

# The MLOps Backbone

Feature Stores and the Engineering of 'Now'

● AI/ML Technical Documentation ● Production Systems ● Fintech Innovation ●

Technical Documentation Series

December 2024

---

## Executive Summary

AI models get the glory, but infrastructure does the work. Fraud detection is useless if it can't access transaction history in 50ms. Chime and Affirm built world-class Feature Stores - systems serving data to models at light speed for real-time decisions.

<b>&lt;10ms</b> Feature Latency	<b>Petabytes</b> Historical Data	<b>99.99%</b> Uptime Target
<b>Chime MLKit</b> Serverless Store	<b>Affirm</b> Time Travel	<b>Zero Skew</b> Train-Serve Gap

---

## The Infrastructure of Inference



In AI fintech, the model is mathematically perfect but useless without infrastructure. A fraud model needs transaction history in <50ms before card swipe approval. MLOps architectures make this possible.

---

## Chime: MLKit & Serverless Feature Store

### • Online vs Offline Feature Store

- **Offline Store (S3):** Petabytes of historical data for model training
- **Online Store (DynamoDB):** Recent critical data for real-time inference

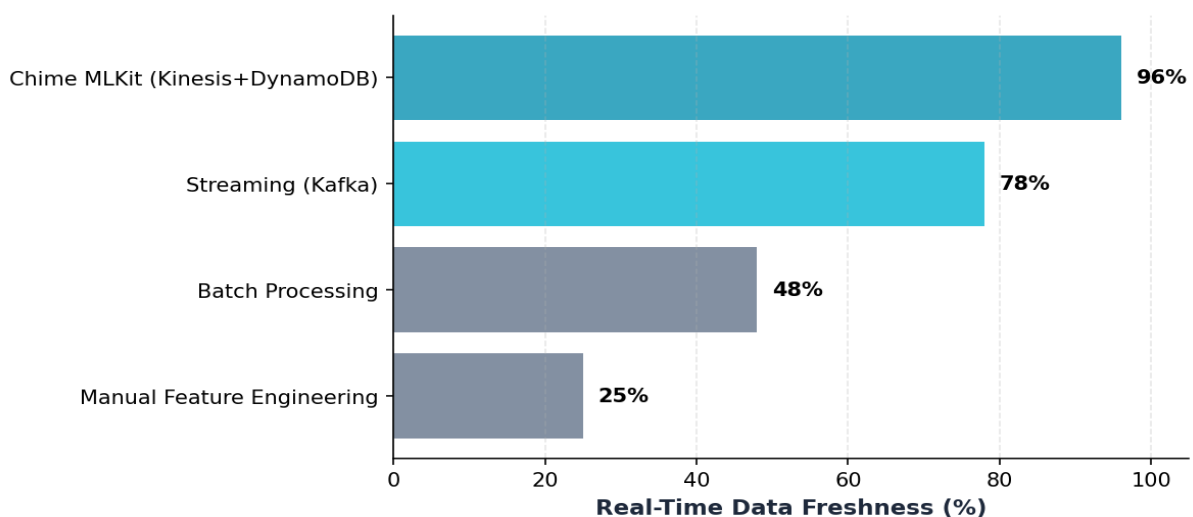
- **Speed Layer:** Sub-10ms latency for live decisions
- **Amazon Kinesis:** Streams events to keep stores in sync

## • The Look-Back Problem

Challenge: Offline store thinks user has \$500, online store (due to lag) shows \$0. Wrong model decision results. Kinesis streaming ensures single-digit millisecond sync - model always sees 'state of world' as it exists right now.

## • MLKit: Configuration over Code

- **Declarative Framework:** YAML config instead of custom Python
- **Auto-Deployment:** Training pipeline + API endpoints automatic
- **Feature Store Connection:** Plumbing abstracted away
- **Speed to Production:** Weeks → Days for new fraud models



**Chime's Breakthrough:** MLKit declarative approach reduced fraud model deployment from weeks to days. Single-digit millisecond latency ensures models see real-time account state, enabling instant fraud detection during card swipes.

## Affirm: Time Travel & Point-in-Time Correctness

### • The Leaky Data Problem

Training on historical loans can accidentally leak future information. If 2023 training data includes 'Total Repaid' from 2024 payments, model cheats. Affirm prevents this with 'Time Travel' feature store.

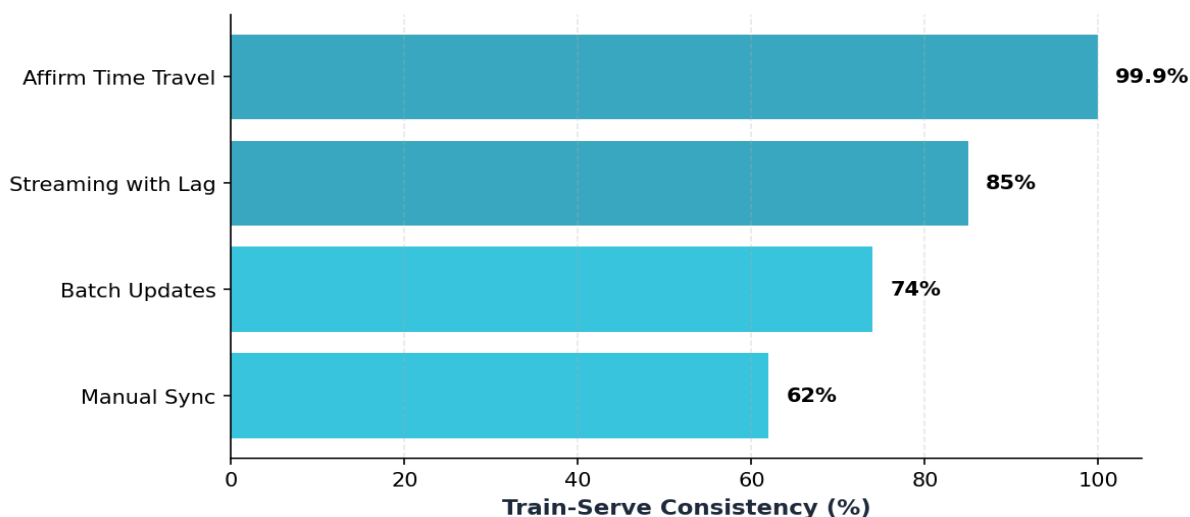
### • Versioning Every Event

- **Timestamp Versioning:** Every data point has precise timestamp
- **Historical Reconstruction:** Recreate exact world state on any date

- **Future Data Hidden:** Only show data that existed at training time
- **Training-Serving Skew Prevention:** Offline = Online behavior guaranteed

### • Fast Rollbacks & Feature Flags

- **Feature Toggles:** Enable/disable features without redeployment
- **Instant Rollback:** Turn off problematic features immediately
- **99.99% Uptime:** Critical during Black Friday traffic
- **Risk Mitigation:** Deploy new crypto balance feature, roll back if issues



**Point-in-Time Correctness:** Affirm's Time Travel feature store reconstructs exact historical state, preventing data leakage. 1% training-serving skew = millions in bad loans. Zero skew = perfect consistency.

## Competitive Advantage

- **Not Just Models:** Advantage is feeding models with fresh data
- **Feature Store = Heart:** High-speed pumping system for intelligence
- **Real-Time Decisions:** Sub-50ms inference at scale
- **Configuration Over Code:** Declarative MLOps for rapid iteration