

Vault-Based Secret Management and Zero-Trust Security Model for Distributed Financial Applications

Amir Siddiki

`amir.siddiki2291@gmail.com`

BITWISE INC <https://orcid.org/0009-0003-8989-7095>

Research Article

Keywords: Vault-Based Secret Management, Zero-Trust Security, Distributed Financial Applications, Kubernetes Security, Cloud-Native Microservices, API Authentication, Runtime Monitoring, Identity-Based Access Control, Financial Cybersecurity

Posted Date: June 5th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9916782/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

The rapid growth of distributed financial applications has increased the need for secure secret management mechanisms capable of protecting API credentials, encryption keys, and authentication tokens across cloud-native environments. Conventional credential storage approaches expose enterprise systems to insider threats, unauthorized access, credential leakage, and configuration vulnerabilities. This study proposed a vault-based zero-trust security framework for distributed financial applications operating within Kubernetes and microservice infrastructures. The framework integrated dynamic credential generation, automated secret rotation, encrypted inter-service communication, identity-driven access policies, and runtime monitoring mechanisms for cloud-native financial services. The experimental evaluation analyzed authentication accuracy, secret exposure reduction, transaction integrity, API communication security, and unauthorized access prevention under distributed financial workloads. The results demonstrated improvements in credential protection, runtime visibility, encrypted communication reliability, and policy-controlled access validation compared to traditional static credential architectures. The proposed framework supports secure financial transaction processing and centralized security management within distributed cloud infrastructure.

I. Introduction

Financial institutions increasingly rely on distributed cloud infrastructure, containerized applications, and microservice-based platforms for payment processing, digital banking, fraud analysis, customer management, and transaction monitoring. Modern financial applications operate across hybrid and multi-cloud environments where services exchange authentication tokens, API credentials, encryption certificates, and financial records continuously through interconnected networks. This transition toward distributed computing has improved operational scalability and real-time service delivery, but it has also expanded the attack surface of financial systems. Cybersecurity threats targeting cloud-native financial applications continue to increase due to credential leakage, unauthorized API access, insecure inter-service communication, insider attacks, and lateral movement across distributed infrastructure. Traditional perimeter-oriented security models are no longer sufficient for modern financial environments because they rely heavily on implicit trust relationships and static credential management practices. Many organizations still store sensitive credentials inside configuration files, application containers, deployment scripts, or centralized repositories with limited runtime protection. Once attackers compromise a service account or authentication token, unauthorized access may spread rapidly across interconnected financial services. Distributed microservice environments also create operational challenges related to identity verification, encrypted communication, runtime monitoring, and centralized policy enforcement. Zero-trust security architecture provides an alternative approach where every user, device, service, and communication request remains untrusted until verification is completed. Continuous authentication, policy-controlled access validation, micro-segmentation, and encrypted communication form the foundation of this security model. Vault-based secret management systems also provide centralized protection for credentials, certificates, API keys, and temporary authentication tokens through dynamic secret generation and automated credential rotation. Combining these

approaches creates a security framework capable of protecting distributed financial applications against cloud-based threats while supporting secure transaction processing and controlled service communication across cloud-native infrastructure.

A. Background and Motivation

Modern financial systems depend heavily on distributed cloud infrastructure to support digital banking, online payment processing, fraud detection, transaction monitoring, and customer analytics. Containerized microservices and API-driven communication allow financial institutions to process large transaction volumes across geographically distributed environments. However, this transition has increased dependency on sensitive credentials such as API keys, database passwords, encryption certificates, and authentication tokens. Traditional security architectures often store credentials statically inside application configurations or service containers, creating serious exposure risks when attackers gain unauthorized access. Financial infrastructures also face growing challenges related to ransomware attacks, credential leakage, insider threats, and unauthorized lateral movement within cloud environments. Zero-trust security principles provide an alternative approach where every request undergoes continuous verification before resource access approval. Vault-based secret management systems further reduce risks associated with static credential storage through temporary secret generation and automated credential rotation. These security requirements motivated the development of a centralized framework capable of protecting distributed financial applications operating within cloud-native infrastructure.

B. Problem Statement

Traditional financial application security models rely mainly on perimeter-based defenses and static credential management practices. These approaches assume that systems operating inside trusted network boundaries remain secure, even though modern financial services function across distributed cloud environments with continuous API communication and external service integration. Static credentials stored inside application containers, configuration files, or deployment scripts increase the possibility of credential leakage and unauthorized access. Once attackers compromise a single service or authentication token, lateral movement across interconnected financial systems becomes possible. Existing cloud security implementations also struggle to provide continuous identity verification, dynamic access control, and centralized runtime visibility across distributed microservices. Many organizations maintain fragmented monitoring systems that fail to synchronize authentication logs, API requests, and secret management events effectively. In addition, manual credential rotation and inconsistent policy enforcement create operational complexity within large-scale financial infrastructures. These limitations demonstrate the need for a unified security framework capable of centralized secret management, encrypted communication, runtime monitoring, and continuous access validation within distributed financial applications.

C. Proposed Solution

This study proposes a vault-based secret management and zero-trust security framework designed for distributed financial applications deployed in cloud-native environments. The framework integrates centralized secret storage, policy-controlled access validation, continuous authentication, encrypted inter-service communication, runtime monitoring, and automated credential rotation within Kubernetes-based microservices. HashiCorp Vault manages API keys, database credentials, certificates, and temporary authentication tokens dynamically during runtime instead of storing them permanently inside service containers. The proposed architecture also incorporates mutual TLS communication, identity-aware access control, API gateway filtering, and SIEM-based monitoring to protect financial transaction workflows and inter-service communication. Continuous policy validation verifies user identities, device status, authentication tokens, and service permissions during every request. The framework further applies micro-segmentation policies to restrict unauthorized lateral movement between financial services operating within distributed cloud infrastructure. This integrated security model supports secure financial transaction processing while reducing credential persistence and unauthorized service interaction risks.

D. Contributions

This research introduces a unified security architecture that combines vault-based secret management and zero-trust access control for distributed financial applications. The study presents a centralized secret management mechanism capable of generating temporary credentials dynamically during runtime for Kubernetes-based financial microservices. The proposed framework integrates continuous authentication, encrypted service communication, policy controlled access validation, SIEM-based monitoring, and automated secret rotation within a single distributed security model. Another contribution involves the implementation of runtime monitoring techniques that synchronize authentication traces, API activities, secret rotation records, and anomaly detection logs across cloud-native financial services. The research also evaluates the proposed framework using simulated distributed financial workloads containing transaction processing systems, fraud detection modules, and customer management services. Comparative experimental analysis demonstrates improvements in authentication accuracy, credential protection, unauthorized access prevention, runtime threat detection, and transaction integrity compared to traditional static credential architectures. These contributions provide a practical security framework for modern financial systems operating across distributed cloud infrastructure.

E. Paper Organization

The remainder of this paper is organized into several sections. Section II reviews previous studies related to zero-trust security, secret management, cloud-native infrastructure protection, and distributed financial system security. Section III describes the proposed methodology, including the system

architecture, authentication workflow, vault-based secret management model, experimental implementation, and evaluation metrics. Section IV presents the experimental results and discussion, focusing on authentication performance, credential protection, runtime monitoring, encrypted communication, and transaction integrity analysis. Section V discusses the limitations associated with the proposed framework and implementation environment. Finally, Section VI concludes the paper and presents possible directions for future research involving AI-assisted threat analysis, post-quantum cryptography, hybrid cloud deployment, and enterprise-scale financial infrastructure protection.

The primary objective of this research is to develop and evaluate a vault-based zero-trust security framework for distributed financial applications operating in cloud-native environments. The study aims to reduce credential exposure risks, unauthorized access incidents, insecure inter-service communication, and lateral movement across Kubernetes-based financial microservices. Another objective involves implementing centralized secret management with temporary credential generation and automated secret rotation for financial transaction systems. The research also seeks to improve runtime visibility, authentication reliability, encrypted communication, and policy-controlled access validation within distributed cloud infrastructure. Through experimental evaluation, the study analyzes the effectiveness of the proposed framework in improving authentication accuracy, transaction integrity, runtime monitoring capability, and operational security performance compared to conventional static credential architectures.

II. Related Work

Distributed financial applications require protected credential management, authenticated communication, runtime monitoring, and controlled access across cloud-native infrastructures. Researchers have examined zero-trust security, federated learning, cloud analytics, anomaly detection, and enterprise monitoring to address security and privacy issues in distributed systems. Existing studies provide the foundation for vault-based secret management and zero-trust security frameworks in financial environments.

A. Cybersecurity and Zero-Trust Security Architectures

Hossain et al. [1] examined cybersecurity and privacy challenges in IoT-enabled ecosystems and discussed communication protection, authentication models, and distributed data security mechanisms. Akhter [2] presented algorithmic internal control methods using MIS event logs for monitoring organizational activities and detecting abnormal operational behavior. Ashogbon and Ukpere [6] evaluated Zero Trust Architecture (ZTA) adoption in banking environments and reported that continuous verification and least-privilege access reduce unauthorized system activity. Singh et al. [7] proposed a blockchain-supported zero-trust framework for FinTech ecosystems to address insider threats and cyber attacks through decentralized verification mechanisms. Thotempudi et al. [9] introduced the Lockbox architecture for secure cloud workload processing using isolated secret storage and protected execution

environments. Zaman [10] discussed vulnerability management and automated incident response strategies for enterprise network protection.

B. Privacy-Preserving Analytics and Financial Risk Management

Financial applications process sensitive customer and transactional information across distributed platforms. Rahman et al. [3] proposed a federated learning framework that supports distributed analytics without exposing raw enterprise data. Islam [4] analyzed AI and big data techniques for predictive analytics and discussed secure data governance requirements in enterprise systems. Rahman [11] developed a financial risk management system that combines federated learning with privacy-preserving AI techniques for fraud monitoring and secure decision support in SMEs. Hasan [5] examined big data-driven business process optimization methods for enterprise analytics and operational analysis. These studies indicate that distributed analytics and privacy-preserving computation can support secure financial operations in cloud-based environments.

C. Cloud Infrastructure Security and Resource Governance

Distributed financial systems rely on cloud platforms for application deployment, workload orchestration, and infrastructure management. Afrin [8] developed cloud-integrated monitoring dashboards using IoT and edge analytics for operational observation and infrastructure tracking. Farooq [12] proposed backup and disaster recovery automation methods for hybrid cloud environments to support service continuity during infrastructure failures. Farooq [13] also investigated resource utilization analytics for cloud infrastructure management and runtime monitoring. Afrin [15] discussed cyber-resilient infrastructure models with automated threat detection for enterprise network systems. Hasan [16] analyzed SAP-centric financial workloads in virtualized cloud environments and presented AI-supported CloudOps techniques for operational management. These studies demonstrate the importance of runtime monitoring, infrastructure visibility, and access management in distributed enterprise systems.

D. Intelligent Monitoring and Enterprise Security Analytics

Enterprise financial applications require continuous monitoring, threat analysis, and operational intelligence for secure service delivery. Hasan [14] proposed machine learning-based KPI forecasting methods for finance and operations teams to support predictive enterprise monitoring. Rahman F. [17] applied data science methods for risk assessment and management in critical infrastructure systems and demonstrated the value of predictive analysis for operational reliability. Rahman T. [18] presented a financial threat monitoring framework for real-time fraud detection and intelligent security analysis in enterprise financial systems. Zaman et al. [19] proposed a resilient edge computing framework with autonomous and secure operational models for distributed environments. Nahar et al. [20] developed an AI-integrated analytics platform for enterprise ESG compliance and operational data management. Previous studies addressed analytics, monitoring, cloud security, and access control separately;

however, limited research integrated vault-based secret management with zero-trust authentication and encrypted service communication for distributed financial applications. The current study addresses this gap through a framework that combines centralized secret storage, dynamic credential rotation, policy-driven authentication, runtime monitoring, and secure inter-service communication within cloud-native financial infrastructures.

III. Methodology

This study presents a vault-based secret management and zero-trust security framework for distributed financial applications deployed in cloud-native environments. The framework combines centralized secret storage, identity-driven authentication, encrypted service communication, runtime monitoring, and policy-controlled access validation within a Kubernetes-based microservice infrastructure. The proposed model addresses credential exposure, unauthorized access, lateral movement, and insecure inter-service communication in distributed financial systems. The experimental environment simulated a distributed financial platform containing payment processing services, customer management systems, fraud detection modules, reporting services, and transaction monitoring components deployed across containerized cloud infrastructure.

A. System Architecture Design

The proposed architecture follows a layered zero-trust structure where each communication request passes through continuous authentication and policy validation before resource access approval. The infrastructure combines Kubernetes orchestration, API gateway filtering, identity and access management services, encrypted service communication, and centralized vault-based secret handling. Application containers and configuration files do not store credentials permanently. Instead, services retrieve temporary secrets dynamically during runtime through HashiCorp Vault. Figure 1 presents the complete architecture of the proposed framework.

The architecture separates financial services through micro-segmentation and encrypted communication policies. Each service communicates through mutual TLS channels generated dynamically using Vault PKI services. Continuous monitoring modules record authentication logs, API requests, and suspicious activities for anomaly analysis.

B. Zero-Trust Authentication Workflow

The authentication workflow applies continuous verification to users, services, devices, and transaction requests throughout active sessions. The system validates identity credentials, device status, access policies, and runtime behavior before granting access to financial services. Temporary tokens and certificates are generated dynamically to reduce long-term credential exposure. Figure 2 illustrates the authentication and vault interaction workflow.

The workflow reduces static credential dependency because tokens and secrets expire automatically after predefined intervals. Access requests failing policy validation are rejected before resource interaction occurs.

C. Secret Management and Encryption Model

The proposed framework uses centralized vault-based secret management for financial microservices operating across distributed Kubernetes nodes. Sensitive credentials such as API keys, database passwords, certificates, and authentication tokens remain encrypted during storage and transmission. Services retrieve temporary secrets from Vault using authenticated service identities. Mutual TLS encryption protects communication between distributed services. Vault-generated certificates create short-lived encrypted sessions that reduce credential persistence risks. Automated secret rotation mechanisms periodically revoke and regenerate credentials to limit exposure windows.

The secret exposure probability is calculated using:

$$P_s = \frac{N_e}{N_t}$$

Where:

P_s represents the secret exposure probability, N_e represents the number of exposed secrets, and N_t represents the total number of active secrets in the system.

Lower values of P_s indicate stronger protection against credential leakage and unauthorized secret disclosure.

The authentication validation score is calculated as:

$$A_v = \frac{M_a + D_v + P_c}{3}$$

Where:

A_v represents the authentication validation score, M_a represents multi-factor authentication status, D_v represents the device verification score, and P_c represents the policy compliance score. The system grants access only when the calculated validation score exceeds the defined security threshold.

D. Experimental Environment and Implementation

The experimental environment used Kubernetes-based container orchestration across distributed cloud nodes. Financial transaction services, customer management systems, fraud analysis modules, and reporting platforms operated as independent Docker microservices. The infrastructure integrated HashiCorp Vault for centralized secret management, Istio service mesh for encrypted communication, OAuth2 and JWT services for identity validation, and Prometheus-Grafana monitoring for runtime

observation. SIEM logging modules collected authentication traces, API activities, transaction records, and anomaly events throughout the experiments.

Table 1
Experimental Environment Configuration

Component	Technology Used	Purpose
Container Orchestration	Kubernetes	Microservice deployment
Secret Management	HashiCorp Vault	Dynamic secret storage
Service Communication	Istio mTLS	Encrypted traffic
Authentication	OAuth2 + JWT	Identity verification
Monitoring	Prometheus + Grafana	Runtime monitoring
Logging System	SIEM	Threat analysis
API Security	API Gateway	Request filtering
Financial Services	Docker Microservices	Transaction processing

The experiments generated concurrent transaction requests, login attempts, API interactions, and inter-service communication events to evaluate system behavior under distributed financial workloads.

E. Performance Evaluation Metrics

The framework evaluation used security-oriented and operational performance metrics. The experiments measured authentication success rates, unauthorized access detection, credential exposure reduction, secret rotation efficiency, API latency, and transaction validation consistency.

The transaction integrity rate was calculated using:

$$T_i = \frac{T_v}{T_r} \times 100$$

Where:

T_i represents the transaction integrity percentage, T_v represents the number of validated transactions, and T_r represents the total transaction requests processed during the experiment.

Higher values of T_i indicate stronger transaction consistency and operational reliability within the distributed financial infrastructure. The experimental analysis compared traditional static credential systems with the proposed vault-based zero-trust framework under identical transaction workloads. Authentication logs, monitoring records, and secret rotation traces were analyzed to identify improvements in distributed financial application security and credential protection.

Future research may include AI-assisted behavioral threat analysis, adaptive policy management, and post-quantum cryptographic mechanisms for financial systems. Additional investigation can examine enterprise-scale deployment, hybrid legacy-cloud integration, blockchain-supported transaction validation, and hardware-assisted trusted execution environments. Further experiments may also evaluate distributed denial-of-service resistance, supply-chain attack prevention, and cross-region financial transaction security under real-world banking workloads.

IV. Discussion and Results

The experimental evaluation examined the performance of the proposed vault-based zero-trust framework in distributed financial applications operating within cloud-native infrastructure. The analysis focused on authentication accuracy, credential exposure reduction, unauthorized access prevention, encrypted service communication, runtime monitoring, and transaction reliability. A conventional static credential architecture was used as the baseline model during comparative evaluation. The experimental environment generated concurrent API requests, financial transactions, authentication sessions, and inter-service communication events across Kubernetes-based microservices. Monitoring modules collected SIEM records, authentication traces, transaction logs, and secret rotation events during all test phases.

A. Authentication and Access Control Performance

The first experiment examined the authentication and access validation mechanism across distributed financial services. The proposed framework applied multi-factor authentication, device validation, policy verification, temporary token generation, and continuous session analysis during every transaction request. The results showed a major reduction in unauthorized access incidents compared to the traditional security model. Temporary authentication tokens and dynamic access policies limited credential reuse during repeated API requests.

Table 2
Authentication and Access Control Performance

Security Metric	Traditional Security Model	Proposed Zero-Trust Framework
Unauthorized Access Attempts	74	11
Authentication Success Rate	89.2%	97.8%
Privilege Escalation Incidents	18	2
Session Hijacking Attempts	13	1
Average Authentication Delay	148 ms	173 ms
Policy Violation Detection Rate	76.5%	96.1%

The proposed framework increased authentication delay slightly because each request passed through several validation stages. Despite the additional processing overhead, the delay remained acceptable for distributed financial systems requiring strict access validation. Monitoring records identified abnormal login patterns, unauthorized device activity, suspicious IP requests, and expired session attempts before financial service interaction occurred.

The comparison indicates that the proposed framework produced higher authentication accuracy, stronger unauthorized access prevention, improved credential rotation performance, and lower secret exposure probability than the conventional architecture.

B. Secret Management and Credential Protection Analysis

The second experiment evaluated centralized vault-based secret management across distributed financial microservices. The traditional environment stored API keys, database passwords, and authentication credentials inside configuration files and application containers. The proposed framework retrieved temporary secrets dynamically from HashiCorp Vault during runtime. The results showed a major reduction in credential exposure incidents after applying centralized secret management and automated credential rotation policies. Temporary tokens expired automatically after predefined intervals, limiting long-term credential persistence. Automated secret rotation completed successfully without interrupting transaction processing or encrypted service communication. Runtime secret retrieval maintained stable operation under high transaction workloads because Vault distributed updated credentials dynamically across Kubernetes services. The proposed framework reduced credential exposure probability significantly compared to the static credential architecture. Centralized secret storage, temporary token generation, and automatic credential expiration limited unauthorized credential reuse inside distributed financial services.

C. Runtime Monitoring and Threat Detection

The third experiment examined runtime monitoring and threat detection performance within distributed financial infrastructure. The SIEM monitoring engine collected authentication logs, API request traces, token validation reports, network communication records, and policy violation events throughout the experimental process. Behavioral monitoring modules analyzed transaction frequency, failed login attempts, abnormal service communication patterns, and suspicious API activity. The proposed framework identified unauthorized API scanning attempts, expired token usage, and abnormal transaction behavior earlier than the conventional architecture. The average threat detection time decreased from 14.6 minutes to 3.2 minutes after applying continuous monitoring and centralized log analysis.

The monitoring framework provided centralized visibility across Kubernetes nodes, API gateways, Vault authentication systems, and financial transaction services. Security administrators received

synchronized event records through a unified monitoring interface. The reduction in detection time improved incident response capability because suspicious activities were isolated before attackers reached additional financial services or transaction records.

D. Microservice Communication and Transaction Integrity

The fourth experiment evaluated encrypted communication and transaction reliability across distributed financial microservices. The framework used Istio service mesh with mutual TLS communication generated through Vault PKI services. Encrypted service communication restricted unauthorized interaction between payment systems, fraud detection modules, customer databases, and reporting services. Runtime monitoring logs recorded encrypted API exchanges and service identity validation during every communication session. More than 98% of transaction requests completed successfully under concurrent workloads containing thousands of distributed API interactions. The transaction integrity percentage remained stable during high-volume transaction processing. The framework maintained strong transaction consistency because encrypted communication and continuous identity validation reduced unauthorized service access and session manipulation attempts. Micro-segmentation policies restricted lateral movement between Kubernetes services. Attack simulations targeting isolated financial modules failed because service identities and communication certificates did not satisfy policy requirements.

E. Limitations of the Study

The study used a simulated distributed financial environment rather than a production-scale banking infrastructure. Real-world systems may involve larger transaction volumes, wider geographic distribution, and stricter compliance requirements. The implementation focused on Kubernetes-based microservices and cloud-native deployment models. Legacy financial systems and hybrid infrastructures were not included in the experiments. Continuous authentication and repeated policy validation introduced additional computational overhead during high transaction workloads. The framework also did not evaluate large-scale DDoS attacks, supply-chain attacks, post-quantum cryptography, or AI-driven threat prediction mechanisms.

F. Comparative Discussion

The overall experimental analysis indicates that the proposed vault-based zero-trust framework provided stronger security protection than conventional static credential architectures in distributed financial applications. Continuous authentication, temporary secret generation, encrypted service communication, and policy-controlled access validation reduced credential exposure and unauthorized access incidents significantly. Automated secret rotation removed long-term credential persistence risks commonly observed in static secret architectures. The framework performed effectively in distributed microservice

environments containing high transaction volumes and continuous inter-service communication workflows. Runtime monitoring modules improved threat visibility and reduced incident response time through centralized SIEM analysis and synchronized authentication logging. Although the framework introduced moderate authentication overhead, the processing delay remained acceptable for financial systems requiring strict identity validation and transaction security. The results also demonstrated that micro-segmentation and service-level authentication restricted lateral movement across Kubernetes infrastructure. The integration of centralized vault-based secret management with zero-trust policy validation improved credential protection, runtime visibility, transaction reliability, and distributed financial application security.

V. Conclusion

This study presented a vault-based secret management and zero-trust security framework for distributed financial applications deployed in cloud-native environments. The proposed architecture integrated centralized secret storage, continuous authentication, encrypted inter-service communication, policy-controlled access validation, and runtime monitoring within Kubernetes-based microservices. Experimental evaluation showed improvements in authentication accuracy, credential protection, unauthorized access prevention, transaction integrity, and threat detection compared to conventional static credential architectures. Automated secret rotation and temporary credential generation reduced long-term secret exposure risks, while micro-segmentation and mutual TLS communication restricted unauthorized lateral movement across distributed financial services. The results demonstrate that the proposed framework supports secure financial transaction processing and controlled service communication within distributed cloud infrastructure.

References

1. Hossain MT, Nabil SH, Razaq A, Rahman M (2025) Cybersecurity and privacy in IoT-based electric vehicle ecosystems. *IJSRED – Int J Sci Res Eng Dev* 8(2):921–933.
<https://doi.org/10.5281/zenodo.17246184>
2. Akhter T (2025), October 6 Algorithmic internal controls for SMEs using MIS event logs. *TechRxiv*.
<https://doi.org/10.36227/techrxiv.175978941.15848264.v1>
3. Rahman M, Haque S, Al Sany SMA (2025) Federated learning for privacy-preserving apparel supply chain analytics. *World J Adv Eng Technol Sci* 17(1):259–270.
<https://doi.org/10.30574/wjaets.2025.17.1.1386>
4. Islam R (2025) AI and big data for predictive analytics in pharmaceutical quality assurance. *SSRN*.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5564319
5. Hasan E (2025) Big Data-Driven Business Process Optimization: Enhancing Decision-Making Through Predictive Analytics. *TechRxiv*. <https://doi.org/10.36227/techrxiv.175987736.61988942/v1>
6. Ashogbon A, Ukpere O (2025) Evaluating the application of Zero Trust Architecture (ZTA) implementation in Nigeria's banking industry. *J Electr Electron Eng* 13(3):131–142.

<https://doi.org/10.11648/j.jeee.20251303.12>

7. Singh A, Pareek V, Sharma A (2025) Blockchain-enabled zero trust framework for securing FinTech ecosystems against insider threats and cyber attacks. arXiv.
<https://doi.org/10.48550/arXiv.2507.19976>
8. Afrin S (2025) Cloud-integrated network monitoring dashboards using IoT and edge analytics. IJSRED – Int J Sci Res Eng Dev 8(5):2298–2307. <https://doi.org/10.5281/zenodo.17536343>
9. Thotempudi VK, Agarwal M, Batta R, Mangal A (2026) Lockbox: A zero trust architecture for secure processing of sensitive cloud workloads. arXiv. <https://doi.org/10.48550/arXiv.2603.09025>
10. Zaman SU (2025) Vulnerability management and automated incident response in corporate networks. IJSRED – Int J Sci Res Eng Dev 8(5):2275–2286.
<https://doi.org/10.5281/zenodo.17536305>
11. Rahman M, Preserving AI, Techniques (2025) IJSRED – Int J Sci Res Eng Dev, 8(6), 1041–1052.
<https://doi.org/10.5281/zenodo.17769869>
12. Farooq H (2025) Cross-platform backup and disaster recovery automation in hybrid clouds. Int J Sci Innov Eng 2(11):220–242. <https://doi.org/10.70849/IJSCI02112025025>
13. Farooq H (2025) Resource utilization analytics dashboard for cloud infrastructure management. World J Adv Eng Technol Sci 17(2):141–154. <https://doi.org/10.30574/wjaets.2025.17.2.1458>
14. Hasan E (2025) Machine learning-based KPI forecasting for finance and operations teams. IJSRED – Int J Sci Res Eng Dev 8(6):2139–2149. <https://doi.org/10.5281/zenodo.17926746>
15. Afrin S (2025) Cyber-resilient infrastructure for public internet service providers using automated threat detection. World J Adv Eng Technol Sci 17(2):127–140.
<https://doi.org/10.30574/wjaets.2025.17.2.1475>
16. Hasan E (2025) Optimizing SAP-centric financial workloads with AI-enhanced CloudOps in virtualized data centers. IJSRED – Int J Sci Res Eng Dev 8(6):2252–2264.
<https://doi.org/10.5281/zenodo.17926855>
17. Rahman F (2025) Data science in power system risk assessment and management. World J Adv Eng Technol Sci 17(3):295–311. <https://doi.org/10.30574/wjaets.2025.17.3.1560>
18. Rahman T (2026) Financial Risk Intelligence: Real-Time Fraud Detection and Threat Monitoring. Zenodo. <https://doi.org/10.5281/zenodo.18176490>
19. Zaman SU, Afrin S, Zaidi SKA, Islam KSA (2026) Resilient edge computing framework for autonomous, secure, and energy-aware systems. World J Adv Eng Technol Sci 18(1):105–121.
<https://doi.org/10.30574/wjaets.2026.18.1.1577>
20. Nahar S, Rahman F, Mim MA (2026) AI-integrated renewable energy and data analytics platform for corporate ESG compliance. World J Adv Eng Technol Sci 18(1):219–235.
<https://doi.org/10.30574/wjaets.2026.18.1.0031>
21. Rahman F, Nahar S, Mim MA (2026) Cloud-native enterprise resource management for multi-sector operations. Global J Eng Technol Adv 26(1):126–141.

- <https://doi.org/10.30574/gjeta.2026.26.1.0012>
22. Sharan SMMI, Fahim MAI, Farooq H (2026) Cloud native fintech analytics platform for IoT enabled retail networks. *World J Adv Eng Technol Sci* 18(1):089–104.
<https://doi.org/10.30574/wjaets.2026.18.1.1582>
 23. Fahim MAI, Farooq H, Sharan SMMI (2026) AI-powered IoT security framework using blockchain and cloud integration. *Global J Eng Technol Adv* 26(1):168–185.
<https://doi.org/10.30574/gjeta.2026.26.1.0003>
 24. Islam R (2026) AI-Integrated Management Information Systems for Manufacturing and Supply Chain Risk Mitigation. Zenodo. <https://doi.org/10.5281/zenodo.18349501>
 25. Zaidi SKA, Islam KSA, Zaman SU, Afrin S (2026) Blockchain-secured communication for industrial IoT and aviation control systems. *IJSRED – Int J Sci Res Eng Dev* 9(1):234–250.
<https://doi.org/10.5281/zenodo.18278261>
 26. Islam KSA, Zaidi SKA, Afrin S, Zaman SU (2026) Federated learning for secure industrial automation and grid optimization. *Global J Eng Technol Adv* 26(1):025–040.
<https://doi.org/10.30574/gjeta.2026.26.1.0360>
 27. Mim MA (2026) Cybersecurity risk mitigation in educational and healthcare IT environments. Zenodo. <https://doi.org/10.5281/zenodo.18988585>
 28. Alam MJ (2026) Information systems for legal documentation management and policy analysis in institutional governance. Figshare. <https://doi.org/10.6084/m9.figshare.31717873>
 29. Alam MJ (2026) Digital contract lifecycle management systems for corporate governance and regulatory oversight. Zenodo. <https://doi.org/10.5281/zenodo.19007705>
 30. Dukkipati SSNC (2026) Cloud-native big data streaming framework for real-time social media intelligence and large-scale public opinion analytics. Zenodo.
<https://doi.org/10.5281/zenodo.19274669>
 31. Bhuiyan MIH (2026) AI-driven customer complaint analytics for systemic risk reduction and consumer protection in the U.S. banking sector. Zenodo. <https://doi.org/10.5281/zenodo.19344701>
 32. Hossain MI (2026) Zenodo. <https://doi.org/10.5281/zenodo.19441203>. Enabled digitalization of container vessel navigation and U.S. port operations using real-time AIS and operational data for shipping and supply chain optimization
 33. Shaik MH (2026) Secure and compliant DevSecOps architecture for automated CI/CD pipelines in regulated cloud environments. Zenodo. <https://doi.org/10.5281/zenodo.19442346>
 34. Junaid E (2026) Secure Kubernetes platform engineering for EKS and GKE: Helm-based standardization and runtime governance. Zenodo. <https://doi.org/10.5281/zenodo.19483524>
 35. Islam MA, Tariq F, Shaik MH, Mirza SB (2026) Identity-centric security models for enterprise web systems. *Saudi J Eng Technol* 11(4):237–246. <https://doi.org/10.36348/sjet.2026.v11i04.008>
 36. Mirza SB, Islam MA, Tariq F, Shaik MH (2026) Diagnostic analytics for enterprise reporting platforms. *Saudi J Eng Technol* 11(4):247–256. <https://doi.org/10.36348/sjet.2026.v11i04.009>

37. Hasan E (2025) Explainable predictive analytics for healthcare decision support. *Int J Sci Innov Eng* 2(10):921–938. <https://doi.org/10.70849/IJSCI02102025105>
38. Khayyam F (2026) Resilient identity and access governance architecture for artificial intelligence-enabled software-as-a-service ecosystems. *Saudi J Eng Technol* 11(4):276–284. <https://doi.org/10.36348/sjet.2026.v11i04.012>
39. Alam MJ (2026) Regulatory reporting information systems for financial and institutional compliance monitoring. Preprints. <https://doi.org/10.20944/preprints202603.1198.v1>
40. Khayyam F (2026) Scalable trust and auditability for generative AI in enterprise CRM platforms: A framework and reference architecture. Zenodo. <https://doi.org/10.5281/zenodo.19607181>
41. Dukkupati SSNC (2026) Design of cloud-native distributed systems for high-availability, multi-region, and multi-currency digital enterprise platforms. Zenodo. <https://doi.org/10.5281/zenodo.19641600>
42. Islam MA (2026) Native scalable student information systems and admission test automation with peak load architecture database performance optimization and observability driven reliability. Zenodo. <https://doi.org/10.5281/zenodo.18987480>
43. Siddiki A (2026) Adaptive Cloud-Native Migration Framework for Secure Enterprise Financial Systems Using OpenShift and Distributed Microservices. Zenodo. <https://doi.org/10.5281/zenodo.20070714>
44. Hossain MT, Nabil SH, Rahman M, Razaq A (2025) Data analytics for IoT-driven EV battery health monitoring. *IJSRED – Int J Sci Res Eng Dev* 8(2):903–913. <https://doi.org/10.5281/zenodo.17246168>
45. Rahman M, Razaq A, Hossain MT, Zaman MTU (2025) Machine learning approaches for predictive maintenance in IoT devices. *World J Adv Eng Technol Sci* 17(1):157–170. <https://doi.org/10.30574/wjaets.2025.17.1.1388>
46. Rahman M (2025), October 15 Integrating IoT and MIS for Last-Mile Connectivity in Residential Broadband Services. *TechRxiv*. <https://doi.org/10.36227/techrxiv.176054689.95468219/v1>
47. Islam R (2025), October 15 Integration of IIoT and MIS for smart pharmaceutical manufacturing. *TechRxiv*. <https://doi.org/10.36227/techrxiv.176049811.10002169>
48. Haque S, Al Sany SMA, Rahman M (2025) Circular economy in fashion: MIS-driven digital product passports for apparel traceability. *Int J Sci Res Eng Dev* 8(5):1254–1262. <https://doi.org/10.5281/zenodo.17276038>
49. Al Sany SMA, Haque S, Rahman M (2025) Green apparel logistics: MIS-enabled carbon footprint reduction in fashion supply chains. *Int J Sci Res Eng Dev* 8(5):1263–1272. <https://doi.org/10.5281/zenodo.17276049>
50. Al Sany SMA, Rahman M, Haque S (2025) ERP–MIS Integration for Intelligent Apparel Production Planning. *World J Adv Eng Technol Sci* 17(1):145–156. <https://doi.org/10.30574/wjaets.2025.17.1.1387>
51. Akhter T (2025), October 6 MIS-enabled workforce analytics for service quality & retention. *TechRxiv*. <https://doi.org/10.36227/techrxiv.175978943.38544757.v1>

52. Hasan E (2025) SQL-driven data quality optimization in multi-source enterprise dashboards. IJSRED – Int J Sci Res Eng Dev 8(6):2150–2160. <https://doi.org/10.5281/zenodo.17926758>
53. Haque S, Sany A (2025) Impact of Consumer Behavior Analytics on Telecom Sales Strategy. Int J Sci Innov Eng 2(10):998–1018. <https://doi.org/10.70849/IJSCI02102025114>
54. Islam KSA (2025) Implementation of safety-integrated SCADA systems for process hazard control in power generation plants. IJSRED – Int J Sci Res Eng Dev 8(5):2321–2331. <https://doi.org/10.5281/zenodo.17536369>
55. Rahman F (2025) Advanced statistical models for forecasting energy prices. Global J Eng Technol Adv 25(3):168–182. <https://doi.org/10.30574/gjeta.2025.25.3.0350>
56. Hossain MT (2025) Cost negotiation strategies and their impact on profitability in fashion sourcing: A quantitative analysis. Global J Eng Technol Adv 25(3):136–152. <https://doi.org/10.30574/gjeta.2025.25.3.0348>
57. Fahim MAI, Sharan SMMI, Farooq H (2025) AI-enabled cloud-IoT platform for predictive infrastructure automation. World J Adv Eng Technol Sci 17(3):431–446. <https://doi.org/10.30574/wjaets.2025.17.3.1574>
58. Rahman M, Alam MS, Sany A, S. M. A., Nahar S (2026) Enterprise AI driven data management framework for cross industry decision intelligence and operational optimization. SSRN Electron J. <https://doi.org/10.2139/ssrn.6330258>
59. Nahar S, Sany SMAA, Rahman M, Alam MS (2026) Digital twin-based AI data management system for real-time industrial and business process optimization. SSRN Electron J. <https://ssrn.com/abstract=6337579>
60. Mirza SB (2026) Predictive Reliability Engineering for Cloud Scale Business Intelligence Platforms through Anomaly Detection Capacity Optimization and Proactive Support Automation. *Zenodo*. <https://doi.org/10.5281/zenodo.19474845>
61. Mim M, Akter M (2026) Integration of business intelligence dashboards in administrative decision making. SSRN Electron J. <https://doi.org/10.2139/ssrn.6402899>
62. Alam MJ (2026) Digital compliance management systems for regulatory documentation and risk monitoring in organizational operations. SSRN Electron J. <https://doi.org/10.2139/ssrn.6410539>
63. Islam KSA (2025) Transformer protection and fault detection through relay automation and machine learning. IJSRED – Int J Sci Res Eng Dev 8(5):2308–2320. <https://doi.org/10.5281/zenodo.17536362>
64. Alam MJ (2026) Digital evidence management systems for legal case documentation and investigation records. SSRN Electron J. <https://doi.org/10.2139/ssrn.6416718>
65. Dukkupati SSNC (2026), February 9 Design and implementation of scalable AI-driven conversational systems for enterprise-level feedback intelligence and decision support. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6263818
66. Rahman M (2025), October 15 IoT-enabled smart charging systems for electric vehicles. *TechRxiv*. <https://doi.org/10.36227/techrxiv.176049766.60280824/v1>

67. Alam MS (2025) October 21). AI-driven sustainable manufacturing for resource optimization. TechRxiv. <https://doi.org/10.36227/techrxiv.176107759.92503137/v1>
68. Alam MS (2025), October 21 Data-driven production scheduling for high-mix manufacturing environments. *TechRxiv*. <https://doi.org/10.36227/techrxiv.176107775.59550104/v1>
69. Hossain MT (2025), October 7 Smart inventory and warehouse automation for fashion retail. *TechRxiv*. <https://doi.org/10.36227/techrxiv.175987210.04689809.v1>
70. Al Sany SMA (2025) The role of data analytics in optimizing budget allocation and financial efficiency in startups. *IJSRED – Int J Sci Res Eng Dev* 8(5):2287–2297. <https://doi.org/10.5281/zenodo.17536325>
71. Haque S (2025) The impact of automation on accounting practices. *IJSRED – Int J Sci Res Eng Dev* 8(6):2312–2323. <https://doi.org/10.5281/zenodo.18074324>
72. Alam MJ (2026) Policy monitoring platforms for institutional governance and organizational accountability. *SSRN Electron J*. <https://doi.org/10.2139/ssrn.6416238>
73. Bhuiyan MIH, Akter T, Afroje S, Chokder R (2026) Operational risk indicators derived from customer interaction data in digital banking platforms. *Saudi J Eng Technol* 11(4):266–275. <https://doi.org/10.36348/sjet.2026.v11i04.011>
74. Zaman SU (2025) Enhancing security in cloud-based IAM systems using real-time anomaly detection. *IJSRED – Int J Sci Res Eng Dev* 8(6):2292–2304. <https://doi.org/10.5281/zenodo.17926883>
75. Rahman M (2025) Predictive maintenance of electric vehicle components using IoT sensors. *World J Adv Eng Technol Sci* 17(3):312–327. <https://doi.org/10.30574/wjaets.2025.17.3.1557>
76. Razaq A, Zaman MTU, Karim MA (2026) Predictive maintenance framework for solar inverters and smart grid assets using edge AI and advanced fault analytics. *Saudi J Eng Technol* 11(4):285–293. <https://doi.org/10.36348/sjet.2026.v11i04.013>
77. Junaid E (2026) Container image hardening pipeline: Vulnerability scanning, policy gates, and runtime efficiency in production Kubernetes. *Zenodo*. <https://doi.org/10.5281/zenodo.20026854>
78. Farabi SA (2025) AI-driven predictive maintenance model for DWDM systems to enhance fiber network uptime in underserved U.S. regions. *Preprints*. <https://doi.org/10.20944/preprints202506.1152.v1>
79. Farabi SA (2025) AI-powered design and resilience analysis of fiber optic networks in disaster-prone regions. *ResearchGate*. <https://doi.org/10.13140/RG.2.2.12096.65287>
80. Nahar S (2025) Optimizing HR management in smart pharmaceutical manufacturing through IIoT and MIS integration. *World J Adv Eng Technol Sci* 17(3):240–252. <https://doi.org/10.30574/wjaets.2025.17.3.1554>
81. Hossain MT (2025) AI-Augmented Sensor Trace Analysis for Defect Localization in Apparel Production Systems Using OTDR-Inspired Methodology. *IJSRED – Int J Sci Res Eng Dev* 8(6):1029–1040. <https://doi.org/10.5281/zenodo.17769857>

82. Farabi SA (2025) AI-augmented OTDR fault localization framework for resilient rural fiber networks in the United States. *arXiv*. <https://arxiv.org/abs/2506.03041>
83. Rabbi MS (2026) AI-Driven SCADA Grid Intelligence for Predictive Fault Detection, Cyber Health Monitoring, and Grid Reliability Enhancement. *Zenodo*. <https://doi.org/10.5281/zenodo.18196487>
84. Zaman MT (2025) Enhancing grid resilience through DMR trunking communication systems. *World J Adv Eng Technol Sci* 17(3):197–212. <https://doi.org/10.30574/wjaets.2025.17.3.1551>
85. Ali A (2026) The impact of packaging design on recycling efficiency in municipal systems. *World J Adv Eng Technol Sci* 18(01):122–139. <https://doi.org/10.30574/wjaets.2026.18.1.0019>
86. Zaman SU, Afrin S, Zaidi SKA, Islam KSA (2026) Resilient edge computing framework for autonomous, secure, and energy-aware systems. *World J Adv Eng Technol Sci* 18(01):105–121. <https://doi.org/10.30574/wjaets.2026.18.1.1577>
87. Mim MA, Sharif MM, Rahman F, Nahar S (2026) Smart IoT infrastructure for workplace efficiency and energy savings. *World J Adv Eng Technol Sci* 18(01):140–156. <https://doi.org/10.30574/wjaets.2026.18.1.0026>
88. Fazle AB, Taimun MTY, Fareed SM, Alam MS (2026) Ergonomic and automation based process redesign in industrial workstations. *Global J Eng Technol Adv* 26(01):091–108. <https://doi.org/10.30574/gjeta.2026.26.1.0010>
89. Dukkupati SSNC (2026), May 7 Design and performance evaluation of event-driven microservices for large-scale enterprise data processing. *SSRN*. <https://doi.org/10.2139/ssrn.6730598>
90. Shaik MH (2026), March 31 AI-driven site reliability engineering framework for proactive failure prediction and resilience optimization in large-scale cloud infrastructure. *SSRN*. <https://ssrn.com/abstract=6535379>
91. Tariq F, Shaik MH, Mirza SB, Islam MA (2026) Service failure detection in distributed microservice platforms. *Saudi J Eng Technol (SJEAT)* 11(5):501–510. <https://doi.org/10.36348/sjet.2026.v11i05.013>
92. Junaid A, Islam MS, Rahat MK, Asraf MS (2026) Radio network performance evaluation in multi-band LTE and 5G deployments. *Scholars J Eng Technol* 14(5):273–280. <https://doi.org/10.36347/sjet.2026.v14i05.008>
93. Khalid MA, Islam MA, Uddin MN, Umar M (2026) Structural design evaluation for steel industrial facilities under wind and seismic loads. *Saudi J Eng Technol (SJEAT)* 11(5):492–500. <https://doi.org/10.36348/sjet.2026.v11i05.012>
94. Sharmin M (2026) Information system–based project coordination and process improvement in service organizations. *Zenodo*. <https://doi.org/10.5281/zenodo.20382574>
95. Islam MA (2026), May 26 Risk based project governance models for large scale transportation and energy projects in the United States. *Research Square*. <https://doi.org/10.21203/rs.3.rs-9825852/v1>
96. Afroje S (2026), June 6 Information system integration for credit card operations and banking service efficiency [Preprint]. *Zenodo*. <https://doi.org/10.5281/zenodo.20508756>

97. Siddiki A (2026), May 9 Intelligent event-driven middleware architecture for scalable financial service integration using Kafka and distributed microservices. SSRN. <https://doi.org/10.2139/ssrn.6737940>

Figures

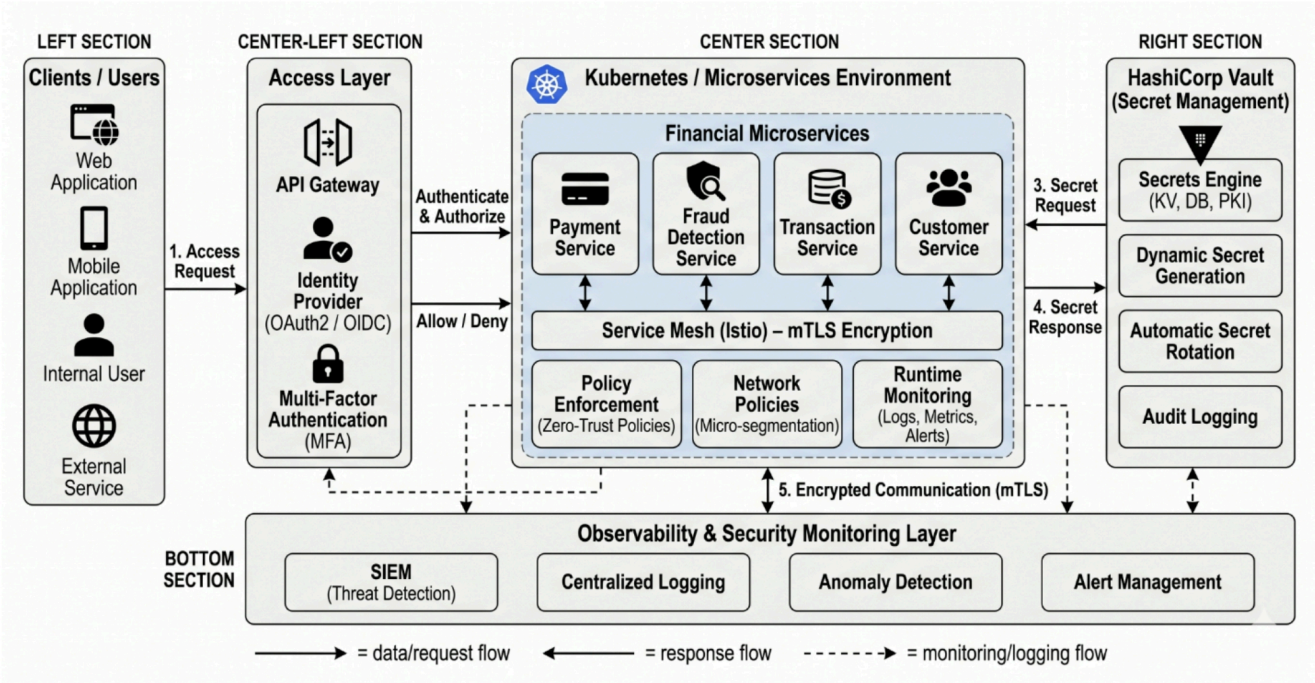


Figure 1

Zero-Trust Vault-Based Architecture for Distributed Financial Applications

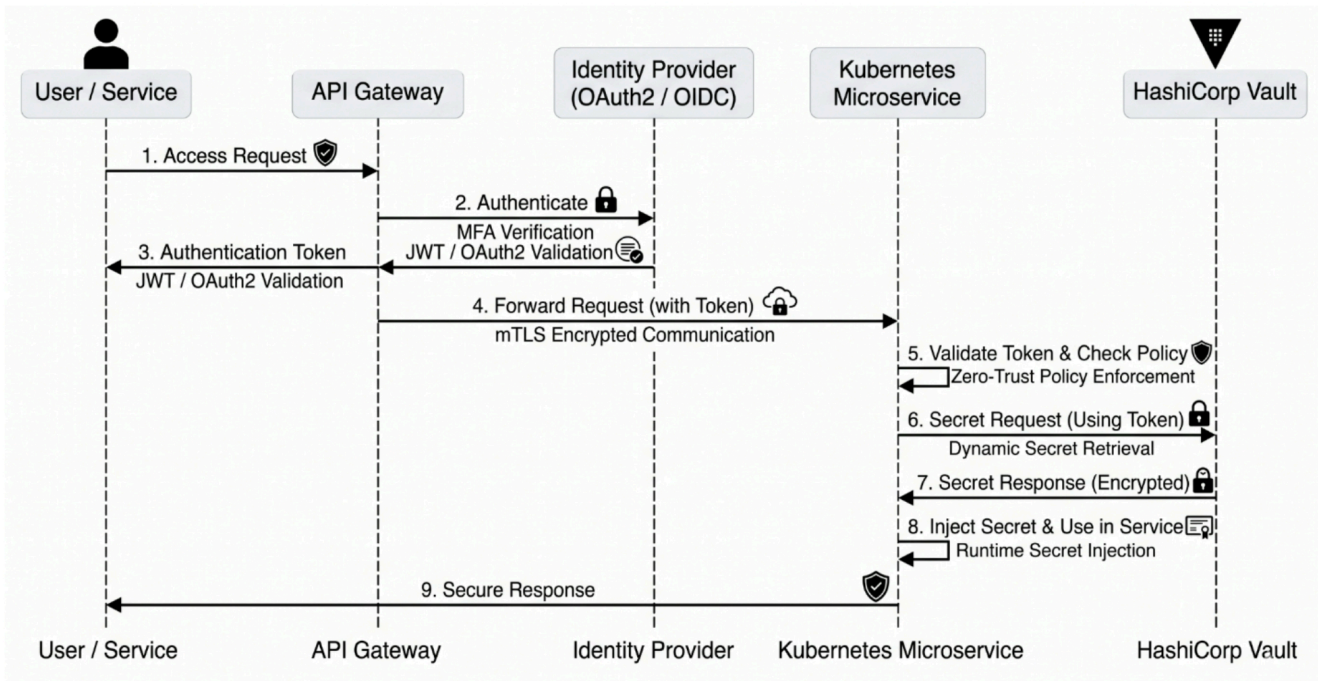


Figure 2

Authentication and Vault-Based Secret Access Workflow

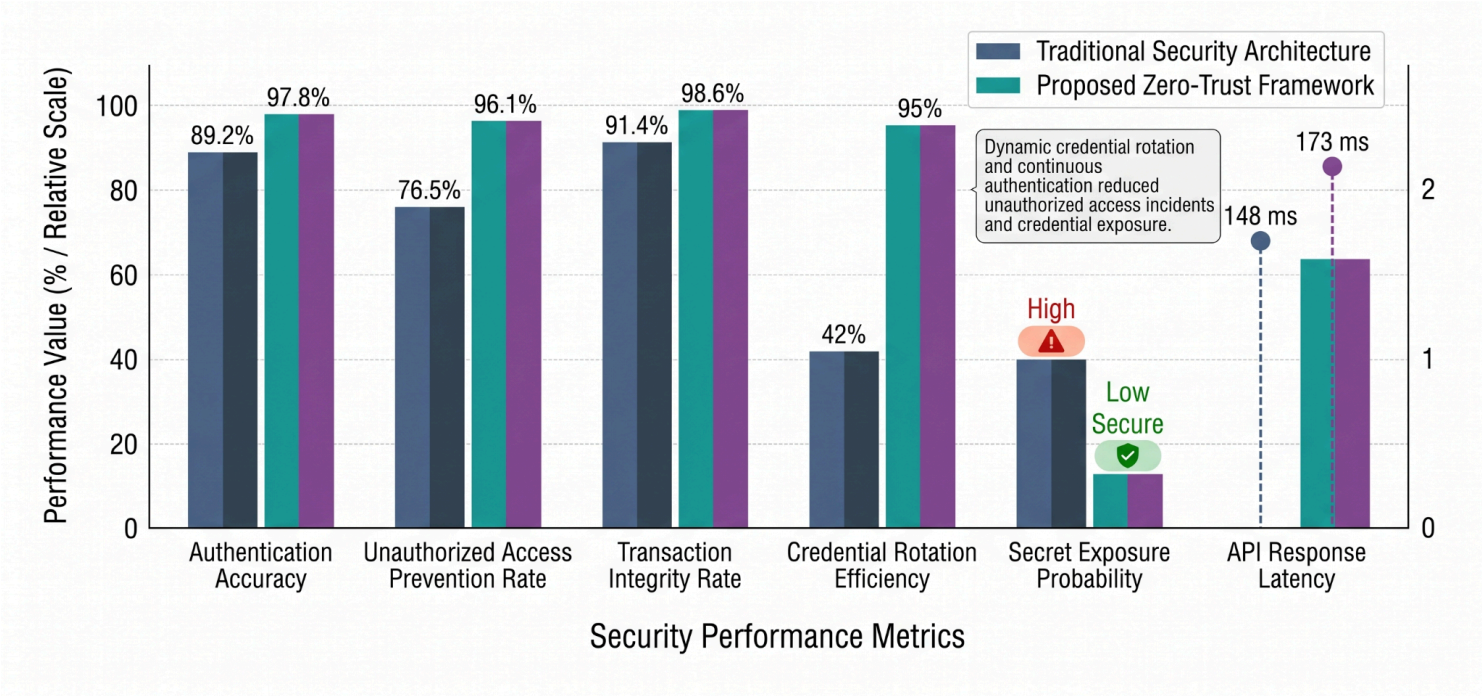


Figure 3

Performance Comparison Between Traditional Security Architecture and Proposed Vault-Based Zero-Trust Framework

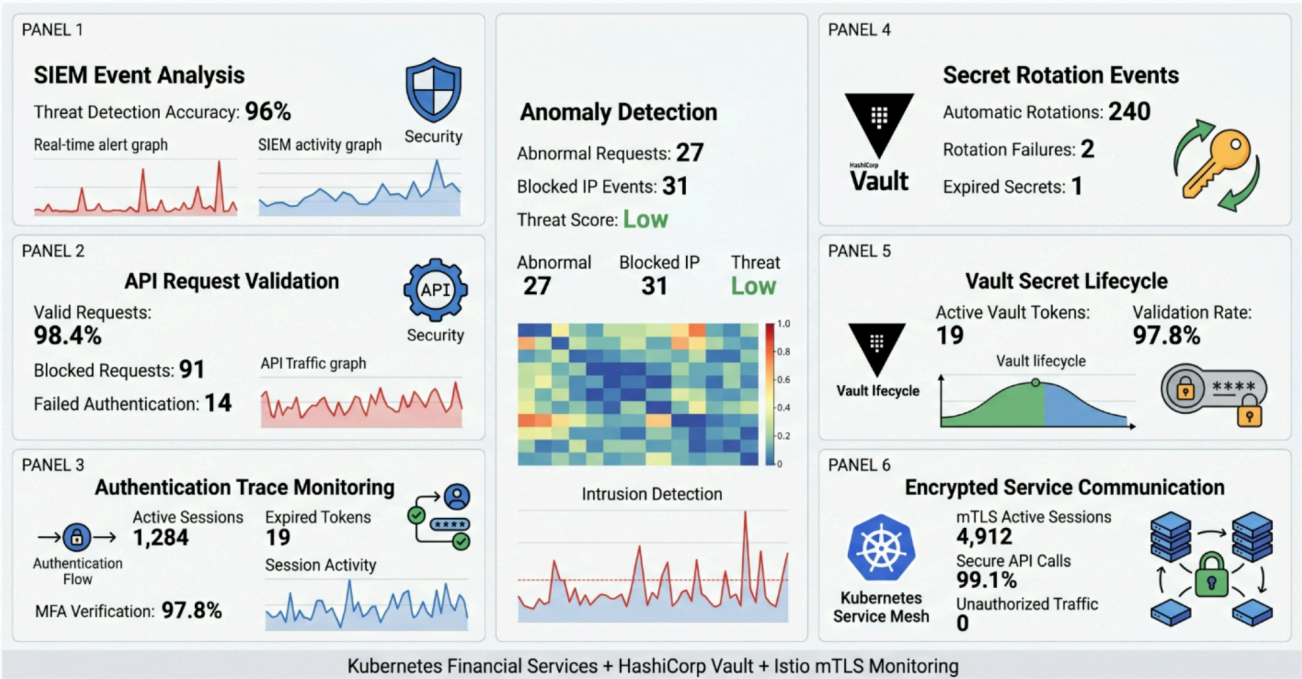


Figure 4

Runtime Security Monitoring and Threat Detection Dashboard