From Data Pipelines to AI Products Operationalising Machine Learning and Generative AI in Enterprise Decision Platforms

By Bharat Kumar Reddy Karumuri
General Motors
Conf42 Database DevOps 2026

The Challenge

The Gap Between Experimentation and Production

Most enterprises have data. Most have models. Few have **production-grade AI**. The bottleneck is not talent or tooling it is the absence of engineering discipline that connects data pipelines, ML workflows, and deployed products into a single, reliable system.

Isolated Analytics

Siloed projects that never reach production

Unreliable Pipelines

Brittle data flows with no observability

Governance Gaps

Models deployed without lineage or auditability

Agenda

What We Will Cover

01

Database DevOps as AI Backbone

CI for data, schema evolution, pipeline observability

02

Resilient Cloud-Native Pipelines

Designing data and ML pipelines for production scale

03

MLOps for Faster, Safer Deployments

Versioning, automation, and continuous model delivery

04

Generative AI & RAG Architectures

LLMs integrated with reliable data infrastructure

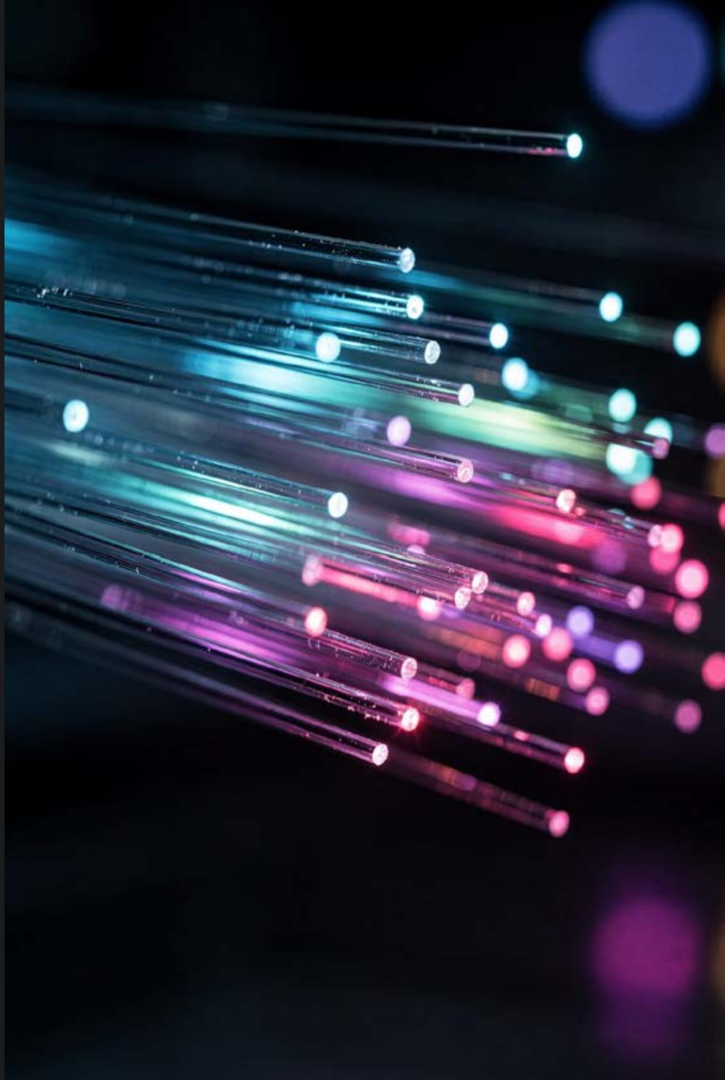
05

Governance, Quality & Observability

Data quality, model lineage, and real-time monitoring

Database DevOps Is the Backbone of Scalable AI

AI systems are only as reliable as the data infrastructure beneath them. Applying DevOps discipline to data continuous integration, schema versioning, automated testing transforms fragile pipelines into production-grade foundations.



The Three Pillars of Database DevOps for AI

Continuous Integration for Data

Automated validation, schema contract testing, and data quality gates built into every pipeline commit catching issues before they propagate downstream.

Schema Evolution Management

Versioned, backward-compatible schema changes managed via migration frameworks, ensuring AI features remain stable as data models evolve over time.

Pipeline Observability

End-to-end lineage tracking, SLA monitoring, and alerting across every data transformation giving teams full visibility from ingestion to model serving.

Designing Resilient Cloud-Native Pipelines

A production pipeline treats every stage as a versioned, testable, and independently deployable unit not a monolithic script.

Key Design Principles

Idempotency

Pipelines can be safely re-run without corrupting state

Modularity

Stages are decoupled, enabling independent scaling and testing

Fault Tolerance

Dead-letter queues and retry logic prevent silent failures

Versioned Datasets

Every dataset snapshot is catalogued for reproducibility

MLOPS

Treating Models as Deployable Engineering Assets

MLOps closes the loop between experimentation and production. By applying software engineering rigour to models versioning, automated testing, staged rollouts teams accelerate delivery whilst reducing deployment risk.

Model Versioning

Every model artefact is tracked with metadata, lineage, and performance benchmarks

Automated CI/CD

Model promotion gates enforce validation before staging and production rollout

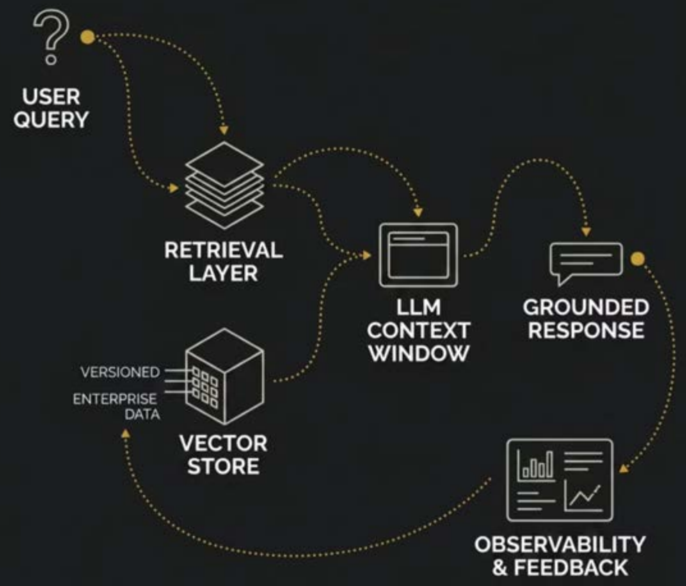
Canary & Shadow Deployments

New models serve a traffic slice first, with live comparison against the baseline

Generative AI Meets Production Data Infrastructure

Large Language Models and Retrieval-Augmented Generation unlock new decision-making capabilities but only when grounded in clean, governed, and reliably delivered enterprise data.

LLM + RAG Architecture for Enterprise Decisions



Why RAG Over Fine-Tuning Alone?

- **Freshness**: Retrieval surfaces up-to-date enterprise knowledge without retraining
- **Auditability**: Every answer is traceable to a source document
- **Cost Efficiency**: Reduces expensive fine-tuning cycles for domain adaptation
- **Governance**: Access controls on the data store extend naturally to LLM responses

Data Quality & Governance

You Cannot Govern What You Cannot Observe

- ## Data Quality Gates

  Schema validation, null checks, distribution drift detection, and referential integrity enforced at ingestion and transformation layers.

- ## End-to-End Lineage

  Full traceability from raw source to model prediction essential for debugging, compliance audits, and root-cause analysis.

- ## Access & Compliance Controls

  Role-based access, PII masking, and data residency policies integrated into the pipeline not bolted on afterwards.

- ## Model & Data Observability

  Unified monitoring across data freshness, feature drift, model performance degradation, and inference latency in production.

Observability Across the Full AI Stack

Silent failures in AI systems are costly. A model that degrades undetected erodes business trust far faster than a visible outage.

- **Data Layer**

  Pipeline SLAs, schema drift alerts, quality scores

- **Feature Layer**

  Distribution shift, freshness checks, missing value rates

- **Model Layer**

  Prediction drift, latency, accuracy, and business KPI impact

Scaling AI for Real-Time Decision Making

Batch intelligence is no longer sufficient. Modern decision platforms require sub-second inference, event-driven feature computation, and seamless integration with both transactional and analytical systems.

REAL-TIME AI

Architecture Patterns That Scale

Every component must be independently scalable and observable — a single slow dependency breaks the real-time contract.

Key Enablers

Online Feature Stores

Low-latency key-value lookups serving pre-computed features at inference time

Streaming Pipelines

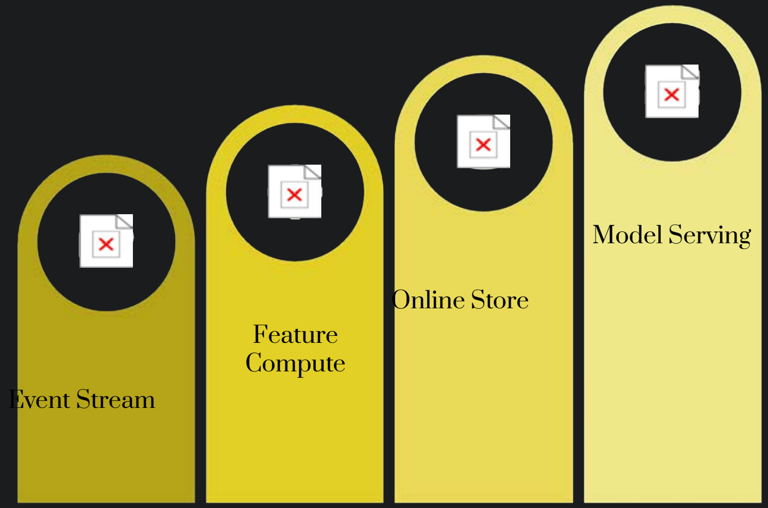
Event-driven feature engineering with Kafka or Flink for near real-time freshness

Model Serving Infrastructure

Autoscaling inference endpoints with SLA-bound latency budgets and circuit breakers

HTAP Integration

Hybrid transactional-analytical processing enables AI decisions within operational workflows



Key Takeaways

From Experimentation to Measurable Business Impact

- ### Treat Data, Models, and Infrastructure as Versioned Assets

  Engineering discipline not model sophistication is the differentiator between AI experiments and AI products.

- ### Database DevOps Is Not Optional for AI

  CI for data, schema governance, and pipeline observability are foundational not nice-to-haves for reliable AI at scale.

- ### Governance and Observability Must Be Built In

  Lineage, quality gates, and monitoring across data and model layers are what make AI systems auditable, trustworthy, and sustainable in production.

Thank You!