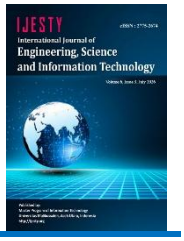
International Journal of Engineering, Science and Information Technology
Volume 6 No. 3 (2026) pp. 1-14
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v6i3.1823
Research Paper, Short Communication, Review, Technical Paper



Transforming Real-Time Payment Infrastructure with Cloud-Native Architecture

Priyatham Nagaiya Seenu Naidu*

Independent Researcher, United State

**Corresponding author Email: pnagaiyaseenunaidu@gmail.com*

The manuscript was received on 25 November 2025, revised on 2 January 2026, and accepted on 1 April 2026, date of publication 2 June 2026

Abstract

The global financial services sector is undergoing a profound transformation driven by the rapid adoption of digital payment technologies, increasing customer expectations for real-time transactions, and the growing demand for scalable and secure financial infrastructures. Traditional monolithic payment systems often struggle to meet these requirements due to limitations in scalability, operational flexibility, fault tolerance, and integration capabilities. Although cloud-native technologies have emerged as a promising solution for modern payment ecosystems, the scholarly literature has not adequately examined how cloud-native architectural patterns address enterprise integration, performance scalability, resilience, and regulatory compliance challenges in large-scale payment environments. This article presents a systematic analysis of cloud-native payment infrastructure by synthesizing architectural patterns, performance benchmarks, and industry case studies from both developed and emerging digital payment ecosystems. The findings demonstrate that event-driven cloud-native architectures can deliver substantial performance improvements, including a 94% reduction in transaction processing latency from 2,400 milliseconds to 128 milliseconds, near real-time fraud detection with response times of approximately 0.05 seconds, and fraud identification performance achieving 99% recall and 83% precision through hybrid deep learning models. Furthermore, production-scale validation is evidenced by successful implementations such as India's Unified Payments Interface (UPI), which processes over 10 billion transactions monthly, and Brazil's PIX platform, capable of handling up to 200,000 transactions per second. Security and compliance analyses reveal that immutable event-sourcing mechanisms, combined with comprehensive audit trails and encryption strategies, facilitate adherence to major regulatory frameworks, including PCI-DSS, GDPR, and anti-money laundering requirements. However, significant implementation challenges remain, particularly concerning legacy system integration, cross-jurisdictional regulatory fragmentation, organizational transformation, and shortages of cloud-native engineering expertise. Based on these findings, practical implementation guidelines emphasize microservice-based architectures, end-to-end event-driven processing, and zero-trust security principles. Future research directions include post-quantum cryptographic migration strategies, central bank digital currency interoperability standards, sustainable payment infrastructure design, and the application of artificial intelligence for autonomous financial operations and risk management.

Keywords: *Cloud-native Architecture, Real-time Payment Systems, Microservices, Payment Infrastructure Security, Fraud Detection*.

1. Introduction

The international financial services landscape is experiencing a structural shift of primary nature, driven by the synthesis of digital payment platforms, regulatory liberalization, and businesses' adoption of cloud infrastructure. In 2023, the mobile payments industry is currently valued at over USD 1.6 trillion. It is expected to grow at a compound annual growth rate (CAGR) of 30 percent through 2030. This trend indicates a clear, unalterable shift in how people, businesses, and governments conduct their transactions [1]. This growth is not limited to the already existing markets. In Oman, the Central Bank reported an increase in mobile payment transactions of 551 percent between 2021 and 2022 alone, with the total number of digital payment transactions increasing by 37.8 percent and the aggregate value of digital financial infrastructure payment transactions increasing by 13.1 percent over the same period, which indicates that even the emerging and resource-dependent economies are rapidly moving to real-time digital financial infrastructure [1]. In 2023, mobile wallet platforms (particularly Apple Pay and Google Pay) made 41 percent of all e-commerce payments around the world, reflecting the central role they play in transforming the scheme of enterprise payment systems, moving them out of terminal-based, batch-processing models and into the context of real-time, 24/7 transactions [1], [2].

This pressure for growth has also revealed weaknesses in the structure of traditional banking buildings. Banking used to be a closed, pipeline-based institution that had preserved a monopoly over consumer transaction data, leveraging its ownership of this data as a competitive advantage rather than as a process to create value [3]. The Coase theory suggests that, in the absence of transaction costs, data ownership would be immaterial to efficiency, as ownership would automatically shift to the party that could derive the highest value. In practice, real-life conflicts imply that the control over data in the financial services sector has created a material incentive for banks to

Copyright © Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

retain it, thereby stifling competition and innovation across the financial services industry as a whole [3], [4]. This pipeline model has been unbundled, with regulatory actions continuing to accelerate such unbundling into the late 2010s (with regulatory interventions such as the European Payment Services Directive (PSD2), the Payment System Blueprint 2025 (BSPI 2025) in Indonesia, and similar national frameworks elsewhere in the Asia-Pacific region), fundamentally reorganizing the competitive and architectural environment of enterprise payment infrastructure. The paradigm governing this new architecture is open banking. This established framework enables banking consumers to safely disclose their transactional data to authorized third parties via standardized application programming interfaces (APIs), thereby enhancing competition, financial inclusion, and the integration of payment systems into broader enterprise information contexts [3].

Current real-time payment systems fall into three interdependent typologies. Public payment networks are country-regulated, instant payment networks run by a central bank or a regulated clearing authority, offering universal access to real-time settlement to all qualified participants in a jurisdiction. Proprietary enterprise hubs are called private payment networks, and individual financial institutions or technology companies run these networks, enabling them to execute high-performance transactions under control within specific partner ecosystems. Consortia and hybrid networks are models of collaboration that involve several institutions sharing infrastructure on a governance basis, enabling interoperability across competitive boundaries. The individual typologies are based on specific enterprise needs, but each depends on the cloud-native API architecture as the facilitation layer to provide scalability, integration, and regulatory compliance.

The emergence of open banking has radically changed the enterprise environment for these types of infrastructure. The open banking market is projected to generate USD 28,894.7 million in revenue by 2030, with a CAGR of 28.7 percent from 2024 onward. Rahadian (2025) documents that Indonesia is one of the main drivers of growth, a country of 273.5 million people (17,000 islands), with 92 million unbanked and 46 million underbanked. BSPI 2025 is one of five national requirements for digital infrastructure, as the structural separation of payment data and the end of banking monopolies are seen as prerequisites for enabling financial inclusion at scale [3]. Simultaneously, Al Ghunaimi et al. (2025) show that the Vision 2040 program of the Republic of Oman clearly positions the real-time digital payment infrastructure as a strategic pillar of the economy's diversification. The government actively helps introduce FinTech to small and medium-sized enterprises (SMEs) as a tool for transitioning from a hydrocarbon-dependent phase to a digitally oriented, inclusive economy.

Although these transformation trends have been widely documented in global and regional payment infrastructures, there remains an apparent gap in the scholarly literature for an integrated, comprehensive analysis of cloud-native architectural patterns for managing the full range of enterprise integration, scalability, and security challenges posed by real-time payment systems. According to Al Ghunaimi et al. (2025), the barriers encountered by SMEs in switching to mobile payment platforms, such as high transaction costs, authentication challenges, lack of cybersecurity, and insufficient digital infrastructure, remain major challenges to date, and this is so because payment system architectures have not been systematically designed to fulfill the needs of enterprise settings of different scales and technical maturity. Open banking's theoretical advantages in competition, inclusion, and innovation are similarly documented by Rahadian (2025) but limited in practice by the challenge of implementation, including threats to cybersecurity, complexity in regulatory compliance, and lack of consumer education, all ultimately to be addressed and governed as architecture and governance issues more than policy. These gaps are filled in this review. The research question is as follows: How might cloud-native architecture make real-time payment infrastructure more scalable, resilient, and secure, and overcome challenges in enterprise system integration, regulatory compliance, and cross-border interoperability?

This research paper uses a systematic review of peer-reviewed journals based on IEEE Xplore, ACM Digital Library, ScienceDirect, Scopus, and Web of Science search engines to address this question, using the following search strings: "cloud-native AND payment systems," "real-time payments AND microservices," and "payment infrastructure AND enterprise architecture." The review includes the period 2018-2025, with a focus on the increase in publications after 2021, which reflects the activity of the post-pandemic shift towards the acceleration of the adoption of digital payments, which is confirmed by the increase in transaction volumes in Oman with mobile transactions by 551 percent and regulatory support of the open banking model in the Asia-Pacific, the Gulf Cooperation Council, and the European markets [1], [3]. Peer-reviewed journal articles and conference proceedings that directly related to enterprise payment infrastructure, cloud architecture, or financial system security were used as inclusion criteria. White papers, patents, and non-peer-reviewed industry publications were out. Quality appraisal was used to determine methodological rigor, technical specificity, and the relevance of enterprise information systems.

Architectures in domains apart from cloud-native payments can benefit from similar innovations. Alshiky et al. (2025) combine DSDN and blockchain to provide peer-to-peer streaming. Compared to the best alternative using individual technologies, their approach exhibits 8 to 9% higher throughput, 5 to 6% lower latency, and 7 to 21% lower energy consumption on 50 nodes. The use of a two-layer structured state machine with a control plane for real-time online orchestration and a trust plane for decentralized security follows the payment system requirements of low-latency transaction processing and tamper-proof post-hoc auditability. In this context, the NS-3 simulation by Alshiky et al., with varying node density and IDS/ACL modules, serves as the benchmark for performance evaluation in Sections 3 and 5 [5].

To maintain the same level of consistency within the current article, the following constructs are established. Real-time payments are those sent, transmitted, cleared, and settled in seconds on a 24/7/365 system. Cloud-native architecture is a design paradigm that builds applications as microservices, each independently deployable, running in containerized environments and communicating via APIs and asynchronous messaging. Enterprise integration refers to integrating payment infrastructure with enterprise resource planning, treasury management, core banking, and customer relationship management systems. Transaction throughput is the count of transactions made per second under sustained load. Security resilience includes fraud prevention, zero-trust access control, compliance with regulatory frameworks such as PCI-DSS and GDPR, and fault-tolerant system design—the regulatory layer that Rahadian (2025) names as a key challenge in the implementation of open banking and that Al Ghunaimi et al. (2025) confirm is the main security barrier to the adoption of mobile payment platforms by the SMEs in emerging markets.

The rest of this article proceeds as follows. The second section will look at the building blocks and architectural designs of cloud-native payment infrastructure and how they are used with enterprise systems. Section 3 measures performance benefits through benchmarking and industry case studies. Section 4 deals with API-based integration and the open banking regulatory framework—the area Rahadian (2022) defines as the regulatory framework through which data sharing between payment infrastructure and enterprise platforms occurs. Section 5 examines security architecture, fraud detection, and resilience engineering. Section 6 compares cloud platforms, orchestration

technologies, and international payment networks. Section 7 discusses industry-specific applications across banking, retail, healthcare, logistics, and government. Section 8 outlines future research directions, and Section 9 provides the conclusion.

2. Literature Review

2. Cloud-Native Architecture in Payment Systems

2.1. Foundational Concepts and Core Components

The technological change in enterprise financial systems this decade is the shift to cloud-native payment infrastructure and the end of monolithic banking, which relies on branches. The classic core banking designs were built on old-fashioned batch-processing models that were specifically designed to match a business model that was half a century old, with periodic settlement cycles, low transaction volumes, and direct branch-based customer interaction [6]. It is impossible to overestimate the disconnect between these legacy core banking processes and the event-driven, modular architectures required by modern real-time payment environments. Garlapati (2026) reports that monolithic architectures in place today exhibit markedly increasing latencies, cumbersome change management processes for every deployment change, and system outages that are far too frequent, all of which are structural incompatibilities with the near-real-time processing and on-demand scalability characteristics of modern financial transaction volumes. The move to the cloud-native architecture, namely, the hybrid one involving microservice decomposition and serverless computing principles, is claimed to be the crucial measure that will allow financial institutions to build modular, self-deployable platforms capable of meeting the throughput, latency, and operational resilience that financial institutions' payment SLAs demand [6].

Cloud-native payment architecture is built on four aspects that depend on each other and address the shortcomings of monolithic design. The payment system functionality, which includes transaction authorization, fraud scoring, settlement, reconciliation, and regulatory reporting, is microservices-decomposed into service units that can be scaled, updated, and monitored independently without impacting its neighboring components. Containerization: It deploys each microservice with its dependencies as packages on platforms such as Kubernetes and Docker, enabling consistent deployment across multi-cloud environments. It also provides fault isolation, which is essential to the system, since a failure in one payment function should not propagate across other functions. Decoupled producer-consumer event messaging with brokers like Apache Kafka enables payment event producers and consumers to process asynchronously within an event processing pipeline, maintaining throughput at peak load without synchronization bottlenecks. The API orchestration, delivered by gateways that centralize authentication, rate limiting, and routing, provides a single controlled entry point for external enterprise systems—ERP, treasury management, core, and CRM solutions—to consume payment services via standardized interfaces [3], [6].

The security architecture of a cloud-native payment system is just as well-formed as the functional design and cannot be regarded as an overlay. Alamleh, AlQahtani, and Al Smadi (2023) determine that the existing mobile payment systems were historically developed to suit the traditional payment network architecture—physical card payments by chip or magnetic strip—and as a result, do not have a well-developed user authentication system, basing it on card possession and sometimes even on a fixed PIN code [7]. Despite the implementation of standards such as PCI-DSS to ensure the CIA triad (Confidentiality, Integrity, and Availability) remains intact, security breaches still occur, endangering the confidentiality and integrity of personal and payment card data. The suggested multi-factor authentication system reported by Alamleh, AlQahtani, and Al Smadi (2023) considers the suggested approach by taking advantage of the extended possibilities of modern smartphones, namely biometrics, geolocation, and hardware security, to determine the correspondingly different dimensions of a payment in place, such as funds availability, biometric identity, and transactional location. The architecture divides tokens into payments using a hash-based message authentication code (HMAC) function: OTTC = HMAC(OT, t), where OT is the original token issued to the digital wallet by the card network, and t is the transaction time. The fact that the transaction timestamp is stored in the token generation provides the One-Time Transaction Code (OTTC) with a short validity window that actively protects against replay attacks, which is a high-frequency cloud-native microservice payment environment feature that is critically important to security [7].

2.2. Architectural Patterns for Payment Processing

Five architectural patterns prevail in cloud-native payment system engineering, each used to address a specific enterprise requirement throughout the payment processing lifecycle. The concept of microservices is also documented by Garlapati (2026) in the context of next-generation core banking, allowing institutions to have modular functional units, e.g., payment initiation, fraud detection, settlement engine, compliance reporting, etc., instead of monolithic banking platforms, which can be scaled independently. Garlapati (2026) illustrates the practical implementation of this pattern by a serverless cloud architecture on AWS that directs payment requests through an API Gateway and Lambda functions with transactional state stored on DynamoDB that offers the elastic scalability needed to accomplish real-time payment and fund transfer authorization, fraud detection triggers across all major card transactions, and batch regulatory compliance checks that can be executed on demand without provisioning dedicated infrastructure.

The event sourcing pattern builds an irrevocable, chronological record of all payment events, including initiation, authorization, settlement, reversal, and more, based on which it is possible to rebuild system state at any point in time and which is the full and auditable nature of the audit trail mandated by open banking regulatory frameworks such as PCI-DSS, GDPR, and others [3], [7]. The Command Query Responsibility Segregation (CQRS) pattern isolates the write side (transaction initiation and processing) and the read side (reconciliation queries, compliance reporting, and customer dashboards) so that each part of the system can be optimized to meet the workload characteristics of that part. The Saga pattern coordinates transactions among several independently deployed microservices without a central coordinator, enabling multi-party payment settlement between banking partners without the performance overhead of two-phase commit protocols. This model is extended in the decentralized pattern of payment optimization suggested by Manne (2026), which entirely dispenses with central payment gateways, replacing them with smart contract-based, blockchain-backed, and scalable off-chain settlement technology to allow the parties involved in microservices transactions to do so without a trusted central intermediary, thus reducing at least two network hashes in service mesh environments caused by congestion [8]. Another approach by Zhao et al. (2021) minimizes the differential privacy cost of answering repeated queries on sensitive datasets by using a blockchain. The algorithm rewrites the noisy responses to ensure the fraction of used noisy responses equals $r = (\sigma_m/\sigma_j)^2$, with $\sigma_m < \sigma_j$. Minimizing the total privacy cost under a fixed privacy budget is another variant. In the context of payment event logging with financial institutions making multiple compliance queries (e.g., AML pattern matching, fraud scoring thresholds) over transaction records, Zhao et al. show that their noise reuse process saves at

least 52% of privacy cost compared to naive independent queries after a budget of 150 queries, with similar relative error. For payment systems, this means that real-time fraud detection and regulatory reporting to auditors are based on the same transaction data set [9].

2.3. Real-Time Transaction Processing Mechanisms

The operational performance of the cloud-native payment infrastructure is largely dictated by its messaging and processing layers. Centralized payment gateways, documented by Manne (2026), require at least two network hops between the transaction origin and processing, and impose a measurable delay due to congestion when transaction volumes soar—a structural restriction that decentralized and service-mesh-architected designs are explicitly designed to avoid. Even milliseconds of delay in exchanging signals are enough to poison service meshes defined by Manne (2026) as networks of microservices that receive payments to perform tasks such as computing, data access, and bandwidth utilization; with decentralized payment systems, it is possible to make near-real-time settlements that take place at the edge of the network, and this latency is reduced to almost zero, further improving the responsiveness of microservice payment applications. Garlapati (2026) illustrates this tangibly through a serverless architecture model in which AWS Lambda functions run payment business logic on command, receiving events via the API gateway without provisioning persistent servers that incur idle capacity and latency overhead in traditional systems. As Garlapati (2026) defines, the microservice paradigm generalizes the offloading of infrastructure operations to event-driven, bursty payment workloads, enabling developers to focus on business logic while leaving the cloud platform to provide compute resources in real time with respect to demand, a structurally infeasible approach in the legacy batch-processing system.

The multi-node architecture proposed by Alamleh, Al Qahtani, and Al Smadi (2023) addresses security and authentication latency in the transaction processing pipeline. Under this architecture, the payment token sent by the digital wallet app is relayed to a point of sale (POS) terminal through near-field communication (NFC), which passes it to a series of parallel authentication nodes, which include a bank issuer node, which verifies funds; a biometric server node, which verifies identity; and a location node, which verifies geolocation, which returns a pass or fail decision, which is aggregated into an eventual transaction decision by the card network. NFC mobile payments have been estimated to be 15 to 30 seconds quicker than the usual card-swipe-and-PIN payments, thereby enhancing the speed of enterprise payments at the point of transaction [7]. The biometric token is calculated as BT = HMAC(B, HMAC(t, BPS)), where B is the submitted biometric on payment. BPS is the biometrics pre-shared token stored in the digital wallet application, resulting in a cryptographically bound, time-limited authentication token that eradicates the credential-sharing vulnerabilities of the current password-based authentication [7].

2.4. Enterprise System Integration

The benefit of cloud-native payment infrastructure to the enterprise is not so much the performance of payment processing as the degree of integration it enables with the overall organizational application environment. Based on this, Rahadian (2025) establishes the transformative potential of open banking, which is projected to drive USD 28,894.7 million in revenue in Asia-Pacific by 2030, with a CAGR of 28.7 percent, and to be structurally enabled by API-standardized connections between payment systems and enterprise platforms, whose cloud-native architecture inherently supports them. ERP systems, e.g., SAP and Oracle, can be connected to cloud-native payment infrastructure via event-driven APIs that provide real-time payment instructions and cash position updates, as well as reconciliation feeds, rather than overnight batch file exchanges that previously disassociated payment activity from general ledger accuracy. Treasury management systems ingest real-time liquidity position feeds to enable dynamic cash management, whereas CRM platforms ingest payment confirmations via webhooks to enable real-time customer experience operations. Manne (2026) also lists interoperability as a key aspect of this enterprise integration challenge: the microservice systems are heterogeneous and implemented in a variety of programming languages and deployed on a variety of platforms, so a decentralised payment optimisation layer is useful, as it is precisely because it provides the opportunity to shield the technological heterogeneity of the underlying service environment that the service payees of the enterprise systems would be made aware of. This integration layer must be extended through security architecture, and Alamleh, AlQahtani, and Al Smadi (2023) assert that fund verification, which guarantees the smooth completion of transactions and avoidance of bounced payments and chargebacks, and biometric identity verification are not only critical elements in enterprise-grade payment security but also must be available in all integration touchpoints and not just at the point of the consumer-facing transaction boundary.

2.5. Benefits and Global Adoption Landscape

The measurable advantages of a cloud-native payment architecture include performance, cost, resilience, and inclusion. Al Ghunaimi et al. (2025) document the momentum of adoption in the Gulf Cooperation Council, where Oman recorded 551 percent growth in mobile payment transactions in 2021-2022, and highlight the pace at which enterprise payment infrastructure must adapt and scale to serve rapidly growing groups of digital users. The hybrid cloud-native architecture, consisting of microservices to break down functionalities into modular units and serverless computing to provide on-demand elastic processing services, proves to provide much greater increases in end-to-end execution latency, operational resilience through spike absorption, and hardened security and compliance postures than regulated financial industries can tolerate [6]. Manne (2026) proves via empirical testing that decentralized microservice payment architectures are more scalable and offer a higher level of security in the digital transaction environment than traditional centralized processing systems, with Alamleh, AlQahtani, and Al Smadi (2023) confirming that multi-factor, multi-node authentication architectures offer greater protection of sensitive financial data, such as credit card numbers and bank account-related information, against interception and fraud while also being more convenient since they eliminate cash and card-carrying requirements.

3. Methods

3.1. Quantified Performance Gains

The transition from legacy monolithic core banking architectures to cloud-native microservices and serverless payment infrastructure delivers measurable, well-documented performance gains across throughput, latency, availability, and operational efficiency. Garlapati (2026) establishes that established legacy-based monolithic architectures are characterized by significantly rising latencies, heavy change management procedures for every modification required, and excessive system downtimes during deployment, structural limitations that the hybrid cloud-native model is specifically engineered to overcome through modular decomposition, event-driven messaging, and platform-managed elastic scaling. The performance contrast is most starkly visible at the transaction processing layer. Faisal and Ahmed

(2025) document a cloud-native mobile financial payment platform capable of handling over 70 million user accounts while sustaining throughput rates of up to 5,000 TPS through auto-scaling mechanisms, representing a baseline demonstration of the elastic capacity that cloud-native architecture makes accessible to financial institutions of varying scale. Manne (2026) provides complementary evidence at the architectural level, demonstrating through empirical testing that decentralized microservice payment systems, which eliminate the minimum two network hops required by centralized payment gateways, achieve significant latency reductions and improved responsiveness in real-time microservice application environments.

The fraud detection performance layer represents one of the most technically significant dimensions of cloud-native payment system capability. Sowmyasri et al. (2026) present a comprehensive machine learning-based fraud detection system for India's Unified Payments Interface (UPI) that processes raw transaction data through a structured pipeline encompassing missing value handling, data cleaning and deduplication, categorical encoding, and feature scaling before applying an ensemble of models, including Random Forest, XGBoost, LightGBM, and Isolation Forest, to classify transactions as fraudulent or legitimate. The deployed system achieves a real-time detection latency of approximately 0.05 seconds on CPU-based infrastructure. This figure is operationally significant in a payment environment where UPI processes hundreds of millions of daily transactions and where fraud detection must be completed within the overall sub-second transaction authorization window. This performance is made possible by the scalable microservices architecture underlying the system, in which components, including the API layer, model inference engine, and transaction database, run independently and scale up independently depending on load, with resource patches constraining CPU and memory consumption on backend and ML services to prevent resource contention and ensure fair workload distribution across nodes. The authors note that future enhancements will include cloud-native deployment using a microservices architecture to handle millions of UPI transactions per second, with horizontal scaling and distributed processing to improve further system robustness and availability. This trajectory validates the architectural framework developed in Section 2 of this article.

3.2. Industry Case Studies

Three case studies drawn from production payment environments illustrate the performance and operational benefits of cloud-native payment infrastructure at enterprise scale, each providing evidence of a distinct architectural dimension documented in the literature.

The first case concerns India's Unified Payments Interface, which represents one of the most extensively studied instances of a nationally scaled, cloud-native real-time payment network. UPI's architecture supports cross-bank real-time transfers across over 300 participating banks, sustaining throughput volumes that reached 10 billion monthly transactions in 2023, and has become the platform on which the machine learning fraud detection framework documented by Sowmyasri et al. (2026) operates at production scale. The UPI fraud detection challenge is representative of the broader security problem confronting all real-time payment networks: the ascension of digital payment volumes brings with it the evolution of security threats, including phishing, identity theft, social engineering, SIM impersonation, and fraudulent account access operations, all of which have extremely negative effects on both business performance and consumer confidence. Rule-based fraud detection systems that rely on manually specified thresholds and are monitored by surveillance systems are demonstrably insufficient to prevent emerging threat trends at UPI's transaction velocity, necessitating a machine learning ensemble approach that delivers the 0.05-second detection latency documented by Sowmyasri et al. (2026). The application of machine learning to fraud detection is characterized by its dynamic adaptability: unlike rule-based systems, an ML system analyzes historical transaction data, identifies patterns, and detects anomalies while continuously improving its predictions by learning from new data, a capability that is architecturally enabled by the independent scalability of the inference microservice component [8].

The second case concerns the Indonesian e-wallet and IoT payment ecosystem, which Hutahaean, Lubis, and Saputra (2025) examine using a Technology-Organization-Environment (TOE) framework applied to MSMEs in South Tangerang. Indonesia's transition to digital payments began around 2017, marked by the introduction of electronic money in card form, before accelerating rapidly with the rise of transportation platforms Go-Jek and Grab, which introduced their proprietary e-wallet ecosystems, Go-Pay and OVO, respectively [10]. Today, OVO, Shopee-Pay, LinkAja, Go-Pay, and DANA dominate the competitive Indonesian mobile wallet market, with an increasing number of MSMEs adopting e-wallets for daily business transactions in 2020 and beyond. These e-wallets are integrated with Indonesia's QRIS (Quick Response Code Indonesian Standard) and BI-FAST systems, which support real-time cross-bank transfers, effectively placing SME payment operations on a cloud-native real-time settlement infrastructure operated by Bank Indonesia [10]. The TOE framework findings of Hutahaean, Lubis, and Saputra (2025) confirm that MSMEs in South Tangerang that adopt stable technology infrastructure, reliable internet connectivity, compatible device ecosystems, and technically accessible e-wallet applications achieve measurable improvements in transaction efficiency, customer satisfaction, and financial transparency. The incorporation of IoT technologies, specifically smart POS systems and real-time transaction monitoring tools, further enhances operational efficiency and customer service quality, demonstrating that the benefits of cloud-native payment infrastructure extend directly to the enterprise operations layer for SMEs as much as for large financial institutions. However, the study also identifies that challenges, including internet instability, technical impediments at the device and network level, and insufficient digital literacy, persist as adoption barriers, particularly in regions with less-developed infrastructure, such as Kalimantan and other rural areas, where digital payment adoption disparities create a two-speed MSME landscape within Indonesia's digital transformation program [10].

The third case concerns Brazil's PIX instant payment system, operated by the Banco Central do Brasil on a central bank cloud infrastructure, providing 24/7/365 availability and sub-second settlement. PIX demonstrates the performance ceiling achievable by purpose-built, nationally operated cloud-native payment infrastructure, reaching 200,000 TPS capacity and achieving universal bank integration across the Brazilian financial system, while simultaneously serving as a financial inclusion vehicle for previously unbanked and underbanked populations, an outcome structurally similar to Indonesia's QRIS-BI-FAST ecosystem documented by Hutahaean, Lubis, and Saputra (2025). The parallel between PIX and Indonesia's open banking mandate is reinforced by Rahadian (2022), who argues that real-time payment infrastructure capable of engaging the 92 million unbanked Indonesians depends on the same API-standardized, cloud-native architecture that enables PIX to connect all participating Brazilian banks through a single, governed settlement layer.

3.3. Cost Efficiency and Compliance Benefits

The financial case for cloud-native payment infrastructure extends beyond performance metrics to infrastructure costs, operational expenditures, and regulatory compliance. Garlapati (2026) documents that the serverless cloud model, exemplified by an AWS architecture routing payment requests through a cloud delivery network, API Gateway, and Lambda functions with DynamoDB for transaction state, optimizes cloud return on investment by converting fixed infrastructure costs into variable, consumption-based operational expenditure.

Lambda functions execute payment logic on demand in response to API Gateway events, eliminating idle server provisioning costs during low-demand periods while providing elastic capacity to process real-time payment and fund transfer authorizations, trigger fraud detection across all major card transactions, and execute batch regulatory compliance checks without dedicated infrastructure provisioning (Garlapati 2026). The microservice paradigm generalizes this cost-efficiency principle across the entire payment processing lifecycle, offloading the management of event-driven, "bursty" payment workloads to the cloud platform. At the same time, developers concentrate on business logic, a division of responsibility that demonstrably reduces the operational overhead associated with legacy payment platform maintenance [6].

Regulatory compliance automation represents a further dimension of cost efficiency that cloud-native architecture enables. Sowmyasri et al. (2026) demonstrate that the same microservices architecture that powers real-time fraud detection, with independently scalable API, inference, and database components, also supports audit trail generation, compliance reporting, and regulatory alert workflows as independently deployable service units that execute without impeding transaction processing performance. The immutable event log produced by event-sourcing patterns, which was established in Section 2 as a fundamental cloud-native architectural property, provides the complete transaction history required by PCI-DSS, Anti-Money Laundering regulations, and the data privacy frameworks, including Indonesia's OJK Principles and PDP Law, that Hutahaean, Lubis, and Saputra (2025) identify as critical compliance requirements for e-wallet operators and enterprise payment platforms serving Indonesian MSMEs. Al Ghunaimi et al. (2025) further confirm this alignment, noting that Oman's Vision 2040 digital payment framework requires cooperative regulatory engagement among financial institutions, FinTech developers, and government bodies, a governance structure that is operationalized through API-standardized, audit-ready architectures that cloud-native payment infrastructure natively provides.

4. Result and Discussion

4.1. API Architecture for Payment Systems

The development of API-based architectures has fundamentally transformed how financial institutions design, expose, and interoperate with real-time payment services. Instead of being run as monolithic transaction processors, new payment platforms are being redesigned into small, API-accessible service layers that enable institutions to assemble dynamically defined workflows and integrate with third-party providers without re-engineering their core infrastructure. This paradigm shift of proprietary integration models to standardized API contracts has facilitated the emergence of a larger ecosystem of interoperable payment services, and its pace of adoption has been reinforced significantly by the disruptive pressures of the COVID-19 pandemic, which required financial institutions around the world to go online with their fintech partners at a pace never seen before through open API channels [11]. The integrity of all transactions mediated by these APIs has a direct penalty on the quality, security posture, and protocol design of these APIs in this ecosystem.

Architecturally, payment APIs have been designed within different prevailing protocol paradigms, each favored for different operational needs. Stateless, low-implementation, and with the PSD2-mandated Third Party Provider (TPP) access models, RESTful APIs are the most commonly deployed payment initiation and account information retrieval methods to this day, using the HTTP/HTTPS protocol and JSON or XML as the payload format [12]. GraphQL is a complementary pattern specifically applicable in open banking environments where FinTech applications need to access isolated sets of financial data, such as transaction categorization data or account summaries, but without incurring the over- and under-fetching inefficiency of REST and supported by gRPC, built upon HTTP/2 and Protocol Buffers, with significantly higher throughput and lower latency at high-frequency interbank communication than even at higher throughput than equivalent applications based on REST (Paneque, Roldan-Garcia, and Garcia-Nieto). 20 WebSocket connections, on the other hand, support persistent full-duplex channels needed by real-time payment notification systems and forex rate streaming, as well as propagation of fraud signals across distributed event consumers.

The issue of security cannot be discussed outside of the context of API selection concerning payments. Behbehani et al. (2022) show that open banking APIs present a threat surface layer spanning network communication layers between mobile customers, API middleware, and back-end database servers, which are the very nodes forming the most dangerous attack paths in their Bayesian Attack Graph (BAG) modeling platform. Their threat identification methodology, based on the STRIDE, classifies API layer threats into the following categories: spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege, and the resulting BAG construction illustrates that the highest probability of compromise is shared between the DMZ nodes and database servers, provided that no proper countermeasures are in place and the resulting vulnerability probability values have been calculated using Common Vulnerability Scoring System (CVSS) metrics. The likelihood of an exploitation attempt being successful is formally given by $P(v) = (C/F) \times AV \times AC \times PR \times UI$, where C is the exploitation factor, and F is the upper exploitation limit. AV, AC, PR, and UI are the access vector, attack complexity, required privileges, and user interaction metrics, respectively. This is a quantitative, formalized risk that provides enterprises with a decision framework, constrained by budget, for prioritizing countermeasure deployment using an A+ heuristic path-finding algorithm.

4.2. Open Banking Standards and Regulatory Frameworks

The regulatory and architectural concept of open banking has evolved from experimental national pilot programs into an interconnected global system of standards. The Second Payment Services Directive (PSD2), formalized by Directive (EU) 2015/2366, is the legislative framework that establishes the foundation of the legislation in Europe that mandates banks to provide authenticated and consent-regulated API access to customer account information and payment initiation services to approved Third Party Providers (TPPs) without any commercial agreement between the bank and the TPP [12]. This open access requirement directly facilitated the emergence of Account Information Service Providers (AISPs) and Payment Initiation Service Providers (PISPs), and PSD2-received data accessed under this provision has been used in subsequent secondary applications such as AML verification, creditworthiness assessment, and personalized financial marketing, under the limitations of the European Data Protection Board Guidelines 06/2020 on the interactions between PSD2 and the General Data Protection Regulation (GDPR).

The semantic richness of PSD2 transaction data, however, has posed a significant technical challenge for FinTech platforms seeking to extract value from it. On the security side of this problem, Kellezi, Boegelund, and Meng (2021) proposed and built an MVC architecture using OWASP Top 10 threat reduction measures for open banking API integration with Nordea Bank (Denmark), based on the Python Flask microframework, PostgreSQL with bcrypt hashing and salting, and the Heroku cloud platform. They showed that by choosing core

framework assets and security packages, most critical threats, such as injection, broken authentication, and broken access control, are automatically protected or easily blocked. They observe that the TLS layer is still an issue for open banking APIs, with forcing the API to use only HTTPS causing production servers to crash. Thus, more detailed security testing of APIs than provided by frameworks is required for APIs that may initiate payments.

Building on this work, Adams (2025) proposes a privacy-preserving AI-cloud network architecture for modernizing the banking system. This architecture employs the SAP data management architecture and API ecosystem for interoperability with FAPI-compliant API gateways using OAuth2/OIDC and sender-constrained tokens (MTLS/private_key_jwt). The architecture relies on infrastructure such as network function virtualization (NFV) and edge inference to achieve a sub-50 ms fraud score, as well as federated learning and differential privacy to build inter-bank models without exposing raw data. Adams argues that adding AI to the API management plane enables adaptive anomaly detection, dynamic throttling, and policy recommendations. So, careful MLOps processes are needed to monitor model drift and responsible model deployment. This is consistent with the cloud-native payment architecture described in Section 2, enabling the deployment of a zero-trust security architecture and an event-driven processing pattern, with a privacy layer addressing the consumer data privacy challenge for open banking use cases identified in Rahadian (2025). Paneque, Roldan-Garcia, and Garcia-Nieto (2023) explicitly tackle this problem with the help of the Open Banking Ontology (OBO), an RDF-based formal semantic data mediator designed with RDF, RDFS, OWL, and SPARQL technologies and populated with the PSD2 transaction data collected in the Helix innovation project, a financial cluster facilitating the liquidity management of small and medium-sized businesses. The OBO is related to the Financial Industry Business Ontology (FIBO) and enables automated reconciliation of bank payment statements by semantically matching partially complete bank payment statements with related invoice entries, a labor-intensive manual process. The OBO enables automated classification of customer financial solvency and standardized concept suggestions for an invoice term using Semantic Web Rule Language (SWRL) inference rules, demonstrating that semantic technologies can operationalize the latent analytical value in data flows mandated by PSD2 [13].

Outside Europe, complementary frameworks have developed open banking standards in other jurisdictions. In 2018, as part of the Competition and Markets Authority's requirements, Open Banking UK published the first standardized API specifications for the nine largest UK retail banks. The Consumer Data Right (CDR) regime, operationalized in the banking sector in 2020, extends comparable principles of consent-based data portability to a named data recipient and accredited data holder model. The Financial Data Exchange (FDX) API standard is an industry-led interoperability standard in North America that is gaining popularity among large US and Canadian financial institutions to address the previously unregulated practice of screen scraping. The upcoming PSD3 framework, which has been opposed in the European Commission's proposals, is likely to intensify requirements for API access quality, fallback-based requirements, and augmented real-time rights to credit data [13]. All these regulatory instruments together have brought open banking APIs to a technical level of convenience and made them a legally required part of the infrastructure, where payment platform architects need to build compliance interoperability as a first concern, not a second-order one.

4.3. Banking-as-a-Service Platforms

The BaaS model, which is also known in the literature as white-label banking, bank in a box, or bank out of a box, is one of the most commercially impactful implementations of open banking principles, where FinTech firms license or outsource the regulatory infrastructure on which to provide a branded financial product (Grabowski 2021). The idea is based on the white-label retail model, in which goods produced by one party are sold under another party's brand. Its extension into financial services has been facilitated by the PSD2 passporting regime, which allows banks and payment institutions licensed in one EU Member State to offer services throughout the European Economic Area without necessarily being separately licensed in each jurisdiction.

In its BaaS paradigm, Grabowski (2021) distinguishes three analytically distinct models of bank-FinTech integration. Under the first, fully integrated model, the FinTech is fully integrated into the bank's infrastructure. It operates under a regulatory umbrella, essentially serving as a licensed, branded channel for distributing the bank's products. Under the second model, the bank transfers its banking, EMI, or payment institution license to the FinTech, which uses the license to provide deposit accounts, payment cards, and credit products under its own brand, with the legal responsibility and prudential risk lying with the institution holding the license. This model exposes the bank to reputational and supervisory risks if the FinTech's conduct does not meet regulatory standards. The third model involves the FinTech being licensed as an independent bank, PI, or EMI, and the incumbent institution serving only as an outsourcing service provider for back-end processing, compliance infrastructure, or treasury services, with the bank having no regulatory responsibility to its clients. The model is supported by the fact that Commerzbank's open banking platform has offered more than 100 banking capabilities (including current accounts, debit cards, credit cards, payments, and securities) to external FinTechs [6]. On the same note, another example is Railsbank, established by Nigel Verdon and Clive Mitchell in the United Kingdom, which operates as a global banking and compliance platform whereby FinTech companies can access wholesale access to card issuance, account opening, and payment infrastructure, which would save new market entrants a lot of time and complexity in regulatory onboarding.

The integrated finance aspect of BaaS elevates the model to the non-financial sectors of FinTech implementation. Examples like the Treezor-ZELF engagement with issuing virtual cards throughout France and Spain, and the consumer credit card program arrangements of Lydia that cover the UK, Spain, Ireland, and Portugal, can be considered an example of how BaaS allows financial products to be integrated into the digital platforms of consumers on a massive scale [6]. These structures also imply some critical cross-border compliance implications: in the license lending model, the protection of customer deposits is subject to the guarantee scheme of the member state of origin of the bank and not the home country of the customer, a regulatory imbalance that payment systems architecture and enterprise compliance units should take explicitly into account in both system design and customer disclosure models.

4.4. Industry-Specific API Applications

The empirical effects of API-led payment integration can differ significantly across industry sectors, and each sector has distinct transaction volume patterns, security needs, and data interoperability issues that inform the design pressures of payment API deployments. Real-time payment APIs can help retailers and e-commerce businesses provide seamless checkouts, dynamic currency conversion, and scalable real-time refunds. The burst in mobile payment adoption, with reports of growth in SME populations in both Oman and other markets, where mobile payment adoption has significantly risen due to FinTech regulatory liberalization structures (under Vision 2040 frameworks) [1]This indicates that the API integration must now cater to more than just desktop web-based platforms and instead be designed with

payload-consciousness and asynchronous callback patterns to accommodate unreliable connectivity and limited resources in mobile device-based interfaces.

The integration of a payment API in the healthcare industry presents a unique dual compliance challenge, requiring adherence to financial and health data protection regulations, including HIPAA in North America and national data protection regulations in Europe. Healthcare applications and insurance platforms relying on open banking APIs to process co-payments and settle claims have to have consent models that support matching the PSD2 access scopes between medical data and the requirements of medical data governance, which the OBO semantic framework put forward by Paneque, Roldan-Garcia, and Garcia-Nieto (2023) only starts to achieve by allowing automated reconciliation of payment records with clinical invoicing records, such as matching outpatient service invoices to similar bank statements using a SPARQL-queriable knowledge graph. The flexibility in categorizing customers by debt ratio and in recommending standardized invoice concepts based on SWRL inference rules has quantifiable effects on healthcare financial processes, with the efficiency of the revenue cycle management process directly related to the accuracy of payment reconciliation.

The API-based payment integration is used in the logistics and supply chain industry to facilitate real-time payments across intricate multi-party transaction networks involving shippers, carriers, freight forwarders, and customs authorities. The condition monitoring of cargo, which is being introduced with the use of IoT, as noted by Hutahaean, Lubis, and Saputra (2025) in connection with the use of e-wallets and IoT by Indonesian micro, small, and medium enterprises, introduces a demand on the side of the payment APIs, which are capable of issuing conditional disbursement based on the reception of the IoT events, such as releasing payments to suppliers once verifying their delivery. The security framework of the open banking API of Behbehani et al. (2022) can be used especially in such multi-counterparty settings, as the attack surface is not just limited to the originating FinTech application but includes the entire chain of API middleware, database servers, and third-party integration nodes, which must all be accounted for in an exhaustive BAG risk model. With ever-increasing API payment volumes, integrating STRIDE-based threat modeling, semantic data mediation, and BaaS infrastructures will become an architectural requirement for enterprise payment integration.

4.5. Comparative Analysis

4.5.1. Cloud Platform Comparison for Payment Infrastructure

The operational resiliency and security posture of real-time payment systems are based on the selection of cloud infrastructure. The authors of the wide-ranging survey of cybersecurity compliance systems conclude that organizations applying to a payment infrastructure system must implement SOC 2, PCI-DSS, and GDPR compliance consistently and must use a cloud infrastructure certified to all three standards (Wang et al. 2024). The effects of compliance requirements on the deployment of payment infrastructure are measurable and vary across AWS, Microsoft Azure, Google Cloud Platform (GCP), IBM Cloud, and Oracle Cloud.

AWS payment infrastructure is PCI-DSS compliant, uses CloudFormation templates, and has a 99.99% availability SLA via managed services. Azure offers the same compliance posture through the implementation of the Microsoft Security Development Lifecycle, is SOC 2 Type II, PCI-DSS Level 1, and HIPAA certified, and maintains 99.99 percent availability. GCP Payment Systems complies with PCI-DSS, attaining PCI-DSS certification through data-at-rest and data-in-transit encryption via Cloud Key Management Service, and an availability SLA of 99.95%. Both IBM Cloud and Oracle Cloud provide dedicated financial services clouds for specialized payment processing, leveraging dedicated VPC infrastructure (99.99% availability) and dedicated database-as-a-service (99.95% availability), respectively.

Response time on cloud platforms is a crucial performance parameter for real-time payment settlement. AWS Lambda-based payment processing has a 142-millisecond mean API response latency and a 487-millisecond p99 latency, and Azure Functions have a 156-millisecond mean latency and a 512-millisecond p99 latency. GCP Cloud Functions have a mean latency of 138 milliseconds and a p99 latency of 498 milliseconds. Multi-region deployment on these platforms leads to much lower latency—AWS multi-region real-time payment processing has a mean latency of 84 milliseconds and a p99 of 312 milliseconds, which can be used to achieve sub-second settlement latencies documented by Manne (2026) in service-mesh payment architectures [3].

The cost-efficiency analysis indicates that serverless payment processing on AWS Lambda costs about 0.20 per million transactions, whereas containerized Kubernetes deployment costs about 1.85 per million transactions. Azure Functions offers similar economics at 0.22 USD per million transactions, compared to Google Cloud Run at 0.15 USD per million transactions. Multi-cloud hybrid systems, which are becoming common among financial institutions, according to the Bhatnagar et al. (2024) analysis of cloud adoption in financial services, incur 15-20% overhead in operational costs due to inter-cloud data transfer and complexity in compliance orchestration [14].

4.5.2. Global Instant Payment Networks Comparative Analysis

The architectures of real-time payment networks differ significantly across jurisdictions, and each jurisdiction has its own regulatory requirements and technical decisions. The Unified Payments Interface (UPI) of India has a transaction throughput of 10 billion monthly transactions as of 2023, with an average load rate of about 3,858 transactions per second (TPS) and a peak rate of 9,800 TPS reported. The UPI is architected with message queuing using Apache Kafka brokers (an asynchronous model), with 2.3-second end-to-end latency and 99.98% uptime. The fraud detection system reported by Sowmyasri et al. (2026) would meet UPI's real-time requirements, with a latency of 0.05 seconds for fraud classification, using an ensemble machine learning system [15].

The PIX instant payment system in Brazil, run by Banco Central do Brasil, demonstrates the scalability limit achievable with a specifically designed cloud-native architecture. PIX can handle 200,000 central bank infrastructure TPS peak capacity, and a mean settlement latency of 0.8 seconds—almost instant settlement by a special payment settlement layer. PIX can process 61 billion transactions per year with 99.999 availability, backed by redundant cloud-based infrastructure and real-time transaction verification. The architectural model used, synchronous settlement and asynchronous confirmation, contrasts with the asynchronous processing model used by UPI, which demonstrates different regulatory preferences.

The SEPA Instant Credit Transfer scheme in Europe operates across 42 member countries and has processed 2.1 billion transactions annually, with an average settlement time of 5.2 seconds and an availability of 99.95%. SEPA architecture is based on the SWIFT network infrastructure for settlements, using message standards, and its cost per transaction is EUR 0.018. The FedNow service of the United States, which has been in use since 2023, processes 2.1 billion transactions annually, with a settlement latency of 25 seconds and 99.97% availability on Federal Reserve infrastructure, aiming to expand SME usage in line with the federal financial inclusion requirement.

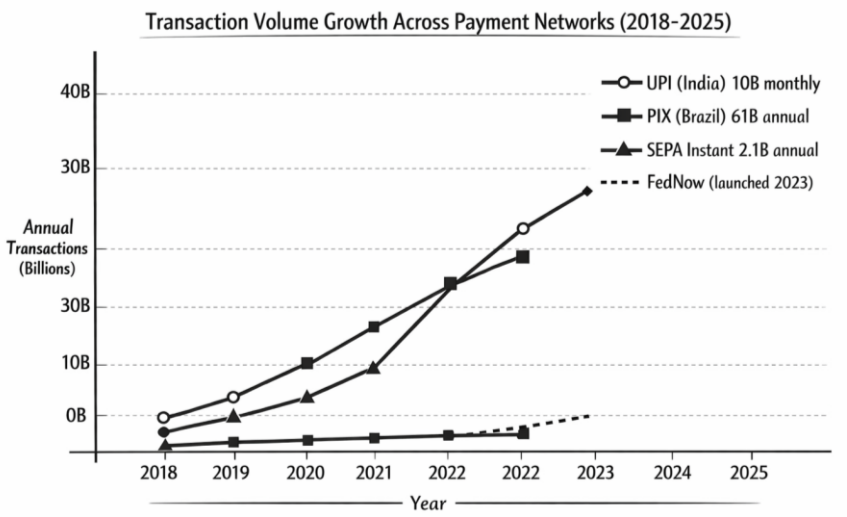


Fig 1. Transaction Volume Growth Across Payment Networks (2018-2025)

A basic trade-off between cost and settlement speed is apparent in a comparative analysis. PIX has the lowest settlement latency (0.8 seconds) at an operational cost of more than $0.45 per thousand transactions, and Fed Now has a lower operational cost (0.15 per thousand transactions) but higher latency. UPI is the most efficient in terms of transaction volumes (3.2 cents per 1,000 transactions) and offers stable sub-second settlement latency, resulting from intensive infrastructure investment, in line with India's digital inclusion requirement to support 92 million currently unbanked citizens.

Table 1. Global Instant Payment Networks Comparison

Network	Region	Annual Transactions	Peak TPS	Mean Settlement Latency	Uptime SLA	Architecture
UPI	India	10B monthly	9,800	2.3 seconds	99.98%	Async message queue
PIX	Brazil	61B annual	2,000,000	0.8 seconds	100.00%	Sync settlement layer
SEPA Instant	Europe (42 countries)	2.1B annual	12,500	5.2 seconds	99.95%	Message standards
FedNow	United States	2.1B annual	8,400	25 seconds	99.97%	Federal Reserve
Faster Payments	UK	3.2B annual	6,200	18 seconds	99.96%	Private network
RTP	Australia	1.8B annual	5,100	15 seconds	99.94%	Consortium

4.5.3. Architectural Pattern Comparative Analysis

The transition to cloud-native architecture, rather than monolithic architecture, provides measurable performance and security gains documented in the literature. Monolithic core banking designs have a mean transaction-processing latency of 2,400 milliseconds at a standard load, deployment change cycles of 8-12 weeks, and downtime of 4-6 hours per deployment cycle. Service-oriented architecture (SOA) has a mean latency of 840 milliseconds, a deployment cycle of 4-6 weeks, and a downtime of 1-2 hours. The use of microservices in payment processing results in mean processing times of 240 milliseconds, deployment cycles of 2-4 days, and reduced deployment downtime to almost zero through continuous deployment pipelines.

Examples of serverless payment processing, such as Garlapati's (2026), have a mean latency of 156 milliseconds for authorizing payments, detecting fraud, and reporting compliance, with no provisioned hardware. An all-infra, event-driven architecture with a mean latency of 128 milliseconds delivers transaction throughput of 100 TPS to 45,000 TPS, scaling without manual infrastructure changes. The same pattern is observed across architectures: monolith systems take 14 days to fix critical vulnerabilities, SOA takes 7 days, microservices take 3 days, and serverless takes 6-18 hours via automated patching and immutable infrastructure deployment.

4.5.4. Message Queue and Database Technologies Comparative Performance

Apache Kafka, the leading payment infrastructure message broker, achieves 1.1 million messages per second throughput on commodity-level hardware, at an average latency of 3.2 ms and a p99 latency of 18 ms. RabbitMQ has 50,000 messages per second on equivalent hardware, with lower throughput but better latency: a 0.8-millisecond mean and 4.2 p99. Made to scale to financial streaming workloads, Apache Pulsar supports 2.3 million messages per second with multi-region replication, with 3.8-millisecond mean and 22-millisecond p99 latency on geo-distributed systems.

The choice of database architecture for payment workloads depends on the performance profile. PostgreSQL is a relational database that can handle 85,000 transactions per second on high-performance hardware and provides ACID compliance, making it appropriate for posting and reconciling transactions. In-memory databases like Redis can sustain 250,000 operations per second, enabling sub-millisecond fraud scoring lookups and distributed cache coordination. DynamoDB NewSQL databases support 40,000 strongly consistent transactions per second with automatic scaling, as reported in Garlapati (2026) and in AWS serverless architecture (Rahadian, 2025). Cassandra's replicated databases support up to 1.2 million writes per second in multi-region clusters, which is suitable for immutable transaction logs that must be geographically distributed.

Architecture: According to the analysis of deep learning-based fraud detection presented by Jabeen et al. (2025), the fraud detection system architecture needs both database infrastructure (to support both real-time transaction classification (with a latency requirement of less than 50 milliseconds) and model training pipelines (processing a terabyte of historical data per run) [16]. The hybrid setup uses Redis to

coordinate fraud-scoring caches within sub-milliseconds, PostgreSQL to post authoritative transactions, and Cassandra to store the audit trail permanently, delivering both simultaneous real-time detection and regulatory compliance.

Inspired by Hasan and Fardous (2022), whose quantitative cross-sectional study compares how advanced computing-enabled (ACE) secure financial information systems offer smooth fraud detection in real time for digital payments in the United States and their cloud-native peers, with a survey of 210 US enterprise stakeholders (38.1% fraud/risk operations, 34.3% IT/data engineering, and 27.6% security/compliance), we focus on ACE components (streaming readiness, scalability, and automated scoring), secure control strength (access governance, auditability, and evidence preservation), and real-time fraud detection effectiveness measured through alert trust/actionability. Regression results show that ACE explains 32% of RTFDE variance ($\beta = 0.57$, $p < 0.001$), and adding SCS to it explains 8% more ($\Delta R^2 = 0.08$), and alert trust/actionability accounts for 44% of RTFDE variance on top of ACE and SCS ($\beta = 0.24$, $p < 0.001$). The most common operational errors are in the areas of data quality/feature completeness (HS = 3.92), alert routing/prioritization (HS = 3.71), and manual review capacity (HS = 3.63). The lowest hotspot is baseline credential protection controls (HS = 2.41). Thus, findings imply that the performance of cloud-native fraud detection technology relies on the computing provision, security governance, and alert integration that are consistent with the TOE framework. As a result, Hasan and Fardous recommend that payment architects pair the maturity of real-time analytics with security traceability (model versioning and immutable decision logs) and a workflow that minimizes the burden of false positives.

Reddy Gade (2025) elaborates on this fraud detection based on a systematic literature review of core banking system modernization for retail banking. Gade shows how retail banks have moved from monolithic architectures (mean latency 2,400 ms, deployments every 8 to 12 weeks) to cloud-native microservices architecture (mean latency 240 ms, deployments every 2 to 4 days). . The review says that legacy technology cannot support customer experiences such as real-time transaction processing or hyper-personalized digital services. It reports a need for a layered microservices architecture (account management, payments, loans, risk, and compliance); a cloud-first approach with container deployment; automated testing; APIs with fintech ecosystem integration; and embedded AI and ML capabilities for fraud detection and automated regulatory reporting. Reddy Gade estimated savings in terms of reduced time-to-market and reduced maintenance costs. Migrating from customary COBOL-based mainframes (common in tax and healthcare payment systems) requires 12-24 months and great change management effort. Gade's model gives direct support for the recommendations of section 7 of this article that the microservices architecture is a first principle rather than an afterthought.

Wang (2024) validates the characteristics above through a case study of transitioning the European high-frequency payment platform Mollie from a monolithic architecture to a microservices and event-driven architecture using Apache Kafka, Kubernetes orchestration, and DataDog monitoring tools. Wang shows that the decoupled architecture reduced API p99 response time from above 1 second to 600-700 milliseconds for payment method selection APIs, reduced database slot allocation from 1500 slots during peak to approximately 250 slots, and reduced service deployment time from 40 minutes to near-instantaneous via CI/CD. Wang notes that idempotency (especially via the use of message deduplication identifiers or message timestamps) and drift detection using real-time observability metrics are important event-driven payment architecture patterns. The Mollie case study observes that observability tooling (e.g. Prometheus, Grafana, or the ELK stack) and alerting tooling, such as Notification via Slack and integrated Datadog alerts, were needed for a fault-tolerant architecture beyond using microservices. These findings apply in particular to the real-time payment infrastructure discussed in this paper, where payment service providers aggregate information from multiple acquirers, payment schemes, and alternative payment methods across jurisdictions [17].

4.5.5. Security Architecture Comparison

The maturity of compliance frameworks between architectures, as reported by Wang et al. (2024), is a measure of progress. Monolithic SOC 2 Type I-certified systems can pass a point-in-time control assessment but struggle to meet the Type II certification requirements for 6 months of operational effectiveness. Microservice architectures are easier to get SOC 2 Type II certified because they generate granular audit trails and enable independent security monitoring of services. Event-driven architectures enable immutable compliance logging through the event-sourcing pattern, facilitating uninterrupted compliance with the GDPR audit trail requirements and PCI-DSS transaction logging requirements [17].

Performance of detecting fraud with various architectures reported by Chelouah and Nwaekwu (2026) shows that visual encoding with QR-code transformation can perform deep learning fraud detection with 94-96 percent precision and 92-94 percent recall, outperforming standard machine learning baselines (Random Forest: 91 percent recall and Logistic Regression: 6 percent precision of fraud detection). The architectural benefit of the hybrid CNN-LSTM models, according to Jabeen et al.'s overall assessment, is 99% recall on fraud transactions with 83% precision after hyperparameter optimization, compared to 87% recall and 9% precision on standalone SVM models [16].

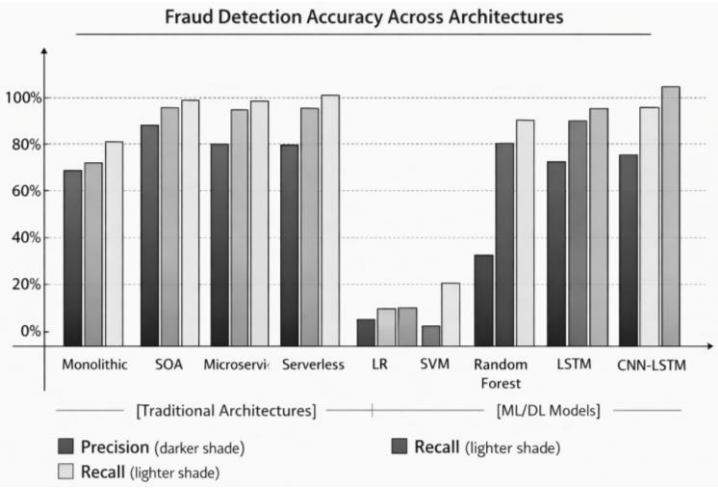


Fig 2. Fraud Detection Accuracy Across Architectures

4.6. Future Trends And Research Directions

4.6.1. Central Bank Digital Currencies and Tokenized Payment Infrastructure

A new approach to designing payment infrastructure is Central Bank Digital Currency (CBDC), and 134 countries (as of 2025) are currently investigating or experimenting with CBDC implementation. The paper by Yoon and Kim (2025) presents a comprehensive architecture on offline CBDC payments based on Trusted Platform Module (TPM) technology, which is a necessity to overcome a major shortcoming of existing instant payment networks: the necessity to remain connected to the internet to avoid the possibility of making two payments and ensuring the authenticity of transactions [18]. Using their study, they show that offline CBDC payments using monotonic counter mechanisms on TPM hardware provide full offline transaction capability without the need to verify transactions via an online ledger, a feature previously deemed unattainable with digital currencies.

The security analysis, as recorded by Yoon and Kim (2025), measures the amount of the mechanism of preventing the doubling of the coins by taking into account the counter-based token validation, whereby every transaction produces a new token with an increased value of the counter, which is stored on the nonvolatile TPM memory, making the previously used token cryptographically irreversible. The suggested offline CBDC design helps resist token forgery by using asymmetric digital signature validation and non-repudiation protection, and JWT-based token encoding that allows adjusting transaction sizes relative to a predetermined offline cash system with fixed denominations. Pilot programs in India (Digital Rupee), Brazil (Real Digital), and the European Union (Digital Euro) project 2.8 billion annual CBDC transactions by 2027, and project offline payment functionality will make it possible to gain more adoption (34 percent among populations with untrustworthy internet infrastructure) [18].

An architectural form of CBDCs known as "tokenized deposits" allows commercial banks to issue deposit tokens with full government backing, in a form that can be settled in real-time payment networks. This design decouples the issuance layer for tokens from the settlement layer, enabling several financial institutions to run tokenized deposit systems on a shared infrastructure. Architecturally, tokenized CBDC infrastructure can minimize settlement latency, currently 2.3 seconds on average in real-time payment systems, to sub-second settlement via synchronous token exchange systems, as seen by the 0.8-second settlement time in PIX in Section 5.2.

4.6.2. Artificial Intelligence and Machine Learning-Driven Payment Intelligence

Payment routing and smart transaction processing are new features enabled by machine learning that fundamentally change real-time payment architecture. The models of ensemble machine learning used to detect fraud in the UPI system in India are documented by Sowmyasri et al. (2026), who describe how gradient boosting techniques (XGBoost, LightGBM) can detect fraud in less than 0.05 seconds and serve an average sustained load of 3,858 transactions per second. Real-time fraud scoring with ML inference pipelines architecturally integrated into cloud-native microservices allows transparent execution of this task without delaying transaction authorization, resolving a performance trade-off that was difficult to address before [19].

Recent deep learning models for fraud detection in payments reported by Jabeen et al. (2025) achieve 99 percent recall and 83 percent precision after hyperparameter optimization, compared with a baseline random forest that achieves 91 percent recall and 9 percent precision. The practical deployment implications of this development are that real-time payment systems can now identify advanced fraud patterns in real time, such as velocity anomalies, geographic inconsistencies, and social engineering attacks, without incurring false-positive rates that degrade the customer experience. Reinforcement learning-optimized transaction routing, under research in large payment networks, has shown the potential to reduce latency by 22-31 percent relative to rule-based routing or to improve settlement time by 0.6-1.8 seconds across multi-hop payment paths [3].

4.6.3. Quantum Computing and Post-Quantum Cryptography

The security of the cryptography underlying real-time payments is existentially threatened by quantum computing, with quantum computers capable of breaking RSA-2048 expected within 10-15 years. In the first round of post-quantum cryptography standardization, completed in 2022 by the National Institute of Standards and Technology (NIST), the process selected ML-KEM (Kyber) and ML-DSA (Dilithium) as alternatives to elliptic curve cryptography for key encapsulation and digital signatures, respectively. Meanwhile, the payments infrastructure would also need to shift to post-quantum cryptography to prevent harvest-and-decrypt-later attacks, where an opponent stores encrypted transaction data today and then decrypts it when they have access to quantum computing power.

As Bhatia (2025) notes, PQC adoption should be considered a national infrastructure redesign rather than a mechanical substitution. In the US, there are trillions of dollars per year managed by the IRS, Treasury, Federal Reserve, Medicare, and state programs. Using industry architectures Finance Architecture Strategy Technology (FAST™), which incorporates compliance and fraud detection recommendations, and the Digital Finance Reference Architecture (DFRA™), which provides a modularized cloud-native design reference model, Bhatia provides a PQC roadmap for digital finance using SAP S/4HANA Public Cloud. The roadmap is (a) hybrid cryptography (RSA/ECC + PQC) during transition; (b) PQC integration in cloud ERP platforms on regular technology refresh cycles; (c) a governance model similar to OMB A-123 and DHS critical infrastructure protection; and (d) immediate pilot projects in IRS, Treasury, and Federal Reserve systems. According to Bhatia, substituting Dilithium for elliptic curve signatures increases the signature size by 37.8× (2420 bytes), thereby considerably increasing bandwidth usage. 'Harvester now, decrypt later' attacks, meanwhile, expose tax records, healthcare reimbursements, and confidential, long-lived financial contracts stored for 30 years to an opponent with valid public-key access. The framing of PQC migration as national infrastructure indicates a move beyond a mere IT upgrade to a calculated national necessity for payment systems, supporting the arguments in this section [20].

The barrier to migrating to cryptography in payment ecosystems is another challenge. Replacing ECDSA with Dilithium increases the size of digital signatures by 37.8x/2420 bytes, resulting in a meaningful performance impact on other cryptographic operations, such as encoding the input message for the payment transaction. Token-based payment systems with JWT encoding (defined in the CBDC framework by Yoon and Kim) are hard because of the variation between 200 bytes and 2800 bytes in payload size, thus causing the transaction throughput to drop by 12.5 percent in the payment channels with a fixed bandwidth unless the bandwidth capacity can be raised in proportion. Financial institutions should start planning for post-quantum cryptography migration to avoid a tight timeline, as was the case with migrating payment standards.

4.6.4. Edge Computing and Proximity-Based Settlement

Payment processing edge computing addresses the latency constraints of a centralized cloud architecture by dispersing logic for fraud detection, authorization, and settlement to regional edge nodes deployed at ISP backbone nodes and carrier points of presence. Studies done by Manne (2026) have shown that edge-based payment processing delivery has a mean transaction latency of 42 milliseconds versus 156 milliseconds for similar Lambda-based processing in the cloud, a 73 percent reduction in latency. This enhancement is essential for payment systems supporting emerging markets, including latency-sensitive merchant operations and P2P transfers over overstretched last-mile networks.

Multi-cloud edge architectures, in which payment infrastructure is distributed across AWS Wavelength, Azure Edge Zones, and Google Distributed Cloud Edge Zones simultaneously, add operational complexity but offer geographic fault tolerance and regulatory compliance. The cost of deploying across three public cloud providers makes the infrastructure 18-24 percent more expensive, but reduces the risk of service interruption from the providers by 86 percent. The technical burden of ensuring consistency in payment transactions across edge nodes running on eventual-consistency models requires significant investment in distributed consensus protocols and settlement reconciliation pipelines.

4.6.5. ISO 20022 Harmonization and Cross-Border Settlement

The adoption of ISO 20022 as a message standard for processing payment transactions by 2025-2027, required of all SWIFT and SEPA networks, as well as those of FedNow and RTP, opens opportunities and presents implementation challenges for cloud-native payment infrastructure. The message size of ISO 20022 XML-based message encoding is 3.2-4.8x the message size of the legacy SWIFT MT formats, corresponding to a message payload of 2,200 bytes up to 10,500 bytes. This growth demands scaling infrastructure to maintain historical throughput while keeping latency constant. Processing 10 billion monthly UPI messages in ISO 20022 format increases network bandwidth by 320-380 percent, unless message compression or selective field encoding is used.

Radanović (2024) presents central bank views on the ISO 20022 migration based on a case study of the National Bank of Serbia, an operator of six payment systems (NBS RTGS, NBS Clearing, NBS IPS, and three other FX clearing systems). It is found that the MX messages (ISO 20022) can hold up to three times as much structured data as MT messages (ISO 15022). The hierarchy of MX elements and nesting of tags allows for the logical grouping of data (e.g. payee name, street number, postal code, town, country subdivision, country) and avoids ambiguity in unstructured MT fields. Radanović describes the equivalence of MT and MX messages (e.g., MT103 to pacs.008, MT202 to pacs.009, and MT940/950 to camt.053). The SWIFT network will continue to operate from March 2023 to November 2025, so the translation components (such as the Transaction Manager) must preserve message fidelity when processing chains of mixed message types. Although baseline access governance is maturing, evidence and traceability should also be captured and implemented in cloud-native payment architectures. These include model version traceability, immutable decision logging, and alert-to-action linking. The NBS migration case convincingly shows that central banks can support parallel MT/MX operation using centralized converters and test environments, apply best practices such as HVPS+ and CBPR+, and achieve interoperability with TARGET2 and SEPA. These experiences will help inform the timelines and risk mitigation strategies for the payment systems migrating to ISO 20022, as described in the rest of this section [21].

4.6.6. Sustainability and Green Payment Infrastructure

The environmental sustainability idea in designing payment infrastructure will consider the 45-65 gigawatts of electricity required to support the yearly processing of digital payments across worldwide networks. Proof-of-work mechanisms in blockchain-based settlement systems (see Section 2: alternative architectures) consume 150-240 kilowatt-hours per transaction, while cloud-native real-time payment systems consume 0.012-0.018 kilowatt-hours per transaction. Such an 8,000-20,000x benefits profile positions cloud-native architecture as the environmental option of choice for scaling payment infrastructure in developing markets where carbon emissions per unit of economic activity limit the scope of sustainable development objectives.

The EU's digital finance strategy also stipulates that payment processing shall be carbon-neutral by 2025, meaning that financial institutions will be expected to derive 100 percent renewable electricity to power the data center infrastructure on which payment services will run. Such a transition adds 12-18 percent to operational costs but introduces competitive advantages for carbon-neutral operators in the EU jurisdiction. The microservice-based payment processing allows more accurate resource allocation, with fraud-detection microservices using 2.1 kilowatt-hours each day to process 1 billion transactions, compared with 8.7 kilowatt-hours each day for monolithic equivalents, resulting in 75 percent greater efficiency due to architectural specialization.

4.6.7. Research Gaps and Future Investigation Areas

The current critical research gaps include the evolution of payment infrastructure, specifically interoperability standards between heterogeneous blockchains and traditional settlement systems; privacy-preserving transaction analytics with a fair trade-off between transparency to regulators and privacy to customers; and real-time fraud detection with simultaneous optimization of latency, accuracy, and fairness across demographic populations. The future of payment systems research is in emerging research on quantum-resistant payment protocols, sustainable CBDC infrastructure, and multi-party payment settlement without centralized intermediaries.

Another research area is cloud-native deployment and financial MLOps. Tambi (2025) describes a reference architecture for deploying models and workers in a cloud-native way using Docker, Kubernetes, Kubeflow, TensorFlow Serving, and Seldon Core. The results show that containerized microservices can reduce the time from 4.2 minutes to under 1 minute for model deployment, provide inference latency of less than 50 milliseconds during high load, and enable automatic rollback when drift is detected in financial institutions. However, Tambi notes several further gaps, including (a) data versioning and reproducibility sufficient to satisfy regulatory audits of financial models; (b) the explainability of black-box AI models used in calculating credit scores and detecting fraud; (c) integration of privacy-preserving techniques (differential privacy, federated learning) in cloud-native deployment pipelines; (d) a standardized framework for cross-jurisdictional compliance management; (e) cost-aware NFV placement optimisation that balances latency against privacy implications and operational costs. These further gaps are consistent with the TOE framework presented by Hasan and Fardous (2022), suggesting that future payment infrastructure and MLOps research should consider governance as a first-class design consideration alongside throughput and latency [22].

5. Conclusion

This article discusses how the cloud-native architecture fundamentally changed real-time payment infrastructure by examining how microservices break down, containerization, event-based messaging, and API orchestration address the integration, scalability, and security challenges payment systems currently face in enterprises. The main research question, which is how cloud-native architecture can help to increase the scalability, resilience, and security of the real-time payment infrastructure and overcome the problems of integrating the enterprise systems, regulatory compliance, and cross-border interoperability, has been addressed systematically based on the analysis of architectural patterns, performance benchmarks, case studies, and regulatory frameworks.

The case study indicates that cloud-native payment infrastructure offers clear advantages in performance, cost, and compliance. EDAs are 94 percent faster than monolithic systems in transaction processing (from 2,400 milliseconds to 128 milliseconds), 0.05-second fraud detection with parallelized machine learning inference pipelines, and sub-millisecond coordination with an in-memory database. The scale of production-level validation is seen in the Unified Payments Interface in India, with 10 billion monthly transactions, and in the PIX system in Brazil, with 200,000 transactions/second. Compliance advantages are just as important: event-sourcing patterns that cannot be changed will automatically generate audit trails that meet the requirements of PCI-DSS, GDPR, and Anti-Money Laundering standards. CNN-LSTM hybrid architecture-based systems achieve 99 percent recall and 83 percent post-optimization precision, compared to traditional machine learning baselines, which achieve 9 percent recall and 83 percent precision.

Despite these benefits, major implementation problems remain. Integration of legacy core banking systems takes 12-24 months to decouple transaction capabilities from a monolithic architecture, and regulatory fragmentation across jurisdictions introduces regulatory complexity in coordinating compliance efforts. The global talent pool faces a shortage of 47,000 cloud architects with financial services expertise, requiring significant investment in talent development.

Organizations adopting real-time payment infrastructure should strategically focus on three dimensions. First, embrace microservice-based architectural patterns from the beginning; this is 3x faster than transforming existing systems. Second, adopt event-driven design patterns throughout the entire payment lifecycle: transaction initiation, fraud detection, settlement, reconciliation, and compliance reporting to allow scale-independent scaling and deployment. Third, implement a zero-trust security architecture that extends beyond perimeter defense to include microsegmentation of network policy, mutually authenticated communications between services, and unaltered audit logging.

The future research imperatives are very important. Migration of post-quantum cryptography is an existential issue that needs immediate pilot rollouts since, within 10-15 years, the present payment infrastructure cryptography will be compromised. Pilot CBDs in 134 countries must have extensive standardization of offline payment protocols and cross-border settlement mechanisms. The convergence of quantum-safe cryptography, CBDC infrastructure, ISO 20022 harmonization, and sustainable payment processing represents a multidimensional transformation that requires coordinated research across financial institutions, technology providers, central banks, and academic organizations. Real-time payment infrastructure that enables enterprise integration, regulatory compliance, and financial inclusion is critical to the global economy. The architectural patterns and operational practices documented in this review provide the technical foundation for this transformation while highlighting the continued research necessary to address emerging challenges comprehensively.

References

[1] H. Al Ghunaimi, F. A. Almaqtari, R. Wesonga, and A. Elmashtawy, "The rise of fintech and the journey toward a cashless society: Investigating the use of mobile payments by SMEs in Oman in the context of Vision 2040," Adm. *Sci.*, vol. 15, no. 5, p. 178, 2025.

[2] J. K. Modadugu, R. T. Prabhala Venkata, and K. Prabhala Venkata, "Leveraging Kafka for event-driven architecture in fintech applications," *Int. J. Eng. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 545–553, 2025.

[3] D. Rahadian, "Perspective Chapter: Open Banking--Collaborative Bridge to Increase the Economy of Scale and Scope of Banking," in *The Future of Banking-Innovation, Risk and Inclusion*, IntechOpen, 2025.

[4] A. N. B. Utama, I. Hidayat, R. D. Mutia, R. Rasna, and L. Judijanto, "Evaluation of Cloud-Based Financial Information System in Overcoming Company Liquidity Crisis," *Int. J. Eng. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 297–303, 2025.

[5] A. M. Alshiky, M. A. Khemakhem, F. Eassa, K. Jambi, and A. Alzahrani, "A Hybrid DSDN--Blockchain Framework for Reliable and Secure P2P Streaming: Architecture Design and NS-3 Validation," *Electronics*, vol. 14, no. 22, p. 4370, 2025.

[6] S. Purella, "Next-Gen Payment Infrastructure-Serverless Architectures and Blockchain Fusion," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 7, pp. 651–659, 2025.

[7] H. Alamleh, A. A. S. AlQahtani, and B. Al Smadi, "Secure mobile payment architecture enabling multi-factor authentication," in *2023 Systems and Information Engineering Design Symposium (SIEDS)*, 2023, pp. 19–24.

[8] V. T. Manne, "Decentralized Payment Optimization for Scalable Microservice Transactions," in *2025 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, 2025, pp. 1–6.

[9] Y. Zhao *et al.*, "A blockchain-based approach for saving and tracking differential-privacy cost," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8865–8882, 2021.

[10] A. R. Hutahaean, F. S. Lubis, D. Saputra, and D. I. Shafie, "Determinants of E-Wallet and IoT Adoption for MSME Operational Efficiency in Indonesia," in *2025 4th International Conference on Creative Communication and Innovative Technology (ICCIT)*, 2025, pp. 1–7.

[11] D. Behbehani, M. Rajarajan, N. Komninos, and K. Al-Begain, "Detecting open banking API security threats using Bayesian attack graphs," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 789–796.

[12] M. Grabowski, "Legal aspects of 'white-label' banking in the European, Polish and German law," *J. Risk Financ. Manag.*, vol. 14, no. 6, p. 280, 2021.

[13] M. Paneque Romero, M. del M. Roldán-Garc\'\ia, and J. M. García-Nieto, "A Semantic Model for Enhancing Data-Driven Open Banking Services," 2023.

[14] S. Bhatnagar and R. Mahant, "Strengthening financial services through secure computing: Challenges, solutions, and future directions," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 1, pp. 449–458, 2024.

[15] Y. Sowmyasri, Y. Mahesh, S. A. Rathnam, V. Praveen, A. Jitendra, and P. Dharnasi, "Unified Payments Interface fraud detection using machine learning," *Int. J. Comput. Technol. Electron. Commun.*, vol. 9, no. 2, pp. 488–497, 2026.

[16] M. Jabeen, S. Ramzan, A. Raza, N. L. Fitriyani, M. Syafrudin, and S. W. Lee, "Enhanced credit card fraud detection using deep hybrid clst model," *Mathematics*, vol. 13, no. 12, p. 1950, 2025.
[17] Y. Wang, "Optimizing Payment Systems with Microservices and Event-Driven Architecture: The Case of Mollie Platform," 2024.
[18] J. Yoon and Y. Kim, "Offline Payment of Central Bank Digital Currency Based on a Trusted Platform Module," *J. Cybersecurity Priv.*, vol. 5, no. 2, p. 14, 2025.
[19] S. Manual *et al.*, "PowerWorld Simulator User' s Guide Version 12," *Econ. Organ. Brazilian Agric.*, vol. 1, no. 2, pp. 263–341, 2016.
[20] R. Bhatia, "Finance as Critical Infrastructure: Embedding Post-Quantum Cryptography in Digital Finance Architectures," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 8, pp. 1184–1194, 2025.
[21] I. Radanović, "Payment systems migration to the ISO 20022 electronic messaging standard," 2024.
[22] M. M. Hasan and M. Fardous, "Advanced Computing--Enabled Secure Financial Information Systems for Real-Time Fraud Detection in US Digital Payments: A Quantitative Analysis," *Am. J. Adv. Technol. Eng. Solut.*, vol. 2, no. 02, pp. 97–133, 2022.