Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Optimizing DevOps and MLOps for Financial Institutions: Architecture and Compliance

Mohammed Ahnouch[1,2*], Lotfi Elaachak[1], Abderrahim Ghadi[1]
[1]*DIS Team, C3S Laboratory, FSTT, University Abdelmalek Essaâdi*
[2]*PRISM, UFR Gestion, Université Paris 1 Panthéon Sorbonne*

ARTICLE INFO

Received: 29 Dec 2024

Revised: 15 Feb 2025

Accepted: 24 Feb 2025

ABSTRACT

Financial institutions operate under dual pressures: the need for rapid innovation driven by competition and evolving customer expectations, contrasted with the necessity of adhering to stringent regulatory frameworks like Basel III, CRD V, and SR 11-7. Modern methodologies like DevOps and MLOps promise agility and efficiency but face significant adoption challenges within this regulated context. This paper addresses this critical intersection by consolidating current research on IT architecture, DevOps, and MLOps specifically for the banking sector. We focus on practices supporting robust data aggregation, risk management, and compliance reporting, while acknowledging persistent challenges such as legacy system integration and rigorous model governance. Recognizing a gap between general principles and practical implementation guidance, we propose two concise, research-grounded architectural blueprints. These blueprints offer actionable models for designing integrated DevOps/MLOps workflows that ensure continuous compliance and operational resilience, providing valuable insights for practitioners and researchers navigating the complex interplay of agile development and financial regulation.

Keywords: DevOps, MLOps, Financial Regulation, Banking Architecture, Compliance, Basel III, CRD V, SR 11-7, Risk Management, Model Governance, Architectural Blueprint, RegTech, DevSecOps

INTRODUCTION

The global financial services industry exists in a state of continuous flux, driven by intense market competition, shifting customer demands for digital services, and an ever-more complex web of regulations (48; 2). International accords like Basel III (9), regional directives such as CRD V (16), and national guidance on critical areas like model risk management (e.g., the US Federal Reserve's SR 11-7 (11)) impose strict operational and reporting requirements. Consequently, financial institutions must constantly evolve their Information Technology (IT) architectures and operational processes to simultaneously achieve agility, maintain resilience, and ensure unwavering regulatory compliance (18). This balancing act represents a central challenge for the sector.

Modern software engineering and operational paradigms, notably DevOps (26; 21) and Machine Learning Operations (MLOps) (46; 13), offer significant potential benefits. DevOps practices aim to break down silos between development and operations, automating delivery pipelines to increase speed and reliability. MLOps extends these principles to the unique lifecycle of machine learning models, addressing challenges like reproducibility, monitoring, and governance crucial for financial

applications from fraud detection to algorithmic trading. The state-of-the-art involves highly automated CI/CD pipelines, infrastructure managed as code, and increasingly sophisticated model management platforms.

However, the adoption of these modern practices within the highly regulated financial context is far from straightforward (15; 2). The core tenets of DevOps and MLOps-speed, iteration, and continuous change-must be carefully reconciled with non-negotiable regulatory demands for security, auditability, data integrity, robust governance, and transparent reporting. Furthermore, many institutions grapple with significant legacy systems, which often represent substantial technical debt and hinder modernization efforts (31). While research explores

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DevOps (43) and MLOps (6) adoption challenges, and the potential of RegTech (8), a gap often exists between high-level principles and concrete architectural guidance tailored for financial compliance.

This paper aims to bridge this gap by providing actionable architectural blueprints. We synthesize current academic knowledge and industry best practices concerning IT architecture, DevOps, MLOps, and regulatory compliance within finance. Our contribution lies in presenting two distinct, yet principled, reference models (Section 3) designed to address common scenarios: modernizing domestic institutions with legacy cores, and managing complex international operations under multiple regulatory regimes. These blueprints provide concrete structures for integrating DevOps and MLOps workflows in a manner that fosters continuous compliance alongside operational excellence, offering practical value to practitioners and a structured basis for further academic inquiry. The subsequent sections review relevant background literature (Section 2), detail the proposed blueprints (Section 3), discuss their implications (Section 4), and offer concluding remarks (Section 5).

BACKGROUND

Understanding the proposed blueprints requires familiarity with current trends and challenges across several interconnected domains: IT architecture evolution, DevOps and MLOps practices, and the impact of financial regulation.

Architectural Evolution in Finance Financial IT landscapes have transitioned from traditionally stable but inflexible monolithic core banking systems (14). These legacy systems often impede innovation due to accumulated technical debt and data silos (31). Service-Oriented Architecture (SOA) was an early attempt at modularity (41), but the current state-of-the-art leans towards microservice architectures (36). These offer enhanced agility, scalability, and independent deployment (7; 39), often facilitated by cloud computing platforms (4; 33). However, cloud adoption in finance necessitates careful consideration of security, data residency, and regulatory oversight (17; 47). Effective API strategies remain crucial for integrating modern applications with persistent legacy components (49).

DevOps and DevSecOps in Regulated Contexts DevOps principles emphasize automation, measurement, collaboration, and rapid feedback loops through practices like Continuous Integration and Continuous Delivery (CI/CD) (26;24; 21).

While proven effective in general software engineer-ing, applying DevOps in finance requires significant adaptation (15). The state-of-the-art involves DevSecOps, which integrates security and compliance checks early ("shifting left") within the automated pipeline (35). Advanced practices include automated security scanning (SAST, DAST, SCA), managing infrastructure reliably using Infrastructure as Code (IaC) (34), and embedding automated compliance validation and evidence gathering throughout the delivery process (43;25). The challenge lies in implementing these controls effectively without unduly hindering development velocity.

MLOps for Governed Machine Learning As machine learning applications become pervasive in finance, MLOps has emerged to manage their unique lifecycle (46; 13; 1). MLOps addresses critical aspects like experiment tracking, model versioning, automated retraining, monitoring for performance degradation or data/concept drift, and ensuring reproducibility (45). In finance, MLOps is inextricably linked with Model Risk Management (MRM) regulations, such as SR 11-7 (11). State-of-theart MLOps in finance therefore incorporates robust governance workflows, mandatory independent validation stages (12), techniques for model explainability (XAI), and rigorous fairness and bias assessments (6; 44; 22). Adapting general MLOps tools and platforms to meet these specific financial governance requirements remains an active area of development (40). Techniques like Federated Learning represent an advanced approach to handle data privacy and residency constraints across jurisdictions (32; 30).

Regulatory Pressures and Technological Responses Financial regulations, particularly standards like BCBS 239 concerning risk data aggregation and reporting (10), heavily influence IT architecture design. They necessitate robust data governance frameworks (38), reliable data lineage, and often employ technologies like event streaming platforms (e.g., Kafka (27)) for timely data processing. The field of Regulatory Technology (RegTech) specifically focuses on using technology to streamline and automate compliance tasks (5; 8). Advanced RegTech approaches include automating regulatory reporting, transaction monitoring, and leveraging Policy-as-Code (PaC) frameworks (e.g., OPA (Open Policy Agent)) to codify and automatically enforce compliance rules within IT systems (20). Despite

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

these technological advancements, effectively integrating modern practices with legacy systems and navigating the complexities of regulatory interpretation continue to pose significant challenges (31; 19).

METHODS

Based on the state-of-the-art and identified challenges, we propose two reference blueprints tailored for mid-sized financial institutions. These models offer structured approaches to implementing compliant DevOps and MLOps.

3.1 Blueprint 1: Hybrid Modernization (Domestic Focus)

3.1.1 Scenario & Challenges

This blueprint targets a domestic institution grappling with a significant monolithic legacy core system. The primary goals are to enhance agility and customer experience through modernization while managing the risks associated with the legacy core and ensuring compliance with domestic regulations (e.g., BCBS 239 for data, SR 11-7 for models). Key challenges include integrating with the restrictive legacy core, overcoming data fragmentation for reporting, accelerating traditionally slow deployment cycles, and establishing robust, governed ML capabilities.

3.1.2 Architectural Framework (Fig. 1)

A layered hybrid architecture is proposed to mitigate the risks of a full legacy replacement. This involves several distinct layers working in concert. The foundational Legacy Core System is interfaced using non-intrusive methods like Change Data Capture (CDC) to extract data events with minimal impact. A crucial API Integration Layer, often implemented using an Enterprise Service Bus (ESB) or a modern API Gateway platform, provides secure and managed access to both legacy functions and new services, handling necessary transformations and security enforcement (49). New business functionalities are developed within a Microservices Layer, utilizing containerization (e.g., Docker) and orchestration (e.g., Kubernetes (Kubernetes Project)) for independent deployment and scalability, guided by Domain-Driven Design principles (36; 7). To meet data aggregation and reporting requirements like BCBS 239, a Unified Data Platform Layer is essential, typically built around a data lake or data warehouse architecture, fed by event streaming platforms (e.g., Apache Kafka (Apache Kafka Project)), and incorporating strong metadata management, lineage tracking, and data quality tooling (38). Finally, the DevOps/MLOps Orchestration Layer provides the automation backbone, encompassing CI/CD tools, version control, IaC for provisioning (34), security scanning tools (43), ML experiment tracking and model management platforms (e.g., MLflow (Kubeflow Project)), and centralized logging and monitoring solutions (e.g., Prometheus (Prometheus Monitoring), Grafana (Grafana Labs)).

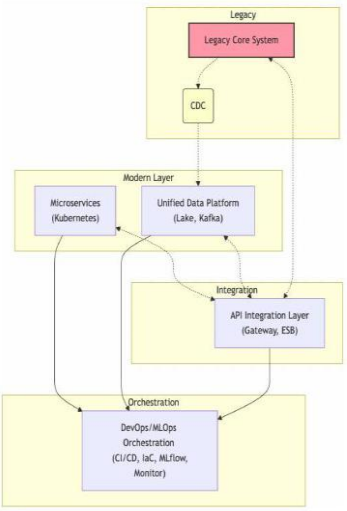

Figure 1: Proposed Hybrid Architecture Framework (Blueprint 1). This layered architecture integrates legacy core systems with modern microservices and unified data platforms, providing a controlled pathway for regulatory-compliant modernization. Each layer serves a specific function with API integration enabling connectivity between legacy and modern components.

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3.1.3 Regulated DevOps Workflow (Fig. 2)

This workflow embeds compliance checks and governance gates directly within the automated CI/CD pipeline, following DevSecOps principles (25; 35). Code committed to version control triggers an automated build process. Subsequently, comprehensive testing occurs, including unit, integration, and contract tests, alongside automated security scans (SAST, DAST, SCA) and policy checks using Policy-as-Code (PaC) frameworks (20). An automated risk gate evaluates the change; low-risk changes proceed, while high-risk changes trigger a mandatory manual review, often managed via a GRC platform. Approved changes are deployed to a staging environment using IaC. Further validation, including security penetration testing and compliance scenario tests, is performed. A final compliance gate verifies that all necessary evidence has been collected and approvals obtained before deployment to production, which should ideally use progressive techniques like canary releases, supported by robust monitoring and automated rollback capabilities. Continuous monitoring and audit logging provide feedback and support regulatory reporting.

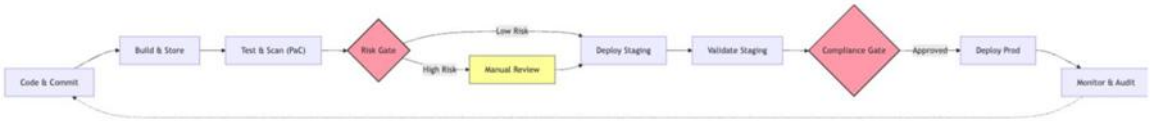

Figure 2: Proposed Regulated DevOps Workflow (Blueprint 1). This workflow incorporates regulatory risk gates and compliance validation before deploying to production environments. The dual-gate approach ensures both technical risk assessment and regulatory compliance are addressed through structured approval processes.

3.1.4 Compliant MLOps Framework (Fig. 3)

This framework systematically integrates Model Risk Management (MRM) governance, aligning with regulations like SR 11-7, across the entire ML model lifecycle (6;12). The process begins with formal Requirements Definition and initial Risk Assessment involving the MRM function. Data Acquisition and Preparation phases emphasize data quality, lineage tracking, bias assessment, and privacy compliance. Feature Engineering requires justification and versioning. During Model Development, emphasis is placed on interpretability for high-risk models, comprehensive experiment tracking (e.g., using MLflow), and applying explainability techniques (XAI) (44; 22). Model Evaluation assesses not just accuracy but also fairness, robustness, and sensitivity. A critical step is Independent Validation by a functionally separate team, scrutinizing the model's conceptual soundness, data, implementation, and documentation against regulatory standards. Deployment is controlled, followed by Continuous Monitoring for performance degradation, data/concept drift, and fairness metrics (45). A Governed Retraining process defines triggers and required re-validation efforts based on change significance, ensuring ongoing compliance under MRM oversight.

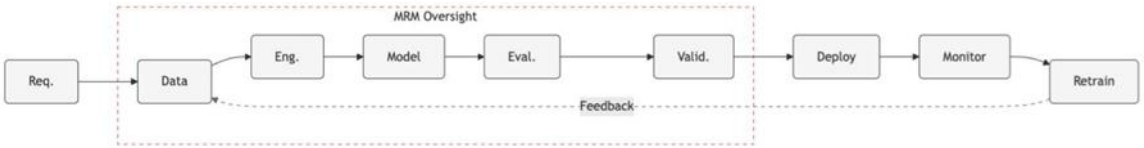

Figure 3: Proposed Compliant MLOps Framework (Blueprint 1). This framework implements SR 11-7 model risk management governance throughout the ML lifecycle. Key stages include independent validation, model evaluation, and continuous monitoring for concept drift, all under continuous oversight from the model risk management function.

3.2 Blueprint 2: Federated Architecture (International Ops)

3.2.1 Scenario & Challenges

This blueprint addresses the complexities faced by institutions operating across multiple countries or regulatory regions. Such institutions need to coordinate global development and operations while strictly adhering to diverse, sometimes conflicting, local regulations, particularly concerning data residency (e.g., GDPR in Europe) and cross-

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

border data transfer limitations. Key challenges include simultaneously managing compliance across multiple jurisdictions, navigating data sovereignty restrictions, standardizing technology stacks and practices globally where appropriate, overcoming time zone and cultural collaboration hurdles, and managing complex global identity and access controls.

3.2.2 Architectural Framework (Fig. 4)

A cloud-native, federated architecture is proposed to balance the need for global consistency with mandatory regional autonomy and compliance (17). This model typically features a Global Shared Services Layer providing foundational capabilities accessible across regions, such as standardized platform services (e.g., federated Kubernetes, service mesh templates), global CI/CD orchestration frameworks, centralized security services (IAM, secrets management, threat monitoring), and a unified Governance Platform (GRC system, policy management allowing regional customization, central MRM framework). Crucially, multiple independent Regional Deployment Layers exist, each operating within a specific jurisdiction's boundaries (potentially on different cloud providers or regions) to enforce local data residency laws. These layers host region-specific applications, localized data platforms with stringent governance, regional ML model serving infrastructure, and local API gateways. Managing Cross-Region Data Exchange requires a dedicated, audited strategy built upon strong data classification, regulator-approved privacy-enhancing techniques (like Federated Learning (32; 30) or differential privacy where applicable), explicit consent mechanisms, and secure transfer protocols, ensuring sensitive raw data does not inappropriately cross borders.

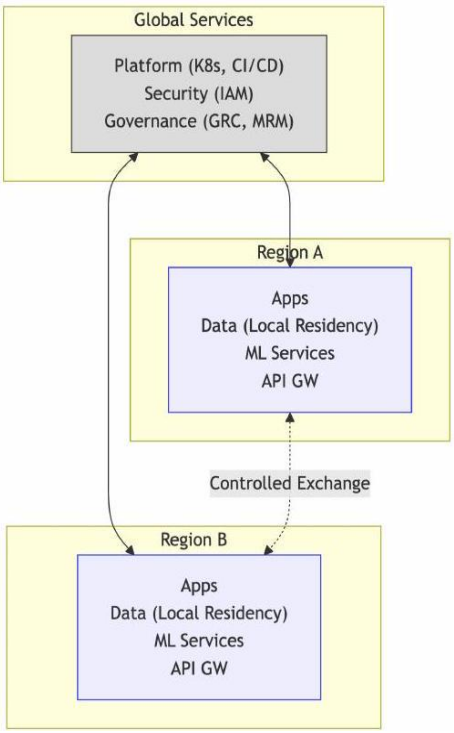

Figure 4: Proposed Federated Cloud Architecture (Blueprint 2). This architecture maintains regional isolation of applications and data while providing global shared services. Controlled data exchange between regions operates within regulatory boundaries, with each region maintaining sovereign control over local data and services.

3.2.3 Multi-Region DevOps Pipeline (Fig. 5)

This federated pipeline design standardizes initial development stages globally while enabling parallel, region-specific compliance and deployment. A Global Build and Validation Process uses common source code repositories (potentially with regional configuration overlays), executes standardized build scripts, performs global security scanning, and enforces baseline Policy-as-Code checks applicable across all regions. This produces a consistent, validated baseline artifact. Subsequently, the workflow branches into parallel Region-Specific Deployment Processes. The baseline artifact is deployed to independent regional staging environments. Validation suites tailored to specific

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

local regulations (e.g., GDPR checks in Europe, CCPA checks in California) are executed. Approval workflows involve regional compliance and risk stakeholders. Deployment strategies and monitoring thresholds can be adapted to local requirements, all orchestrated using globally consistent tooling but executed within regional boundaries. Effective Cross-Region Coordination mechanisms (e.g., release management, global change advisory boards) are vital.

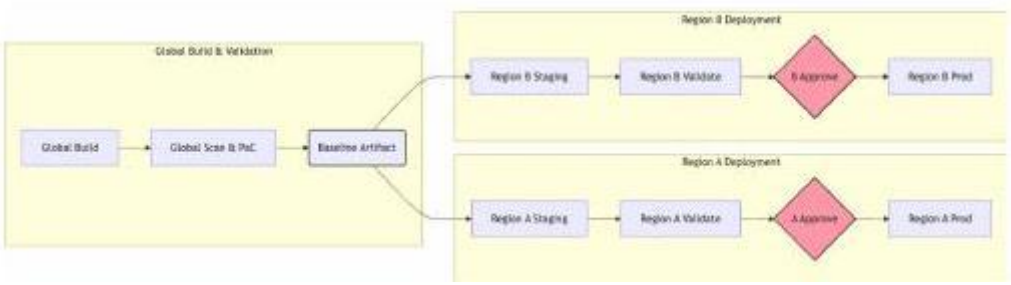

Figure 5: Multi-Region DevOps Pipeline.

3.2.4 Multi-Jurisdiction MLOps (Fig. 6)

This MLOps framework employs layered governance to manage models across diverse regulatory landscapes. A Global Governance Layer provides central oversight, including a Unified Model Inventory mapping models to applicable regulations, a Global Risk Management Framework defining baseline standards (while accommodating regional specifics like SR 11-7 vs. EBA guidelines), and a centralized, immutable Audit Trail for global reporting. Below this, a Global MLOps Platform offers standardized tooling, such as Experiment Tracking platforms, a potentially federated Feature Store architecture balancing sharing with access controls, and a central Model Registry track-ing model versions, metadata, and crucially, validation status per jurisdiction. Finally, the Regional Deployment Layer handles localized model execution and, critically, enforces region-specific validation. Models approved globally must still pass validation against local regulatory interpretations and data characteristics before deployment within that region, using local infrastructure to ensure compliance with data residency and performance needs. Regional Data Governance strictly enforces local rules. For use cases requiring cross-border insights without pooling raw data (e.g., global fraud detection), Federated Learning (Fig. 7) provides a privacy-preserving alternative (32; 30).

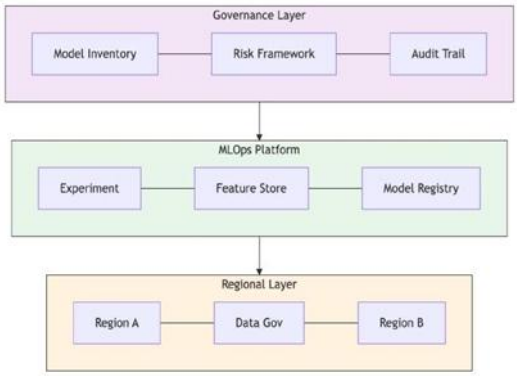

Figure 6: Proposed Multi-Jurisdiction MLOps Framework (Blueprint 2). This layered approach establishes global governance and platform services with region-specific deployment and validation. The framework enables centralized model development while ensuring regional regulatory compliance and data governance requirements are met.

3.2.5 RegTech Integration

The complexity of Blueprint 2 makes integrated Regulatory Technology (RegTech) solutions highly advantageous (5;8). Key applications include automated tools for monitoring regulatory changes across jurisdictions, robust Policy-as-Code (PaC) frameworks (like OPA (Open Policy Agent)) capable of managing and enforcing layered global and

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

regional policies within CI/CD and infrastructure (20), standardized Explainable AI (XAI) frameworks to meet diverse regulatory expectations for model transparency (22), and consolidated compliance monitoring dashboards providing a unified view across regions.

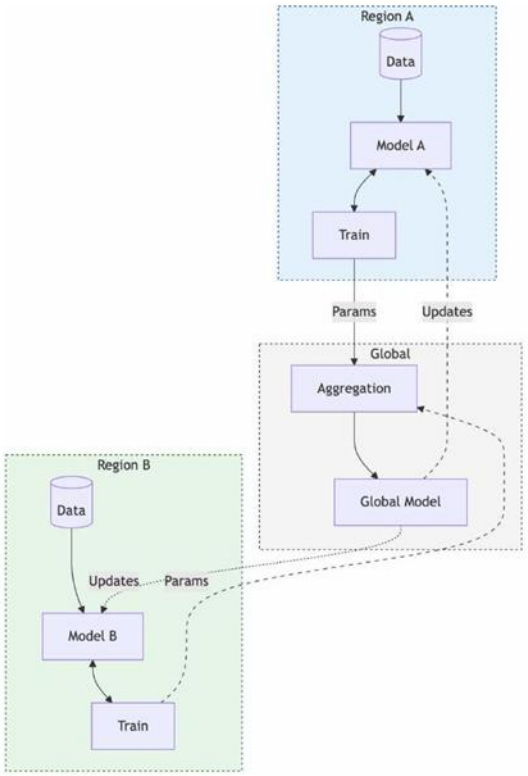

Figure 7: Proposed Federated Learning Approach for Cross-Border Use Cases (Blueprint 2). This approach enables machine learning across jurisdictional boundaries without moving raw data. Local models train on regional data, sharing only parameters with a global coordination service that aggregates insights while preserving data sovereignty.

DISCUSSION

The two blueprints present distinct, viable strategies for embedding DevOps and MLOps within regulated financial environments. Blueprint 1 offers an evolutionary path, focusing on controlled, incremental modernization centered around integrating with a persistent legacy core using APIs and establishing unified domestic data and ML governance. In contrast, Blueprint 2 provides a framework for managing the inherent complexity of multi-jurisdiction operations through federation, separating regional concerns (like data residency and local validation) from globally managed standards and platforms. The choice depends fundamentally on the institution's operational scope and legacy constraints. Table 1 provides a concise comparison.

Despite their architectural differences, both blueprints are built upon a common set of foundational principles derived from established research and best practices (21; 15; 25). Successfully achieving compliant agility hinges on consistently applying these principles. A primary principle is the necessity to integrate compliance and security checks early and continuously throughout the development lifecycle ("Shift Left"), making them integral parts of the process rather than separate, late-stage afterthoughts (35; 43). Furthermore, extensive automation of governance processes is paramount. This encompasses using Infrastructure as Code (IaC) for repeatable and auditable environment provisioning

Equally important is establishing comprehensive monitoring and observability across the entire technology stack (21). This extends beyond basic infrastructure health checks to include application performance monitoring, security event logging, tracking ML model performance metrics in production, and specifically monitoring for data and

Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

concept drift which can invalidate models over time (45). Robust observability provides the critical feedback necessary for operational stability, rapid incident response, and triggering corrective actions such as model retraining or system adjustments. Strong, integrated data governance is another cornerstone, particularly crucial given regulations like BCBS 239 (10); this involves ensuring high data quality, maintaining clear data lineage from source to consumption, implementing comprehensive data cataloging, and enforcing granular access controls to protect sensitive information (38; 46). Moreover, adherence to rigorous Model Risk Management (MRM) standards, exemplified by regulations like SR 11-7

The selection and specific adaptation of either blueprint must be guided by the institution's unique context. Key factors include the complexity of the applicable regulatory landscape (single vs. multiple jurisdictions), the significance and constraints of the existing legacy systems and associated technical debt

Table 1: Blueprint Comparison.

Dimension	Blueprint 1 (Hybrid)	Blueprint 2 (Federated)
Scenario	Domestic, Legacy Mod.	Intl., Multi-Jurisdiction
Architecture	Hybrid API-centric	Federated Cloud
DevOps	Sequential Gates	Global + Parallel Regional
MLOps	Standard SR 11-7	Multi-Gov, Fed. Learning
Data Strategy	Unified Platform	Regional Residency
Principle	Incremental Control	Global + Local Autonomy

CONCLUSION

Financial institutions continually strive to reconcile the demands of rapid technological innovation with the imperatives of stringent regulatory compliance. This paper addressed this core challenge by synthesizing current research and best practices into two actionable architectural blueprints for implementing DevOps and MLOps methodologies within a robust financial control framework. Blueprint 1 (Hybrid Architecture) provides a pragmatic path for domestic institutions focused on modernizing around persistent legacy systems, emphasizing controlled integration and unified governance. Blueprint 2 (Federated Architecture) offers a scalable model for institutions managing international operations, balancing global standards with essential regional autonomy, particularly concerning data residency and localized compliance validation.

Both blueprints underscore that achieving compliant agility is not merely about adopting tools, but about fundamentally integrating compliance and risk management throughout the entire software and model development lifecycle. While they provide concrete reference models, their successful implementation depends heavily on the specific context of the institution, including its regulatory environment, technological heritage, strategic goals, and organizational readiness. They serve as practical starting points for practitioners tasked with designing and evolving compliant, agile IT systems, and offer structured conceptual models for academics investigating the ongoing evolution of these practices in the complex financial services domain. Future research directions include quantitatively evaluating the cost-benefit trade-offs of specific patterns within these blueprints, developing more sophisticated and domain-specific Policy-as-Code frameworks for complex financial regulations, exploring the integration of advanced AI techniques for tasks like automated audit and compliance verification, and further investigating optimal organizational structures to support these technologically advanced operating models.

REFRENCES

[1] Alla, S. & Adari, S. K. (2021). Beginning MLOps with MLFlow. Apress.
[2] Anagnostopoulos, I. (2018). Fintech and regtech: Impact on regulators and banks. Journal of Economics and Business, 100, 7-25.

Research Article

[3] Apache Kafka Project] Apache Kafka Project. Apache kafka. Accessed: 2025-04-30. https: //kafka.apache.org/. Armbrust, M. et al. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.
[4] Arner, D. W., Barberis, J., & Buckley, R. P. (2016). The evolution of FinTech: A new post-crisis paradigm? Georgetown Journal of International Law, 47, 1271. Title based on journal lookup, original was truncated. Page number seems to be start page.
[5] Baier, L., Ferreira, F. J. T. E., & Tranquillini, S. (2019). Challenges for machine learning model governance. In Business Information Systems (BIS), volume 354 of Lecture Notes in Business Information Processing (pp. 275-287).: Springer.
[6] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. IEEE Software, 33(3), 42-52.
[7] Baptista, J. & Wilson, A. D. (2020). RegTech: Harnessing technology for regulatory compliance in financial services. Journal of Risk Management in Financial Institutions, 13(2), 130-141.
[8] Basel Committee on Banking Supervision (BCBS) (2010). Basel III: A global regulatory framework for more resilient banks and banking systems. Technical report, Bank for International Settlements (BIS). Revised December 2010, June 2011, and subsequent updates. Title based on document lookup, original was truncated.
[9] Basel Committee on Banking Supervision (BCBS) (2013). BCBS 239: Principles for effective risk data aggregation and risk reporting. Technical Report BCBS 239, Bank for International Settlements (BIS). Title based on document lookup, original was truncated.
[10] Board of Governors of the Federal Reserve System (2011a). SR 11-7: Guidance on Model Risk Management. Technical Report SR 11-7, Board of Governors of the Federal Reserve System. Duplicate of BoardGov2011, included as requested.
Board of Governors of the Federal Reserve System (2011b). Supervisory Letter SR 11-7: Guidance on Model Risk Management. Technical Report SR 11-7, Board of Governors of the Federal Reserve System.
[11] Breck, E., Cai, S., Nielsen, E., et al. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data) (pp. 1123-1132).
[12] Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry, 59(7), 647-659.
[13] Ebert, C. & Paasivaara, J. H. (2016). DevOps: A definition and practical approach. In Proceedings of the 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 483-486).
[14] European Banking Authority (EBA) (2019). Directive (EU) 2019/878 of the european parliament and of the council of 20 may 2019 amending directive 2013/36/eu as regards exempted entities, financial holding companies, mixed financial holding companies, remuneration, supervisory measures and powers and capital conservation measures (CRD V). Official Journal of the European Union L 150/253. Full title provided for clarity.
[15] Financial Stability Board (FSB) (2019). Third-party dependencies in cloud services: Considerations on financial stability implications. Technical report, FSB. Title based on FSB publication list, original was truncated.
[16] Financial Stability Board (FSB) (2022). Supervisory and Regulatory Approaches to Climaterelated Risks: Final Report. Technical report, FSB.
[17] Fitzgerald, B. & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176-189. Year corrected to 2017 based on journal publication date.
[18] Forsgren, N. & Fylling, M. E. (2021). Policy as Code: Continuous Compliance and Remediation. O'Reilly Media. Title based on publisher lookup, original was truncated.
[19] Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press.
[20] Goodman, B. & Flaxman, S. (2017). European Union regulations on algorithmic decisionmaking and a "right to explanation". AI Magazine, 38(3), 50-57. Title based on journal lookup, original was truncated. [Grafana Labs] Grafana Labs. Grafana: The open observability platform. Accessed: 2025-04-30. https://grafana.com/.

Humble, J. & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional.
[21] Jin, Z., Jamshidi, P., Garlan, D., et al. (2021). Continuous compliance assessment for DevOps. ACM Transactions on Software Engineering and Methodology (TOSEM), 30(4), 1-33.
[22] Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution Press.
[23] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. In Proceedings of the ACM SIGMOD Workshop on Networking Meets Databases (NetDB). Title based on paper lookup, original was truncated. Pages omitted as common for workshops. [Kubeflow Project] Kubeflow Project. Kubeflow. Accessed: 2025-04-30. https://www. kubeflow.org/. [Kubernetes Project] Kubernetes Project. Kubernetes. Accessed: 2025-04-30. https:// kubernetes.io/.
[24] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60.
[25] Martini, A. & Bosch, J. (2016). Technical debt in software development: A systematic mapping study from the perspective of researchers and practitioners. Journal of Systems and Software, 118, 198-222.
[26] McMahan, B., Moore, E., Ramage, D., et al. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) (pp. 1273-1282). Title based on conference program lookup, original was truncated.
[27] Mell, P. & Grance, T. (2011). The NIST Definition of Cloud Computing. Technical Report NIST SP 800-145, National Institute of Standards and Technology. Morris, K. (2016). Infrastructure as Code. O'Reilly Media.
[28] Myrbakken, H. & Colomo-Palacios, R. (2017). Integrating security into DevOps: A systematic literature review. In Product-Focused Software Process Improvement (PROFES), volume 10611 of Lecture Notes in Computer Science (pp. 317-331).: Springer.
[29] Newman, S. (2015). Building Microservices. O'Reilly Media. [Open Policy Agent] Open Policy Agent. Open policy agent. Accessed: 2025-04-30. https:// www.openpolicyagent.org/.
Otto, B. (2011). Data governance. Business & Information Systems Engineering, 3(4), 241-244.
[30] Pahl, C. & Jamshidi, P. (2016). Microservices: A systematic mapping study. In Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER), Vol. 2 (pp. 137-146).
[31] Paleyes, A., Urma, R., & Lawrence, N. D. (2022). Challenges in deploying machine learning: a survey of case studies. ACM Computing Surveys (CSUR), 55(6), 1-29. Title based on journal lookup, original was truncated. Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. Computer, 40(11), 38-45. [Prometheus Monitoring] Prometheus Monitoring. Prometheus - monitoring system & time series database. Accessed: 2025-04-30. https://prometheus.io/.
[32] Rahman, A. A., Parnin, C., & Williams, L. (2019). Exploring the integration of security practices in DevOps. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 241-250).
[33] Saleiro, P., Kuester, B., Hinkson, L., et al. (2018). Aequitas: A bias and fairness audit toolkit. arXiv preprint arXiv:1811.05577.
[34] Schelter, S., Grafberger, D. V., & Boden, C. (2018). Automating large-scale data quality verification. Proceedings of the VLDB Endowment, 11(12), 1781-1794.
[35] Sculley, D., Holt, G., Golovin, D., et al. (2015). Hidden technical debt in machine learning systems. In Advances in Neural Information Processing Systems 28 (NIPS 2015) (pp. 2503-2511).
[36] Sharma, B. & Thulasiram, R. K. (2017). Cloud computing in the financial services sector: A risk and compliance perspective. Journal of Risk and Financial Management, 10(3), 15. Title based on journal lookup, original was truncated.
[37] Thakor, A. V. (2020). Fintech and banking: What do we know? Journal of Financial Intermediation, 41, 100833.
[38] Woods, E. (2016). Designing API-First Applications. O'Reilly Media.