financial services security, microservices architecture, regulatory compliance, fraud detection.

INTRODUCTION

Banking institutions have experienced a dramatic restructuring of their technological foundations, abandoning traditional centralized systems in favor of distributed microservices frameworks that utilize cloud-native APIs (Munnangi, V. 2025). This metamorphosis has completely redefined how financial organizations architect, implement and sustain their core technological ecosystems. The abandonment of conventional monolithic structures has enabled superior operational adaptability, enhanced system extensibility, and increased productivity while maintaining the stringent security measures and regulatory adherence standards critical within the financial domain (Dhandapani, A. 2025).

Modern financial technology architecture depends extensively on interface-centric design approaches, which establish application programming interfaces as the fundamental building blocks for system connectivity (Munnangi, V. 2025). These approaches emphasize APIs as crucial elements that enable seamless interaction between distributed services and external collaborations. The interface-centric philosophy demands that communication protocols be developed and established before the creation of supporting applications, thus ensuring consistent, thoroughly documented, and standardized interaction mechanisms across the complete technological ecosystem.

Related Work / Literature Review

Cloud-native API development incorporates extensive organizational and operational factors beyond mere technical deployment (Dhandapani, A. 2025). Financial enterprises implementing cloud-native strategies must balance the benefits of distributed frameworks with established regulatory supervision and risk management protocols. This balance requires a thorough assessment of data governance requirements, cross-border compliance mandates, and the fluid integration of existing systems with modern cloud platforms.

With each industry having distinct regulatory requirements, consumer expectations, and operational restrictions (Munnangi, V. 2025), cloud-native API deployment shows major variations across banking, finance, and insurance industries. Integrating cloud-native APIs with current core banking systems often presents problems for conventional banking institutions; hence, sophisticated middleware solutions and thorough data transformation methods are needed. Fintech companies, operating with little legacy system constraints, can implement whole cloud-native frameworks utilizing serverless computing, containerization, and microservices from the project starting point.

Problem Statement

Financial institutions face significant challenges in implementing secure, scalable, and compliant API architectures that can handle real-time fraud detection while maintaining regulatory adherence.

***Corresponding Author:** Krishna Seemanapalli

DOI- https://doi.org/10.5281/zenodo.17819614

Primary concerns include balancing security requirements with operational efficiency in distributed systems while ensuring regulatory compliance across multiple jurisdictions such as PCI DSS, GDPR, and PSD2. Organizations must simultaneously implement real-time fraud detection capabilities without impacting transaction processing performance, manage the complexity of integrating legacy banking systems with modern cloud-native architectures, and maintain comprehensive data privacy and consent management protocols in open banking environments. These interconnected challenges require sophisticated solutions that address technical, regulatory, and operational requirements concurrently.

Proposed Solution: Three-Tier API Architecture

To address these challenges, we propose a comprehensive three-tier API architecture framework that integrates multiple technological and operational layers. The first tier focuses on security and authentication, incorporating OAuth2 authentication with comprehensive token management, API gateway protection with advanced threat detection capabilities, Hardware Security Module integration for cryptographic operations, and zero-trust architecture implementation throughout the system. The second tier addresses compliance and governance requirements through regulatory compliance automation covering PCI DSS, GDPR, and PSD2 standards, third-party provider onboarding and monitoring systems, comprehensive customer consent management frameworks, and robust audit trail and documentation systems. The third tier encompasses intelligence and processing capabilities, featuring real-time AI-driven fraud detection algorithms, behavioral analytics for comprehensive risk assessment, automated decision-making systems, and advanced performance optimization with scalability controls. This integrated approach ensures that security, compliance, and operational intelligence work together seamlessly to create a robust financial API ecosystem.

ARCHITECTURE DESIGN AND IMPLEMENTATION

Security Framework Components

OAuth2 protocol execution within banking API ecosystems demands sophisticated access token oversight approaches that accommodate the distinct security challenges of financial transactions (Gbenle, T. P. *et al.,* 2022). Financial organizations must construct thorough access token lifecycle oversight processes that include token creation, authentication, refresh operations, and protected invalidation procedures.

Table 1: Cloud-Native API Security Framework Components (Gbenle, T. P. *et al.,* 2022; Cabrera-Gutiérrez, A. J. *et al.,* 2022)

Security Component	Implementation Method	Regulatory Compliance	Integration Complexity
OAuth2 Authentication	Token-based authorization	PCI DSS, GDPR	Medium
API Gateway Protection	Traffic filtering, rate limiting	PSD2, GDPR	High
Hardware Security Modules	Cryptographic key management	PCI DSS, FIPS 140-2	High
Zero-Trust Architecture	Continuous verification	SOX, GDPR	Very High
End-to-End Encryption	AES-256, TLS 1.3	PCI DSS, GDPR	Medium

Table 2: Regulatory Compliance Requirements Matrix (Pedroza, G. *et al.,* 2021; Premchand, A., & Choudhry, A. 2018)

Regulation	Scope	Key Requirements	API Implementation Impact
PCI DSS	Payment processing	Cardholder data protection	Secure API endpoints, encryption
GDPR	Data protection	Privacy by design, consent management	Data minimization, audit trails
PSD2	Open banking	Strong customer authentication	Third-party API access controls

Publisher: SARC Publisher

SOX	Financial reporting	Internal controls, audit trails	API logging, monitoring
Basel III	Banking supervision	Risk management, capital requirements	Risk scoring APIs, reporting

Domain-Driven Design Methodologies for Financial Enterprises

Domain-driven design approaches inside financial business architecture need thorough business domain modeling that synchronizes technical deployments with corporate goals (Farsi, H. *et al.*, 2021). Financial companies must build complex domain boundaries reflecting business capabilities, regulatory demands, and operating workflows specific to banking contexts. Domain-driven design approaches demand thorough business domain analysis, thorough bounded context identification, and detailed ubiquitous language development supporting effective communication between technical teams and business stakeholders.

Container Orchestration Leveraging Kubernetes in Regulated Contexts

Container orchestration leveraging Kubernetes within regulated contexts necessitates sophisticated deployment strategies accommodating stringent compliance requirements and security protocols (Ghosh, R. K., & Ghosh, H. 2023). Financial corporations must establish comprehensive container management frameworks encompassing security policy enforcement, resource allocation controls, and regulatory compliance monitoring across distributed container environments.

Table 3: Microservices Architecture Patterns for Financial Services (Ghosh, R. K., & Ghosh, H. 2023; Farsi, H. *et al.*, 2021)

Architecture Pattern	Use Case	Benefits	Implementation Complexity
Domain-Driven Design	Business capability alignment	Clear boundaries, team autonomy	Medium
Event-Driven Architecture	Real-time processing	Scalability, loose coupling	High
CQRS (Command Query Responsibility Segregation)	High-performance reads	Optimized queries, scalability	High
Saga Pattern	Distributed transactions	Consistency, fault tolerance	Very High
Circuit Breaker	Fault tolerance	System stability, graceful degradation	Medium

Service Mesh Establishment and Serverless Computing

Service mesh establishment within financial architecture frameworks facilitates sophisticated inter-service coordination patterns, enhancing security, observability, and reliability (Ghosh, R. K., & Ghosh, H. 2023). Financial corporations need to implement complete service mesh solutions that include extensive observability across distributed service environments, traffic management, and security policy enforcement. Service mesh establishment necessitates detailed configuration management, sophisticated security controls, and comprehensive monitoring capabilities supporting complex financial service coordination while maintaining regulatory compliance.

Serverless computing architectures within financial environments provide sophisticated resource allocation strategies accommodating variable transaction requirements while maintaining cost efficiency (Farsi, H. *et al.*, 2021). Financial corporations must establish comprehensive serverless architectures encompassing event-driven processing, automatic scaling capabilities, and sophisticated resource management across distributed computing environments.

Publisher: SARC Publisher

Governance, Security, and Lifecycle Management

API Gateway Defensive Structures and Attack Prevention

API gateways operate as fundamental security barrier points within financial service structures, implementing numerous protective mechanisms to defend against emergency cyberattacks (Gbenle, T. P. *et al.,* 2022). These gateways employ sophisticated attack recognition algorithms, traffic restriction protocols, and request examination mechanisms that detect and eliminate hostile activities before accessing core banking systems. The establishment of thorough documentation, monitoring, and alert systems within API gateways enables immediate attack evaluation and rapid incident management capabilities necessary for maintaining financial system dependability.

Hardware Security Module Integration Techniques

Hardware Security Modules represent crucial components in banking API security structures, providing tamper-proof environments for cryptographic operations and sensitive access token storage (Cabrera-Gutiérrez, A. J. *et al.*, 2022). Banking organizations integrate HSMs through various architectural techniques, including network-attached HSMs for distributed environments and embedded HSMs for high-performance transaction processing. HSM integration with API structures requires a detailed evaluation of performance optimization, backup mechanisms, and conformity to industry standards for cryptographic hardware usage.

Zero-Trust Structural Concepts in Banking Contexts

Zero-trust security models have become essential structures for banking API protection, eliminating assumed trust relationships and requiring constant verification of all system interactions (Cabrera-Gutiérrez, A. J. *et al.,* 2022). Banking organizations implementing zero-trust structures must establish detailed access controls, continuous authentication mechanisms, and thorough activity monitoring across all API endpoints. Zero-trust concept implementation requires fundamental alterations to network segmentation, identity oversight, and authorization processes that conform to banking regulatory requirements while maintaining operational effectiveness.

Complete Encryption Procedures for Confidential Banking Information

Complete encryption implementations within banking API systems require robust cryptographic procedures that protect confidential information throughout its complete lifecycle (Gbenle, T. P. *et al.,* 2022). Banking organizations implement advanced encryption standards that encompass information at rest, information in transit, and information in processing states, ensuring thorough protection against unauthorized access and information breaches. Complete encryption implementation requires a detailed evaluation of access token distribution mechanisms, cipher selection, and performance optimization to maintain transaction processing speed while ensuring maximum security coverage.

PCI DSS Directives for Payment Interface Processing

Payment Card Industry Data Security Standard directives within payment interface processing environments demand thorough conformance to strict security protocols governing cardholder information handling (Pedroza, G. *et al.,* 2021). Financial organizations must construct resilient information protection mechanisms encompassing protected information transmission, encrypted storage protocols, and access control measures specifically tailored for payment processing environments. PCI DSS directives necessitate continuous oversight of payment processing systems, periodic security evaluations, and comprehensive documentation of all cardholder information interactions within interface frameworks.

GDPR Information Safeguarding and Privacy-by-Design Establishment

General Data Protection Regulation conformance within banking interface systems requires privacy-by-design establishment that incorporates information-safeguarding principles directly into system architecture (Pedroza, G. *et al.,* 2021). Financial organizations must establish comprehensive information-safeguarding strategies encompassing information minimization principles, purpose limitation controls, and transparent information processing procedures. GDPR establishment necessitates sophisticated consent management mechanisms, information subject rights fulfillment systems, and comprehensive audit trails demonstrating ongoing conformance with European information safeguarding regulations.

Publisher: SARC Publisher

PSD2 Alignment for Open Banking Ventures

Within open banking projects, the alignment of the Payment Services Directive 2 requires the thorough implementation of robust customer authentication systems and safe interface access techniques (Premchand, A., & Choudhry, A. 2018). Financial institutions must create sturdy third-party access controls, transaction monitoring systems, and customer authentication processes matching European banking laws. PSD2 alignment calls for thorough risk assessment frameworks, fraud prevention tools, and thorough regulatory reporting systems showing continuous conformance with open banking requirements.

Implementation Framework and Tool Ecosystem

The implementation framework encompasses comprehensive toolsets for container orchestration, service mesh management, and monitoring capabilities. Kubernetes orchestration demands detailed configuration management, comprehensive monitoring capabilities, and sophisticated security controls coordinating with financial industry regulations while maintaining operational efficiency.

Auto-scaling solutions within financial architecture frameworks provide sophisticated resource allocation strategies accommodating transaction volume variations while maintaining performance standards (Ghosh, R. K., & Ghosh, H. 2023). Financial corporations must deploy comprehensive auto-scaling solutions encompassing predictive scaling algorithms, resource allocation controls, and sophisticated performance monitoring across distributed computing environments. Auto-scaling solutions necessitate detailed configuration management, comprehensive monitoring capabilities, and sophisticated resource optimization strategies supporting variable financial service requirements while maintaining cost efficiency.

Circuit breaker implementations within financial architecture frameworks provide sophisticated fault tolerance mechanisms, maintaining system durability during service failures (Farsi, H. *et al.*, 2021). Financial corporations must establish comprehensive circuit breaker deployments encompassing failure detection algorithms, fallback mechanisms, and sophisticated recovery strategies across distributed service environments. Circuit breaker implementations demand detailed failure analysis capabilities, comprehensive monitoring systems, and sophisticated recovery mechanisms protecting financial services from cascading failures while maintaining operational continuity.

Real-time data management within financial environments necessitates sophisticated streaming solutions accommodating high-velocity data flows while maintaining accuracy and reliability (Ghosh, R. K., & Ghosh, H. 2023). Financial corporations must deploy comprehensive streaming solutions encompassing event processing engines, data transformation pipelines, and sophisticated analytics capabilities across distributed data environments. Real-time data management demands detailed stream management capabilities, comprehensive monitoring systems, and sophisticated data quality controls supporting financial service requirements while maintaining regulatory compliance.

METHODOLOGY

Investigation techniques and case selection parameters focus on comprehensive organizational and operational factors beyond mere technical deployment (Dhandapani, A. 2025). The methodology encompasses several critical evaluation areas that provide comprehensive assessment capabilities for cloud-native API implementations in financial environments.

Processing volume performance indicators serve as fundamental measures of system capacity and extensibility, illustrating the ability of API frameworks to handle maximum transaction loads while preserving optimal response times and system reliability (Dhandapani, A. 2025). These indicators must accommodate diverse transaction types processed by financial organizations, including payment processing, account management, loan applications, and regulatory reporting operations.

Fraud identification precision measurements represent fundamental success indicators for financial API deployments, directly affecting customer trust, regulatory compliance, and operational risk management approaches (Munnangi, V. 2025). Developing baseline precision measurements allows financial institutions to evaluate the performance of integrated artificial intelligence and machine learning algorithms within their API structures. These measurements must establish an optimal equilibrium between fraud prevention effectiveness and customer experience quality,

Publisher: SARC Publisher

ensuring authentic transactions avoid improper identification or processing interruptions.

Regulatory conformity assessment systems provide essential measurements for determining how well cloud-native API deployments comply with regulations (Dhandapani, A. 2025). These systems have to record compliance with several legal standards, including financial services monitoring, sector-specific criteria, and data protection rules. Assessing legal compliance goes beyond simple checklist completion to include ongoing monitoring, maintenance of audit documents, and incident management systems.

EMPIRICAL EVALUATION / CASE STUDIES

AI Model Performance in Financial APIs

Table 4: AI Model Performance Metrics in Financial APIs (Niu, K. *et al.*, 2016; Zhang, J. *et al.*, 2017)

AI Model Type	Primary Function	Performance Metric	Regulatory Consideration
Fraud Detection	Transaction monitoring	Precision, recall, F1-score	Explainability requirements
Risk Scoring	Customer assessment	ROC-AUC, accuracy	Bias detection, fairness
Anomaly Detection	Behavioral analysis	False positive rate	Model interpretability
Decision Trees	Rule-based decisions	Accuracy, interpretability	Audit trail compliance
Neural Networks	Pattern recognition	Accuracy, processing speed	Black box concerns

Real-time Fraud Detection Implementation

Building real-time fraud detection algorithms inside financial interfaces calls for sophisticated machine learning methods ingrained directly into API layers (Niu, K. *et al.*, 2016). Financial institutions need to create sophisticated algorithmic systems incorporating real-time decision-making processes operating smoothly inside distributed API architectures, pattern recognition skills, and behavioral analysis techniques. Real-time fraud detection algorithm creation calls for thorough data preprocessing pipelines, sophisticated feature engineering methods, and advanced algorithm inference mechanisms harmonizing with high-frequency transaction processing responsibilities while maintaining minimal latency impacts.

Anomaly Recognition Procedures for Transaction Monitoring

Anomaly recognition procedures within financial transaction monitoring environments demand sophisticated pattern recognition algorithms capable of identifying irregular behavioral patterns across diverse transaction types (Niu, K. *et al.*, 2016). Financial corporations must establish comprehensive anomaly recognition structures encompassing statistical analysis procedures, machine learning algorithms, and behavioral modeling capabilities functioning continuously within API-driven transaction processing systems. Anomaly recognition procedures require detailed baseline establishment processes, sophisticated threshold management mechanisms, and comprehensive alert generation systems supporting financial service monitoring duties while maintaining operational efficiency.

Behavioral Analytics for Risk Judgment

Behavioral analytics within financial risk judgment environments demand sophisticated analytical structures capable of assessing customer behavior patterns and risk indicators (Zhang, J. *et al.*, 2017). Financial corporations must create comprehensive behavioral analytics solutions encompassing customer profiling algorithms, risk scoring mechanisms, and predictive modeling capabilities integrated within API-driven decision-making systems. Behavioral analytics require detailed customer behavior modeling processes, sophisticated risk judgment algorithms, and comprehensive scoring mechanisms supporting financial service risk judgment duties while maintaining regulatory conformance.

Algorithm Training and Continuous Improvement Workflows

Algorithm training and continuous improvement workflows within financial machine-learning environments demand sophisticated training pipelines capable of adapting to evolving fraud patterns and transaction behaviors (Niu, K. *et al.*, 2016). Financial corporations must establish comprehensive algorithm training structures encompassing data collection processes, feature engineering pipelines, and algorithm validation mechanisms functioning continuously within API-driven learning systems. Algorithm training and continuous improvement workflows require detailed training data management processes, sophisticated validation mechanisms, and comprehensive algorithm deployment strategies supporting financial service machine learning duties while maintaining performance standards.

Publisher: SARC Publisher

Harmony Between Rule Engines and Machine Learning Methods

Sophisticated integration structures that can blend rule-based reasoning with predictive analytics are necessary for rule engine alignment with machine learning processes in financial decision-making settings (Zhang, J. *et al.*, 2017). Financial firms need to develop complete rule engine solutions that work flawlessly with API-driven decision systems. These solutions should include business logic management, machine learning procedure integration, and decision orchestration capabilities. In order to support financial service automation tasks while preserving operational flexibility, rule engine alignment necessitates intricate business rule administration procedures, advanced procedure integration mechanisms, and extensive decision orchestration structures.

A/B Testing Strategies for Algorithm Improvement

A/B testing strategies within financial algorithm improvement environments demand sophisticated testing structures capable of evaluating algorithm performance and optimizing decision-making processes (Zhang, J. *et al.*, 2017). Financial corporations must create comprehensive A/B testing solutions encompassing experiment design processes, statistical analysis mechanisms, and performance evaluation capabilities integrated within API-driven testing systems. A/B testing strategies require detailed experiment design processes, sophisticated statistical analysis mechanisms, and comprehensive performance evaluation structures supporting financial service algorithm optimization duties while maintaining operational continuity.

Explainable AI for Duties Related to Auditing and Regulatory Compliance

In financial regulatory compliance settings, explainable AI necessitates complex interpretability frameworks that may offer clear justifications for regulatory compliance and audit duties (Niu, K. *et al.*, 2016). In order to comply with API-driven compliance systems, financial firms need to implement complete, explainable AI solutions that include audit trail-generating capabilities, decision explanation mechanisms, and algorithm interpretability procedures. Explainable AI necessitates thorough audit documentation structures, complex explanation generation mechanisms, and interpretability algorithm development methods that meet financial service regulatory responsibilities while preserving decision transparency.

DISCUSSION

The integration of cloud-native API strategies with advanced security frameworks and AI-driven fraud detection systems demonstrates significant potential for financial institutions. The three-tier architecture approach provides a comprehensive framework that addresses the complex requirements of modern financial services while maintaining regulatory compliance.

The implementation reveals that multi-layered security approaches effectively mitigate emerging cyber threats while regulatory compliance automation reduces operational overhead substantially. Real-time AI integration enhances fraud detection capabilities without compromising transaction processing efficiency, and microservices architecture patterns enable scalable and resilient financial systems that can adapt to varying operational demands. The combination of domain-driven design, container orchestration, and AI-powered decision-making creates a robust foundation for digital transformation in financial services, enabling institutions to balance innovation with regulatory requirements effectively.

Cross-jurisdictional regulatory coordination causes significant problems for financial institutions running across several regulatory systems (Premchand, A., & Choudhry, A. 2018). Financial organizations must negotiate complex regulatory settings, including differing information security demands, banking supervision norms, and consumer protection rules. The proposed architecture addresses these challenges through automated compliance mechanisms and standardized interface protocols that can adapt to varying regulatory requirements while maintaining operational consistency.

Interface harmonization across financial establishments calls for thorough systems guaranteeing interoperability while upholding security and conformity standards (Premchand, A., & Choudhry, A. 2018). The three-tier architecture facilitates this harmonization through standardized API specifications, consistent authentication mechanisms, and unified data formats that enable seamless integration among financial service providers while maintaining security and regulatory compliance standards.

Limitations and Future Work

While the proposed three-tier architecture demonstrates significant benefits, several

Publisher: SARC Publisher

limitations and areas for future research warrant consideration. Implementation complexity may pose substantial challenges for smaller financial institutions that lack the technical resources and infrastructure necessary for comprehensive deployment. Cross-jurisdictional regulatory compliance requires ongoing adaptation as regulatory frameworks continue to evolve across different markets and jurisdictions. AI model explainability remains a persistent challenge for regulatory acceptance, particularly in environments requiring transparent decision-making processes. Additionally, integration with legacy systems may require significant infrastructure investment and extended migration timelines.

Future research opportunities encompass several critical areas including the development of standardized implementation frameworks specifically designed for smaller institutions with limited resources. Enhanced AI explainability mechanisms represent a crucial area for advancing regulatory compliance and transparency in automated decision-making systems. Cross-border regulatory harmonization initiatives could significantly reduce compliance complexity for multinational financial institutions. Advanced threat detection capabilities using quantum-resistant cryptographic methods will become increasingly important as quantum computing capabilities advance. Finally, the integration of blockchain technologies for creating immutable audit trails presents promising opportunities for enhancing transparency and regulatory compliance in financial API systems.

Disaster recovery and business continuity blueprints within financial and architectural frameworks call for advanced backup techniques and business continuity tools to guarantee operating lifespan (Farsi, H. *et al.*, 2021). Future work should address comprehensive disaster recovery plans, including data backup policies, system replication approaches, and sophisticated recovery techniques in geographically distributed computing environments. These areas require detailed recovery planning, extensive testing methods, and advanced business continuity plans protecting financial services from operational disruptions while maintaining regulatory compliance.

CONCLUSION

Financial service companies have found that cloud-native API approaches provide important advantages when managing security, compliance, and scalability challenges in current digital environments. Banks and other financial organizations achieve strong operational results while meeting regulatory requirements through sophisticated security frameworks, compliance procedures, and expandable cloud architectures.

Security measures with multiple layers, such as OAuth2 login systems, API gateway shields, and special hardware modules, give strong protection against new online dangers. Rules like PCI DSS, GDPR, and PSD2 help financial companies stay legal while creating new services. Building blocks like small services, container management, and serverless systems help companies adjust resources quickly and stay stable.

Computer programs that learn can spot fraud better by watching how people behave and making quick decisions automatically. Financial companies can reach their business goals by using these API methods that put security, following rules, and growing operations first. Banks get better results when they combine these technology tools with security rules and growth plans into one digital change program. When cloud technology, rule-following systems, and smart computer programs work together, financial companies can serve customers better while keeping the strong risk control needed for long-term success in busy financial markets.

REFERENCES

1. Munnangi, V. "Cloud-Native API Strategies for Financial Services: Ensuring Security, Compliance, and Scalability." *European Journal of Computer Science and Information Technology* 13.15 (2025): 10-37745.
2. Dhandapani, A. "Microservices Architecture in Financial Services: Enabling Real-Time Transaction Processing and Enhanced Scalability." European Journal of Computer Science and Information Technology (EJCSIT), vol. 13, issue 32, May 31, (2025).
3. Gbenle, T. P., Abayomi, A. A., Uzoka, A. C., Ogeawuchi, J. C., Adanigbo, O. S., & Odofin, O. T. "Applying OAuth2 and JWT Protocols in Securing Distributed API Gateways: Best Practices and Case Review." (2022).
4. Cabrera-Gutiérrez, A. J., Castillo, E., Escobar-Molero, A., Álvarez-Bermejo, J. A., Morales, D. P., & Parrilla, L. "Integration of hardware security modules and permissioned blockchain in industrial iot networks." *IEEE Access* 10 (2022): 114331-114345.

Publisher: SARC Publisher

5. Pedroza, G., Muntes-Mulero, V., Martin, Y. S., & Mockly, G. "A model-based approach to realize privacy and data protection by design." *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, (2021).
6. Premchand, A., & Choudhry, A. "Open banking & APIs for transformation in banking." *2018 international conference on communication, computing and internet of things (IC3IoT)*. IEEE, (2018).
7. Ghosh, R. K., & Ghosh, H. "Microservices, Containerization, and MPI." *Distributed Systems: Theory and Applications, Wiley–IEEE Press*, (2023).
8. Farsi, H., Allaki, D., En-Nouaary, A., & Dahchour, M. "Following Domain Driven Design principles for Microservices decomposition: is it enough?." *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, (2021).
9. Niu, K., Jiao, H., Deng, N., & Gao, Z. "A real-time fraud detection algorithm based on intelligent scoring for the telecom industry." *2016 International Conference on Networking and Network Applications (NaNA)*. IEEE, (2016).
10. Zhang, J., Yang, J., & Li, J. "When rule engine meets big data: Design and implementation of a distributed rule engine using spark." *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, (2017).

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

Cite this article as:
Seemanapalli, K. "API-Level Fraud Detection in Financial Systems: Real-Time AI and Behavioral Analytics Integration." *Sarcouncil Journal of Engineering and Computer Sciences* 4.12 (2025): pp 27-35.

Publisher: SARC Publisher