

AI Productionization

Reference Architecture

This brief outlines a reusable reference pattern for discovery, architecture, implementation, and operational rollout.

Problem

AI prototypes often lack evaluation gates, serving reliability, release ownership, monitoring, drift detection, and rollback plans.

Productionization turns a useful model into an operated system with measurable behavior and governance.

Typical Architecture

Data sources feed feature or data pipelines, models, evaluations, registry, serving API, monitoring, feedback loops, and retraining workflows.

Components

Evaluation harness, model registry, release gates, inference API, monitoring, drift checks, incident playbooks, rollback path, and governance records.

Security Considerations

Control training and inference data access, avoid logging sensitive prompts or outputs, review model behavior, and secure serving endpoints.

Integration Pattern

Expose model capabilities through versioned APIs and evented feedback while keeping model lifecycle controls separate from consuming applications.

Implementation Approach

Baseline current model behavior, define production acceptance criteria, add monitoring and release gates, pilot serving, then formalize operating ownership.

Operational KPIs

Accuracy or task quality, latency, error rate, drift score, evaluation pass rate, rollback time, model version adoption, and incident rate.

Deployment Model

A controlled serving environment with CI/CD, registry integration, secrets management, observability, and operational feedback capture.

BMDigitals Engagement Model

Production readiness review, evaluation design, serving architecture, monitoring implementation,

release governance, and managed model operations.

Reference architecture. Final implementation depends on the client's operational and technical environment.

