Scaling MLOps with Self-Service Platforms

Architectures for Automation, Governance, and Velocity

A blueprint for operationalizing ML at scale in regulated financial environments

BY: - Rajeev Reddy Chevuri

Campbellsville University, USA

COnf42 MLOps 2025

Today's Agenda

1 The Evolution of ML Operations

From model-building to enterprise-grade production systems

2 Self-Service Platform Architecture

Core components, integration patterns, and governance models

3 Measured Impact & Implementation Roadmap

Case studies, metrics, and actionable next steps

4 Future Trends & Strategic Positioning

AutoML, serverless architecture, and edge deployments

The MLOps Maturity Shift

The financial industry has evolved from:

- Siloed data science experiments
- Manual deployment processes
- Inconsistent environments
- Ad-hoc governance
- Limited reproducibility

To **integrated platforms** that standardize the complete ML lifecycle while ensuring regulatory compliance.

Why Self-Service Platforms Matter

71%

Reduction in Data Prep Time

Through standardized pipelines and reusable components

45%

Lower Training Costs

Via optimized resource allocation and compute orchestration

38%

Improved Deployment Success

Through integrated CI/CD and consistent environments

Self-service platforms don't just improve efficiency—they transform how financial institutions manage ML risk and compliance.

Core Platform Architecture

Laying the groundwork for robust, scalable, and compliant machine learning operations in financial institutions

A robust core platform architecture is essential for operationalizing ML at scale in financial institutions. It provides a stable, secure, and compliant foundation, supporting the entire ML lifecycle while addressing regulatory challenges and simplifying MLOps.

Foundational Infrastructure

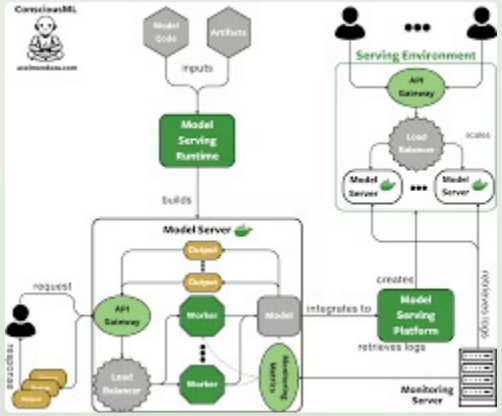
Scalable compute (CPUs, GPUs), resilient storage, and secure networking. Key considerations include cloud-native/on-premises, containerization (Kubernetes), and data security.

Data & Feature Management

Unified layer for data ingestion, processing, and feature engineering. Includes validation, versioning, and a centralized feature store for consistent, reusable, and auditable ML data assets.

ML Model Lifecycle Management

Tools for model experimentation, training, versioning, deployment, and continuous monitoring. Covers CI/CD for ML, model registries, automated testing, and observability for regulatory compliance.



Reference Architecture: Self-Service ML Platform

A comprehensive self-service ML platform is crucial for financial services. It streamlines operations by abstracting infrastructure complexity, allowing data scientists to focus on model development. The platform ensures stringent regulatory controls, facilitates comprehensive audit trails for model lineage and deployment, and establishes clear separation of duties—all vital for navigating financial compliance and enhancing security.

Key Platform Components

Containerized Orchestration

Kubernetes-based scheduling with custom resource definitions (CRDs) for ML workloads, enabling consistent execution across environments with resource isolation.

Distributed Compute Framework

Spark and Dask clusters providing scalable data processing with built-in lineage tracking, meeting regulatory data provenance requirements.

Model Registry & Artifact Store

Versioned model storage with compliance metadata, approval workflows, and cryptographic validation of model artifacts.

CI/CD Pipeline Integration

Automated testing, validation, and deployment pipelines with mandatory security scanning and approval gates.

The Governance Layer

Automated Policy Enforcement

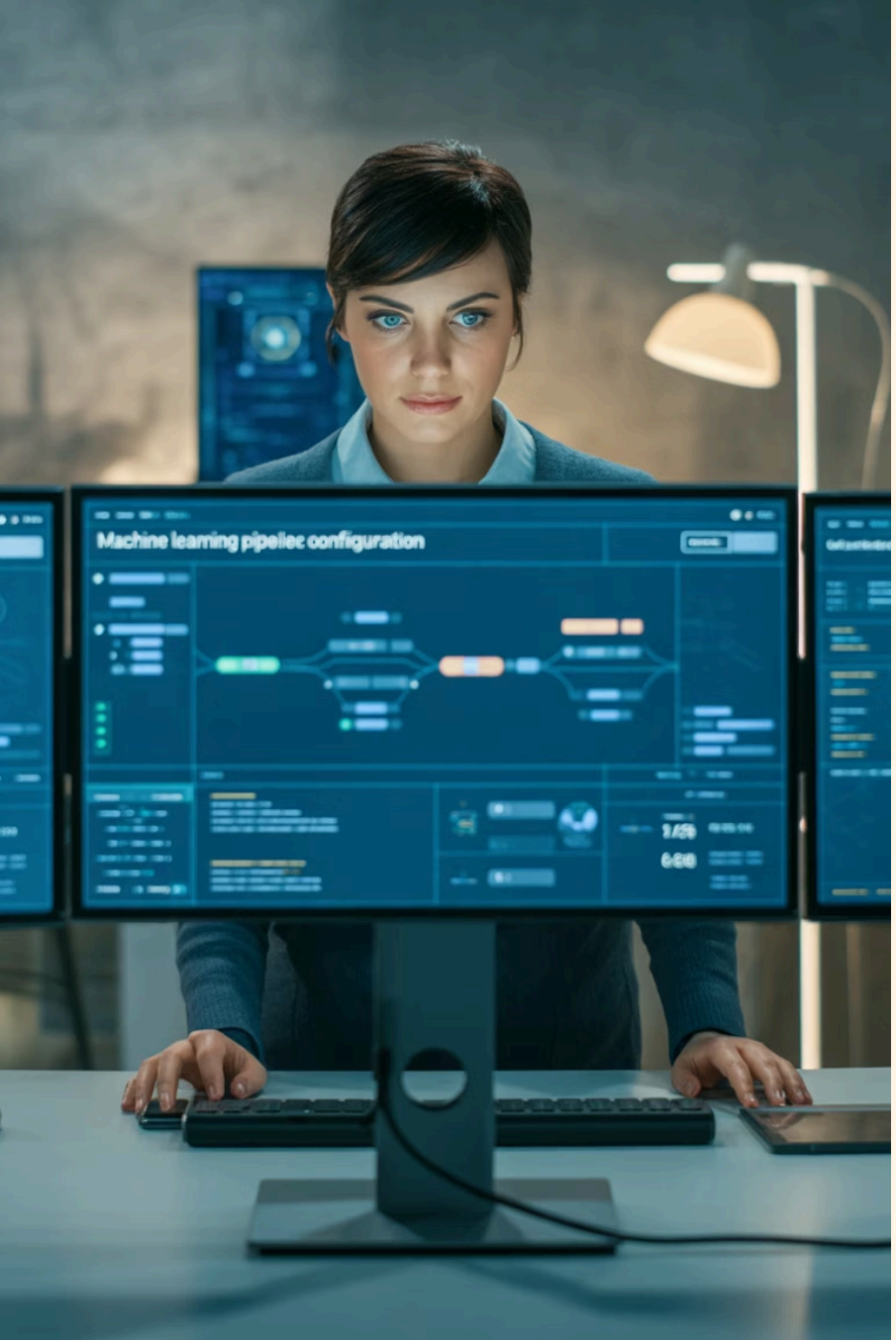
Runtime enforcement of data access controls, model risk limits, and security requirements

Compliance Documentation

Auto-generated model cards, lineage diagrams, and validation reports

Audit & Observability

Comprehensive logging, monitoring, and explainability tools



Self-Service Experience: Developer View

Abstraction with Control

- Curated environments with pre-approved packages
- Template-based experiment configuration
- Infrastructure-as-code for custom needs

Seamless Workflow

- One-click deployment to staging environments
- Automated compliance checks and documentation
- Integrated debugging and monitoring tools

Implementation: Building vs. Buying

In-House Development

Pros: Complete customization for financial regulations, integration with legacy systems

Cons: 12-18 month timeline, specialized DevOps and platform engineering resources required

Commercial Platforms

Pros: Faster time-to-value (3-6 months), pre-built compliance features

Cons: Potential vendor lock-in, integration challenges with proprietary financial systems

Hybrid Approach

Recommended: Core platform from vendors with custom governance layers and integration points

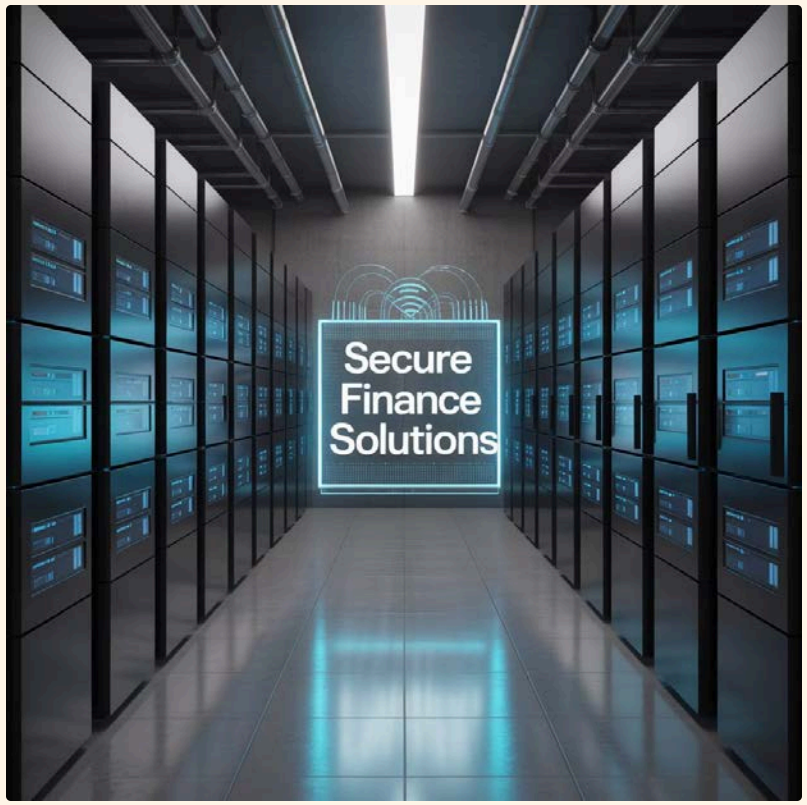
Typical implementation timeframe: 6-9 months to full production readiness

The Future of MLOps

Emerging architectures that will reshape ML operations in financial services

Edge-Native ML Architecture

Financial institutions are increasingly moving ML workloads to edge environments to reduce latency, enhance privacy, and lower cloud costs.



Traditional Cloud ML

Centralized training and inference

Hybrid Deployment

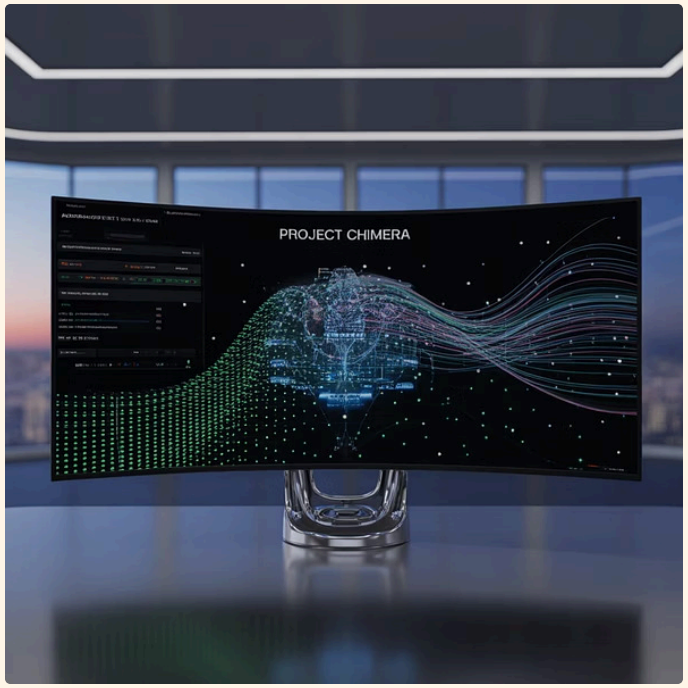
Cloud training, edge inference

Edge-Native ML

Federated learning, on-device models

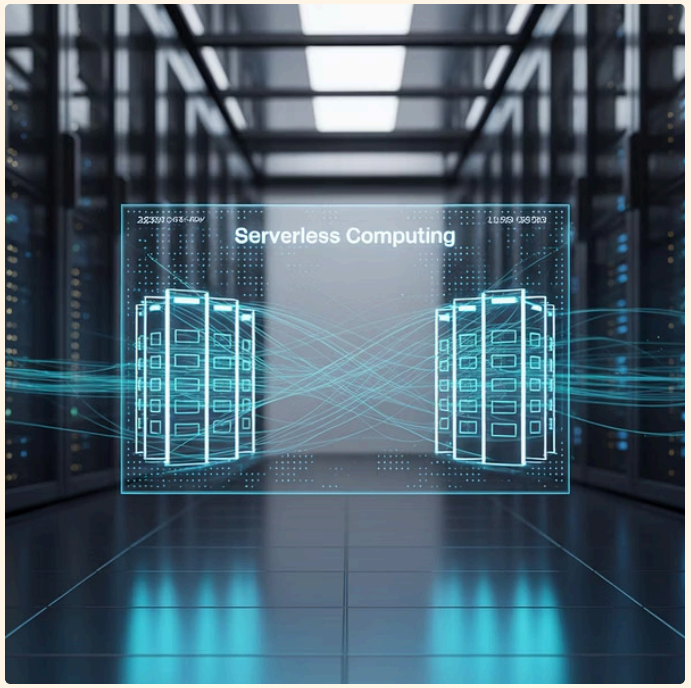
Edge deployments deliver **40% lower latency** for time-sensitive financial applications like fraud detection and trading algorithms.

Emerging Trends in Financial MLOps



Enterprise AutoML

Automated model selection and hyperparameter optimization with compliance guardrails and explainability requirements baked in.



Serverless Training

Event-driven, pay-per-computation ML training removing infrastructure management while maintaining regulatory controls.

Privacy-Preserving ML

Federated learning and homomorphic encryption enabling model training across data silos without exposing sensitive financial data.

Key Takeaways & Next Steps

Self-service platforms are essential

Not optional for scaling ML in regulated environments

Governance must be built-in

Not bolted-on after implementation

Start with a minimum viable platform

Then expand capabilities incrementally

Plan for the edge-native future

Architecture decisions today will impact flexibility tomorrow

Your 90-Day Roadmap

1. Audit current ML workflows and compliance gaps
2. Establish platform requirements with both technical and regulatory stakeholders
3. Prototype with small, cross-functional team (2-3 sprints)
4. Implement MVP focused on highest-friction areas
5. Measure success against clear KPIs: deployment time, governance efficiency, and developer satisfaction

Thank You