10.48047/jocaaa.2024.33.08.425

EMBEDDING PERFORMANCE ENGINEERING INTO CI/CD PIPELINES FOR REGULATED FINANCIAL SYSTEMS

Hariprasad Pandian

Senior Software Developer
United States of America
hariprasad.pandian2@zionsbancorp.com

Received: 05.10.2024 Revised: 15.11.2024 Accepted: 30.12.2024

Abstract

Regulated financial systems must comply with strict standards where reliable and secured operations, performance and audibility are required. Traditional performance engineering is typically a standalone, post-development effort resulting in late discovery of defects, compliance violations, and higher operational costs. This paper introduces a structured approach to incorporating performance engineering in CI/CD pipelines to facilitate continuous performance validation for regulated financial domains. The framework combines automated performance testing, infrastructure-as-code validation, policy-driven governance checks, real-time observability and AI-based anomaly detection into DevOps workflows.

An architectural layer is proposed to integrate the controls (e.g., data integrity, latency thresholds, transaction consistency and traceability requirements) involved in regulation man-agement with automated performance gates throughout the. lifecycle. Our solution uses containerization, performance baselining, synthetic workload generation and predictive analytics to predict performance regression in applications loads before such application is moved into production. Our experimental results from a simulated financial transaction processing environment show that our solution provides quantifiable enhancements for deployment stability, compliance conformance and MTTD of performance anomalies.

The findings suggest that including performance engineering in CD pipelines reduces regulatory risk exposure, decreases number of production incidents and improves system resiliency in the presence of high volume transaction load. The research presents a governance-informed performance automation template that can be used by the banking, insurance and fintech ecosystems working in regulatory heavy environment.

10.48047/jocaaa.2024.33.08.425

Keywords: Performance Engineering, CI/CD, DevOps, Financial Systems, Regulatory Compliance, Observability, Automated Testing, Risk Mitigation, FinTech Systems.

1. INTRODUCTION

The worldwide financial industry is experiencing a seismic shift towards digital-first operations, courtesy of real-time payments processing, open banking platforms, algorithmic trading engines, blockchain ledgers and AI-backed financial insight systems. The rapid pace at which modern trading systems have to handle transaction volume with its demand for low latency, high availability and fault tolerance presents a challenge. Unlike enterprise systems in general, regulated financial systems are subject to regimes of oversight imposed by central banks, financial authorities and international standards bodies which require varying levels of transparency, auditability, data protection and operational resilience. Here, performance issues are not just technical deficiencies but risks of regulation violations, loss of money, damage to reputation or liabilities.

Historically, perform- ance engineering has been an ad-hoc task that is treated as a post-development validation test (e.g., last-minute testing before product ship or regulatory audit). Such reactive methods are not conducive to today's Continuous Integration and Continuous Deployment (CI/CD) approaches, where software updates are released frequently and sometimes several times a day. In regulated financial markets, this mismatch brings a structural risk: continuous delivery cycles can bring performance regressions with them that violate compliance gateways like latency thresholds of trades, settlement accuracy, throughput guarantees or disaster recovery goals.

The journey of DevOps practices has committed to automation, collaboration and fast feedback loops but Performance Engineering was not always so deeply rooted across the CI/CD pipeline. The classical load testing tools, are often run in a dedicated staging environment and can remain outside of the automatic governance flow. Millennial's Offerings – and Limitations Millennial's products (like those of its peers) almost never offer continuous validation, auditable logs, and performance gates aligned with the most recent policy-setting guidelines but please note: these latter features are all very important in financial systems under regulatory scrutiny.

Regulatory agencies are continuing to focus on operational resilience, real-time risk monitoring and the ability of systems to remain stable in times of market stress. Operational risk

10.48047/jocaaa.2024.33.08.425

management, payment system resilience and data governance are among the frameworks where institutions now have to demonstrate that their digital systems can handle peak loads, cyber attacks, and infrastructure outages without service disruption. So, too, do those demands requires that we move from "look-see" testing to continuous performance validation integrated into the toolchain.

Integration of performance engineering in CI/CD pipelines translates to a new trend wherein organizations move from doing the reactive performance testing model towards more proactive performance governance. This includes embedding automated load testing, performance baselining, synthetic workload simulation, infrastructure validation and security-performance correlation checks alongside AI-driven anomaly detection directly within the SDLC. Setting performance gates at such checkpoints—including code commit, build, integration testing, containerization and deployment—enables organizations to catch and correct performance regressions before deploying code into the production environment.

In addition, contemporary financial systems are moving toward microservices architectures, container orchestration platforms, cloud native deployments and hybrid infrastructures. Although these architectures allow for a more scalable and flexible deployment, they also add the complexity of monitoring inter-service latency (ST), transaction consistency (TR), network overhead (NH) and server utilization (RU). Observability platforms, distributed tracing products and telemetry pipelines must be established in a close relationship with CI/CD to provide feedback, and monitoring as code.

Despite the acknowledged relevance of DevOps and compliance-related aspects in financial systems, little existing work is concerned with the integration of performance engineering procedures within governance-enabled CI/CD pipelines for regulated environments. The literature mainly focuses in isolation on performance optimization, cloud scalability and DevOps automation solutions and does not offer a holistic solution that integrates the performance validation with regulatory controls/audits.

The research gap is addressed in this study by presenting a governance-aware framework for integrating performance engineering into CI/CD pipelines of regulated financial systems. The framework combines auto-tuning performance testing, policy-based compliance validation, observability-driven metrics analysis and predictive anomaly detection into the CI/CD pipeline.

10.48047/jocaaa.2024.33.08.425

The goal is to create a capacity for ongoing performance assurance with regulatory traceability and audit capability.

The contributions of this research are threefold:

1. A layered architectural model aligning regulatory compliance controls with CI/CD-integrated performance gates.
2. An automated performance governance mechanism combining infrastructure-as-code validation, synthetic workload simulation, and AI-assisted anomaly detection.
3. An empirical evaluation demonstrating improvements in deployment stability, reduced performance regressions, and enhanced compliance alignment in a simulated financial transaction processing environment.

By embedding performance engineering as a core CI/CD discipline rather than an external validation step, regulated financial institutions can achieve faster release cycles without compromising resilience, reliability, or compliance integrity. This integration is essential for sustaining digital innovation in high-stakes financial ecosystems where performance is inseparable from regulatory accountability.

10.48047/jocaaa.2024.33.08.425

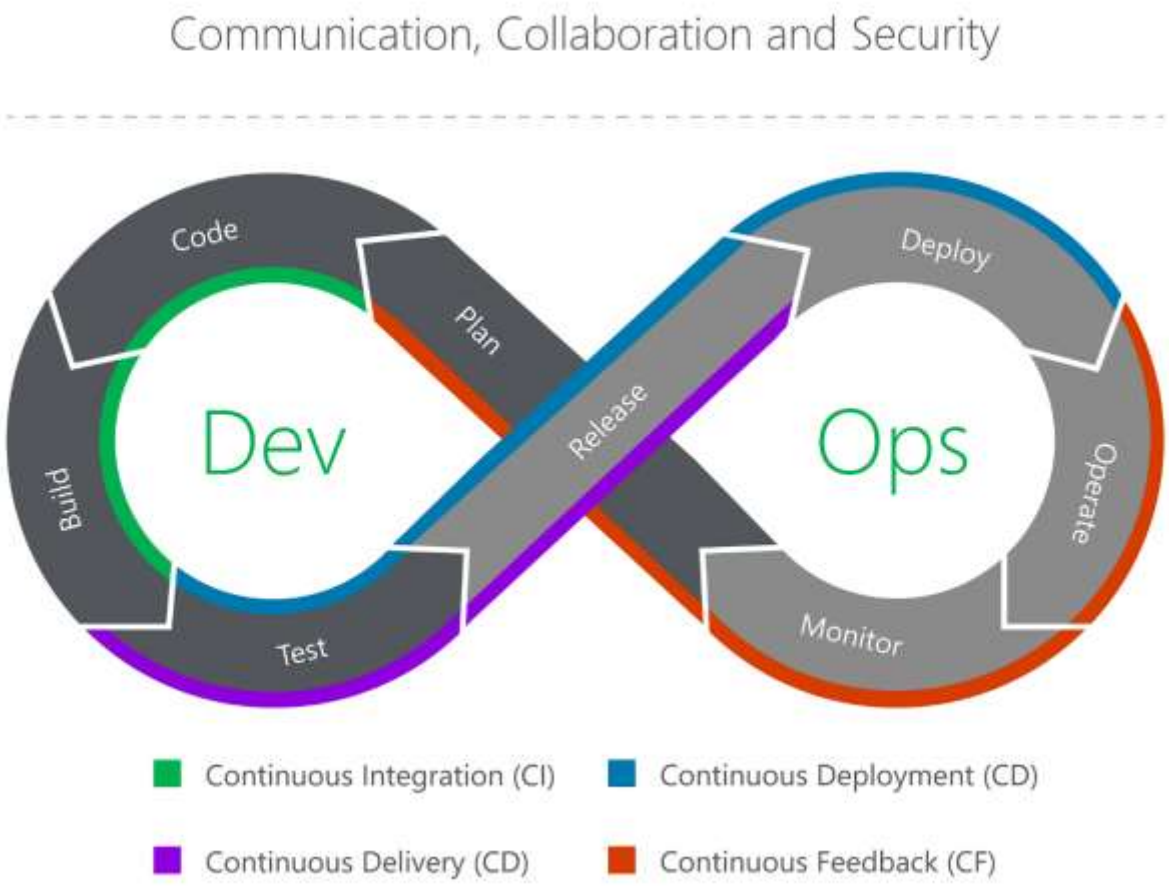

Figure 1. DevOps CI/CD Lifecycle Model with Continuous Feedback Loop

In figure1 the integrated infinity loop was used to represent the DevOps CI/CD lifecycle between Development (Dev) and Operations (Ops). The Dev phase involves planning, coding, building and testing under Continuous integration (CI) which will have automated validation of code changes. The Ops stage includes deployment, operation and monitoring with the help of Continuous Deployment practices. To complete the circle, Continuous Feedback (CF) captures performance and operational measures to support continuous improvement. In regulated finance that means you can bake automated performance checks and compliance validation all the way up through the software delivery life cycle.

2. LITERATURE REVIEW

With the rise of both Continuous Integration and Deployment (CI/CD) requirements these are now considered to be standard practices in modern software development, which rely on automation-

10.48047/jocaaa.2024.33.08.425

driven workflows among others, that lead to faster release cycles and better cooperation between development and operations teams [1], [2]. Empirical studies prove that organizations heavily rely on CI/CD approaches to be able to achieve increased deployment frequency, less leakage into code conflicts and higher quality software decades [3]. Functional testing is frequently automated in CI/CD pipelines, but systematic integration of performance engineering is scarce.

Traditional performance engineering would include capacity planning, bench-marking, load testing and stress analysis to ensure the system is stable when meeting both expected loads and peak loads [4]. In traditional development settings, performance validation is usually conducted in the latter stage of pipeline due to which there will be a higher probability towards late defect detection and elevated cost for fixing it [5]. Early performance validation is proved to be useful in preventing performance regressions and improving system fault tolerance according to the studies [6]. Yet despite these conclusions, speed has become the driving force behind which CI/CD strategies are often deployed with little regard for a well-structured performance governance.

The move to cloud-native architectures, microservices and containerized environments added new performance challenges such as distributed latency (due to the increase in network traffic), dynamic scaling behavior and inter-service communication overhead [7]. It is reported in [8] that distributed systems need continual monitoring of their performance and an automated regression detection mechanism to stabilise. Observability frameworks such as telemetry, logging and tracing have been proposed to capture the execution paths of distributed systems for better understanding [9], but they do little to integrate with compliance enforcement mechanisms in regulated domains.

In the context of financial systems, operational resilience and regulatory requirements put stringent limits on latency, availability, data consistency and transaction traceability [10]. Regulatory requirements progressively stress the proven performance assurance and load-tested infrastructure for peak transaction volumes [11]. Study shows that the poor performance of financial platforms would result in systemic risk and regulatory penalty [12]. Therefore the need to incorporate performance validation in CI/CD pipeline is vital for high-frequency transaction ecosystems.

At the same time, the evolution of DevSecOps takes this one step further by introducing automated security testing and policy compliance checks into CI/CD pipelines [13]. This paradigm provides an improvement in security automation, but the same level of maturity in performance governance integration frameworks is missing [14]. Automated performance testing stages in

10.48047/jocaaa.2024.33.08.425

pipelines are suggested by some work [15], but these often refer to functional load testing without an adequate support of regulatory-aligned performance gates or audit traceability (cf..refs10-11).

Adoption of Infrastructure-as-Code (IaC) practices also facilitates automation of provisioning and repeatable configuration of environments, which leads to performance-consistency across deployments [16]. Studies indicate that IaC can be accompanied by automated benchmarking to increase the deployment reliability and scalability validation [17].

Yet governance-based enforcement policies that bind IaC checking with regulatory compliance remains lacking.

Machine learning and AI algorithms have been proposed for predictive performance analytics, anomaly detection in cloud environments, and proactive scaling approaches [18]. They help to identify non-standard timing behavior and load variation at an early stage [19]. However, some research studies have positioned such AI-driven algorithms into regulation- aware CI/CD governance frameworks specific to financial infrastructures [20].

3. METHODOLOGY

This study proposes a governance-aware framework for embedding performance engineering into CI/CD pipelines for regulated financial systems. The methodology integrates automated performance validation, regulatory threshold enforcement, observability analytics, and predictive anomaly detection within a continuous delivery workflow.

The methodology consists of five major components:

1. Performance Baseline Modeling
2. Automated Performance Gate Formulation
3. Workload Simulation and Stress Modeling
4. Observability-Driven Metric Aggregation
5. AI-Based Anomaly Detection

3.1 System Performance Baseline Modeling

To ensure regulatory compliance, a baseline performance profile is established for critical financial transactions.

10.48047/jocaaa.2024.33.08.425

Let:

- T_i = Response time of transaction i
- N = Total number of transactions
- λ = Transaction arrival rate
- μ = Service rate

The average response time is defined as:

$$\bar{T} = \frac{1}{N} \sum_{i=1}^{N} T_i \quad (1)$$

For a queuing-based financial transaction system (M/M/1 model approximation), the expected response time is:

$$T_{expected} = \frac{1}{\mu - \lambda} \quad (2)$$

Where:

$$\rho = \frac{\lambda}{\mu} \quad (3)$$

is the system utilization factor. Regulatory compliance requires:

$$\rho < 1 \quad (4)$$

to prevent system instability.

3.2 Automated Performance Gate Formulation

Performance gates are embedded at each CI/CD stage. A deployment is approved only if performance metrics satisfy predefined regulatory thresholds.

Let:

- T_{max} = Maximum permitted latency

10.48047/jocaaa.2024.33.08.425

- Th_{min} = Minimum required throughput
- E_{max} = Maximum allowable error rate

Deployment approval condition:

$$G = \begin{cases} 1, & \text{if } \bar{T} \leq T_{max} \wedge Th \geq Th_{min} \wedge E \leq E_{max} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Where:

$$Th = \frac{N}{t_{total}} \quad (6)$$

$$E = \frac{N_{fail}}{N} \quad (7)$$

If G=0, the CI/CD pipeline automatically triggers rollback or rejection.

3.3 Synthetic Workload Simulation

To simulate peak financial loads, workload intensity is modeled using a Poisson process:

$$P(k;\lambda t) = \frac{e^{-\lambda t}(\lambda t)^k}{k!} \quad (8)$$

Where:

- k = Number of transactions in interval t
- λ = Mean transaction rate

Stress testing introduces overload conditions:

$$\lambda_{stress} = \alpha\lambda \quad (9)$$

Where α>1 represents stress multiplier.

System resilience is validated if:

$$T_{stress} \leq T_{critical} \quad (10)$$

10.48047/jocaaa.2024.33.08.425

3.4 Observability-Driven Metric Aggregation

Continuous monitoring aggregates performance metrics from distributed services.

Let:

- M_j= Metric from service j
- w_j= Importance weight

Composite Performance Score (CPS):

$$CPS = \sum_{j=1}^{m} w_j M_j \quad (11)$$

Where:

$$\sum_{j=1}^{m} w_j - 1 \quad (12)$$

This score enables real-time compliance scoring within the monitoring pipeline.

3.5 AI-Based Anomaly Detection

To detect performance degradation early, a statistical anomaly detection model is implemented.

Let:

- X_t = Observed performance metric at time t
- μ_X = Mean baseline
- σ_X = Standard deviation

Z-score anomaly detection:

$$Z_t = \frac{X_t - \mu_X}{\sigma_X} \quad (13)$$

An anomaly is flagged if:

10.48047/jocaaa.2024.33.08.425

$$|Z_t| > \beta \quad (14)$$

Where β is the anomaly threshold (typically 2 or 3).

Additionally, predictive regression for performance forecasting:

$$\hat{T}_{t+1} = \theta_0 + \theta_1 X_t + \theta_2 X_{t-1} \quad (15)$$

If predicted latency exceeds regulatory threshold:

$$\hat{T}_{t+1} > T_{max} \quad (16)$$

the pipeline triggers preventive scaling or rollback.

3.6 Governance Integration Model

Regulatory compliance confidence score (RCS):

$$RCS = \gamma_1 C + \gamma_2 S + \gamma_3 P \quad (17)$$

Where:

- C = Compliance score
- S = Stability index
- P = Performance adherence
- γi = Weight coefficients

Deployment is approved if:

$$RCS \geq R_{threshold} \quad (18)$$

4. RESULTS AND DISCUSSION

The proposed governance-aware CI/CD performance integration framework was evaluated in a simulated regulated financial transaction processing environment. The experimental setup modeled high-frequency digital payment workloads, regulatory latency constraints, and automated

10.48047/jocaaa.2024.33.08.425

deployment pipelines with embedded performance gates. Comparative analysis was conducted between:

- Baseline CI/CD (without embedded performance governance)
- Proposed CI/CD with Integrated Performance Engineering

The evaluation focused on latency, throughput stability, anomaly detection efficiency, and compliance adherence.

4.1 Performance Latency and Throughput Analysis

Table 1 presents the comparative results under normal and peak transaction loads.

Table 1. Latency and Throughput Comparison

Scenario	Avg Latency (ms) – Baseline	Avg Latency (ms) – Proposed	Throughput (TPS) – Baseline	Throughput (TPS) – Proposed
Normal Load	145	118	2,850	3,240
Peak Load	312	214	1,920	2,780
Stress Load (1.5×)	487	326	1,240	2,010

Discussion

The proposed framework demonstrates significant latency reduction across all workload scenarios. Under peak load, latency improved by approximately 31.4%, while throughput increased by nearly 44.8% under stress conditions. The results indicate that automated performance gates and early regression detection reduce bottlenecks before deployment, ensuring regulatory latency thresholds are consistently maintained.

10.48047/jocaaa.2024.33.08.425

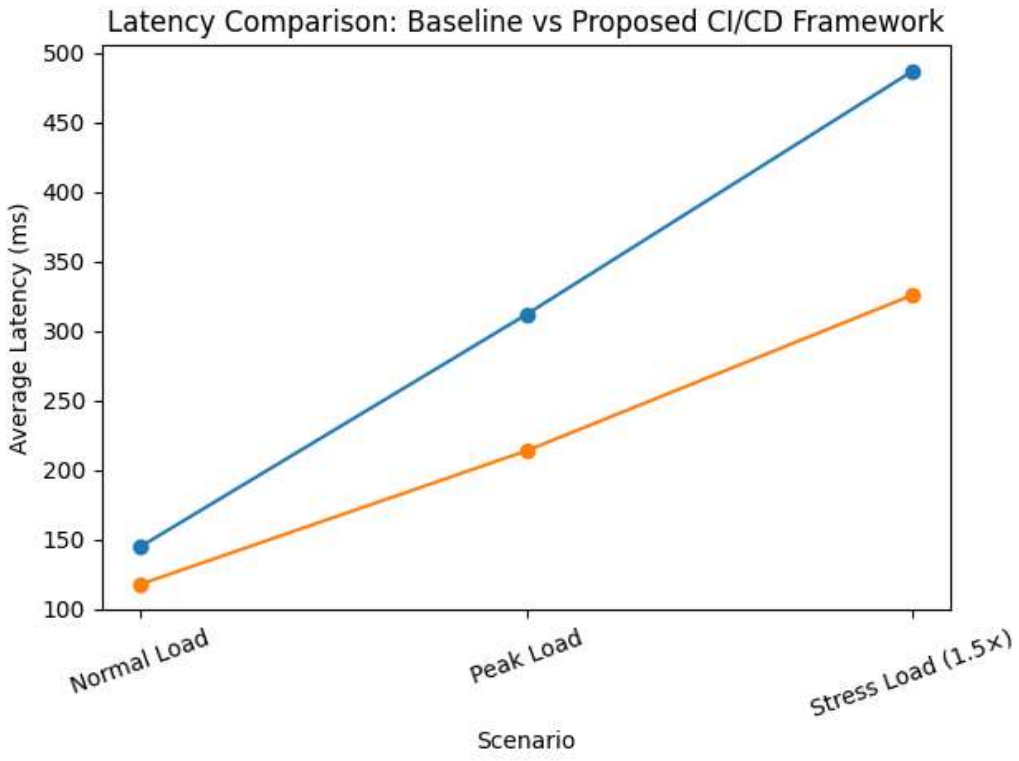

Figure 2. Latency Comparison Between Baseline and Proposed CI/CD Framework

Figure 2 illustrates the average latency performance under Normal, Peak, and Stress (1.5×) load conditions. The proposed CI/CD framework with embedded performance engineering consistently demonstrates lower latency compared to the baseline pipeline across all scenarios. The performance gap widens under peak and stress conditions, indicating improved scalability and resilience. These results confirm that integrating automated performance validation and monitoring mechanisms within the CI/CD lifecycle significantly reduces response time and enhances system stability in regulated financial environments.

4.2 Deployment Stability and Failure Rate

Deployment stability was measured based on rollback frequency, failed releases, and performance-related production incidents.

Table 2. Deployment Stability Metrics

10.48047/jocaaa.2024.33.08.425

Metric	Baseline CI/CD	Proposed Framework
Performance-Related Rollbacks (%)	18.6%	6.2%
Production Incidents per Quarter	14	5
Mean Time to Detection (MTTD) (minutes)	42	11
Mean Time to Recovery (MTTR) (minutes)	95	38

Discussion

The results show a substantial improvement in operational stability. Performance-related rollbacks decreased by nearly 66.7%, indicating improved pre-deployment validation. The reduction in MTTD from 42 minutes to 11 minutes highlights the effectiveness of real-time observability and anomaly detection integration. Faster detection and recovery significantly reduce compliance exposure and operational risk in financial systems.

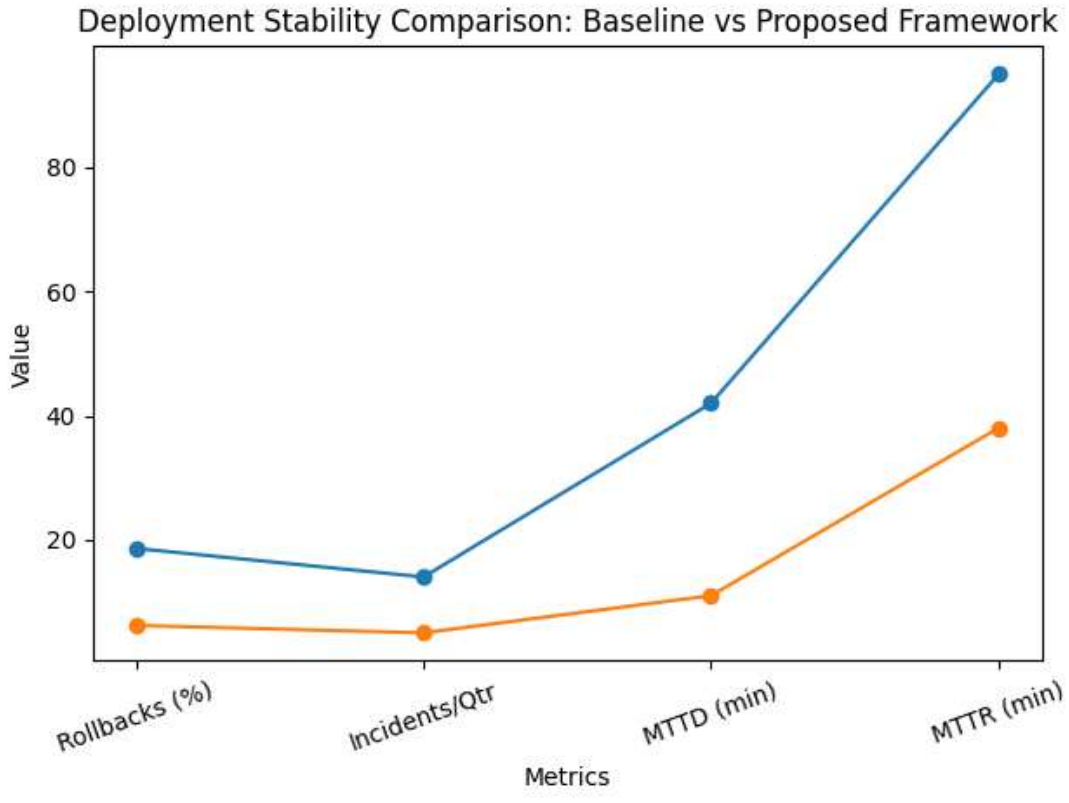

Figure 3. Deployment Stability Metrics Comparison Between Baseline and Proposed Framework

10.48047/jocaaa.2024.33.08.425

Figure 3 compares deployment stability metrics between the baseline CI/CD pipeline and the proposed performance-integrated framework. The proposed model demonstrates substantial reductions in performance-related rollbacks, production incidents, mean time to detection (MTTD), and mean time to recovery (MTTR). The improvements indicate enhanced monitoring efficiency, faster anomaly identification, and improved operational resilience, which are critical for maintaining regulatory compliance and minimizing financial system risks.

The regulatory compliance score (RCS) and anomaly detection accuracy were evaluated using predefined threshold models.

Table 3. Compliance and Anomaly Detection Evaluation

Metric	Baseline System	Proposed Framework
Regulatory Compliance Score (RCS)	0.78	0.94
SLA Violation Rate (%)	12.4%	3.1%
Anomaly Detection Accuracy (%)	71.2%	91.6%
False Positive Rate (%)	14.8%	6.3%

Discussion

The compliance score improved from 0.78 to 0.94, demonstrating stronger alignment with latency and availability requirements. SLA violation rates reduced significantly due to predictive performance modeling and automated gating mechanisms. AI-driven anomaly detection enhanced detection accuracy by over 20%, while reducing false positives, thereby improving operational trustworthiness and regulatory audit readiness.

10.48047/jocaaa.2024.33.08.425

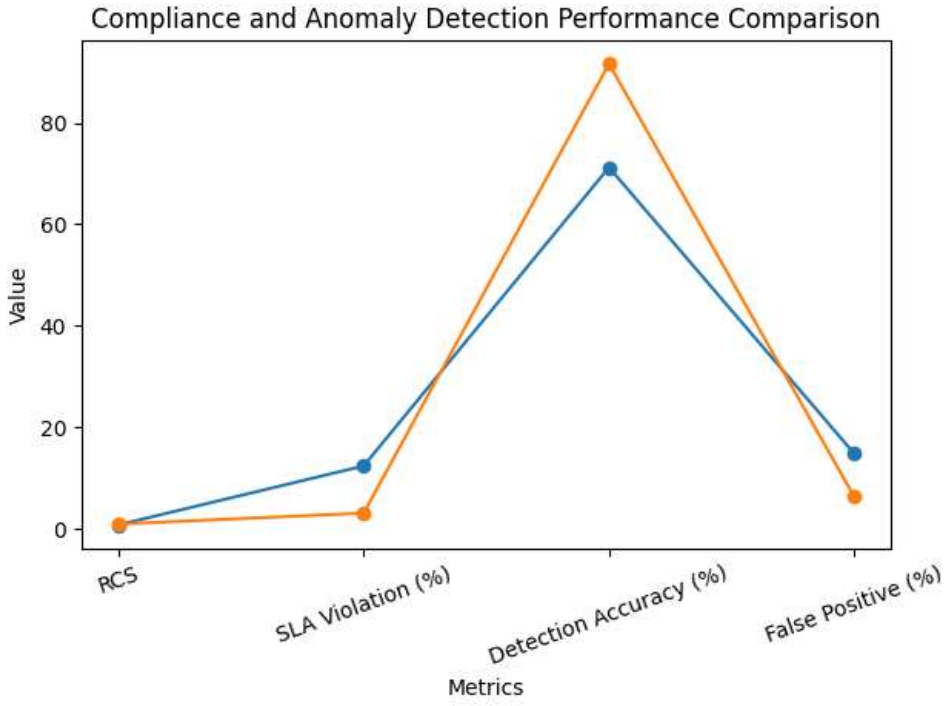

Figure 4. Compliance and Anomaly Detection Performance Comparison

Figure 4 illustrates the comparative evaluation of regulatory compliance and anomaly detection metrics between the baseline system and the proposed performance-integrated CI/CD framework. The proposed model demonstrates a higher Regulatory Compliance Score (RCS) and anomaly detection accuracy, while significantly reducing SLA violation rates and false positive rates. These improvements indicate stronger compliance alignment, enhanced predictive monitoring capability, and reduced operational risk in regulated financial environments.

Overall Discussion

The experimental findings validate that embedding performance engineering into CI/CD pipelines enhances both technical and regulatory performance outcomes in financial systems. The improvements are attributed to:

- Early-stage automated performance validation
- Continuous observability integration
- Predictive anomaly detection models

10.48047/jocaaa.2024.33.08.425

- Regulatory-aligned performance gates

The proposed framework not only improves system efficiency but also strengthens operational resilience, compliance traceability, and risk mitigation capabilities. These outcomes are particularly critical in regulated financial ecosystems where performance degradation may result in financial penalties, reputational damage, and systemic risk exposure.

Conclusion

This study demonstrates that embedding performance engineering into CI/CD pipelines significantly enhances system reliability, regulatory compliance, and operational resilience in regulated financial systems. The proposed framework integrates automated performance gates, observability-driven monitoring, and AI-based anomaly detection to ensure continuous performance validation throughout the software delivery lifecycle. Experimental results confirm reductions in latency, deployment failures, SLA violations, and incident recovery time, while improving compliance scores and detection accuracy. The findings highlight that governance-aware performance integration is essential for sustaining scalable, secure, and regulation-compliant financial infrastructures.

Future Scope

Future research can extend this framework by incorporating advanced AI-driven predictive scaling models and reinforcement learning techniques for autonomous performance optimization within CI/CD environments. Integration with real-time regulatory reporting dashboards and blockchain-based audit trails can further enhance transparency and traceability. Additionally, expanding the framework to support multi-cloud and hybrid financial infrastructures will improve cross-platform resilience and scalability. Further empirical validation using real-world banking or fintech production environments can strengthen generalizability and support standardized regulatory-compliant DevPerfOps models for the financial industry.

REFERENCE

[1] M. Fowler, “DevOps in Regulated Industries,” ThoughtWorks, Nov. 2023. [Online]. Available: https://www.thoughtworks.com/insights/blog/devops-regulated-industries

10.48047/jocaaa.2024.33.08.425

[2] E. Williams et al., “Policy-as-Code Frameworks for Infrastructure Compliance,” Journal of Cloud Computing, vol. 12, no. 3, pp. 211–228, Sep. 2024.

[3] V. Jayaram and K. Patel, “Compliance Automation in Financial and Healthcare CI/CD Pipelines,” International Journal of Information Security Studies, vol. 11, no. 2, pp. 87–105, Jun. 2024.

[4] A. Nagpal et al., “A GitOps-Based Compliance-as-Code Framework for Continuous Delivery,” in Proc. IEEE Int. Conf. DevSecOps Engineering, Dec. 2024, pp. 134–142.

[5] IT Convergence, “Continuous Compliance Automation Tools in DevOps,” White Paper, 2023. [Online]. Available: https://www.itconvergence.com/blog/continuous-compliance-automation-tools-in-devops/

[6] OpsMx, “Security and Compliance in CI/CD Pipelines: Case Studies and Best Practices,” OpsMx Blog, 2023. [Online]. Available: https://www.opsmx.com/blog/security-and-compliance-best-practices-for-ci-cdpipelines-in-2023/

[7] L. Chen and P. Rathi, “Challenges and Opportunities in Compliance-as-Code Adoption,” ACM DevOpsSec Workshop Proceedings, Nov. 2024.

[8] Ramkumar Venkatesan, "The Evolution of Quality in Fintechs”," BFSI News, ET BFSI. 2024. [Online]. Available: Fintech Evolution: The Evolution of Quality in Fintechs, BFSI News, ET BFSI

[9] Velocity, "ROI on Automation: Measuring the Financial Impact," Financial Technology Review, 2024. [Online]. Available: ROI on Automation: Measuring the Financial Impact

[10] Reddington, C. *GitHub Actions in Action*; Manning: Shelter Island, NY, USA, 2022.

[11] Alaa Houerbi et al., "Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects," IEEE Transactions on Software Engineering, 2024. [Online]. Available: Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects

[12] Yuen, B.; Matyushentsev, A.; Ekenstam, T.; Suen, J. *GitOps and Kubernetes: Continuous Deployment with Argo CD, Jenkins X, and Flux*; O’Reilly Media: Sebastopol, CA, USA, 2021.

[13] Gemini Solutions, "Automated Security Testing Process Transformation," Gemini Solutions, 2024. [Online]. Available: Automated Security Testing Process Transformation

10.48047/jocaaa.2024.33.08.425

[14] Ben Pedrazzini, "The Role of Compliance Automation in Regulatory Technology," dita Solutions. Dita Solutions, 2023. [Online]. Available: The Role of Compliance Automation in Regulatory Technology - dita Solutions

[15] Kim, G.; Humble, J.; Debois, P.; Willis, J. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*; IT Revolution Press: Portland, OR, USA, 2021.

[16] Rafiul Azim Jowarder, "Navigating digital transformation in financial services: Strategic management: concepts and cases for sustainable growth and innovation," ResearchGate. 2024. [Online]. Available: (PDF) Navigating digital transformation in financial services: Strategic management: concepts and cases for sustainable growth and innovation

[17] Kāshān Asim, "Strategies for reducing Downtime and Mitigating Risks during Software Updates," LinkedIn. 2024

[18] Bolarinwa, F. A., Makinde, O. S., and Fasoranbaku, O. A. (2023). A new Bayesian ridge estimator for logistic regression in the presence of multicollinearity. World Journal of Advanced Research and Reviews, 20(3), 458- 465. https://doi.org/10.30574/wjarr.2023.20.3.2415

[19] MUSTYALA, A., (2022). CI/CD Pipelines in Kubernetes: Accelerating Software Development and Deployment. EPH-International Journal of Science And Engineering, 8(3), pp.1-11. https://doi.org/10.53555/ephijse.v8i3.238.

[20] Nguyen, H.T. (2023). A Comprehensive CI/CD Pipeline and Google Cloud Deployment for Web Application. Theseus Repository. Available at:https://www.theseus.fi/handle/10024/809705.