Scenario Overview

Zero Margin Limited has developed AI models for a major retail client, including:

1. Demand Forecasting Model: Predicts sales for product categories.
2. Customer Segmentation Model: Groups customers by purchasing behavior.
3. Product Recommendation Engine: Suggests items to customers.

These models, validated in a research environment, must now be deployed to production for real-time predictions with periodic updates. The client requires 99.9% uptime and continuous performance monitoring. This document outlines the MLOps architecture, versioning strategy, monitoring system, and governance framework to meet these needs.

1. MLOps Architecture

We propose a Google Cloud Platform (GCP)-based MLOps architecture leveraging managed services for scalability, reliability, and security. Data flows from sources (CRM, ERP, logs) into a Data Lake on BigQuery and Cloud Storage, processed via Dataflow/Dataproc, and stored in a Feature Store (Vertex AI Feature Store) for training. Training pipelines (Vertex AI Pipelines) train models, which are versioned in Vertex AI Model Registry and deployed on GKE clusters as microservices. CI/CD pipelines (Cloud Build/Deploy) automate updates, and Grafana/Prometheus provide observability.

Architecture Diagram

The diagram below illustrates the system components and their interactions:

- Data Ingestion: CRM, ERP, and logs feed into a Data Lake via Dataflow.
- Feature Store: Vertex AI Feature Store manages features for training and inference.
- Training Pipeline: Vertex AI Pipelines handle training, versioning models in the Model Registry.
- CI/CD: Cloud Build/Deploy automated deployment with canary rollouts.
- Serving: GKE clusters (multi-zone, autoscaling) serve predictions via REST/gRPC APIs.
- Monitoring: Prometheus/Grafana track metrics, with alerts for performance issues.
- Security: VPC, IAM, and Secret Manager ensure secure operations.

Scalability, Reliability, and Security

Scalability: GKE clusters autoscale with multi-zone deployment, and Dataflow/BigQuery scale for data processing.

Reliability: Kubernetes health checks, multi-zone GKE, and canary deployments ensure 99.9% uptime. If a new model version fails, traffic reverts to the stable version.

Security: Private VPCs, Cloud IAM for least-privilege access, encryption (KMS for secrets, SSL/TLS for transit), and GDPR-compliant storage secure the system. Secrets are managed via Secret Manager, and access is audited via Cloud Audit Logs.

2. Data and Model Versioning

Training Data Versioning
Raw datasets are stored in Cloud Storage/BigQuery, versioned using DVC (with GCS as remote storage) or BigQuery table snapshots. Each training run links to a specific data version via Git metadata, ensuring reproducibility.

Model Versioning
Models are versioned in Vertex AI Model Registry, with artifacts (weights, metrics) stored as Docker images in Artifact Registry. Versions are tagged semantically (e.g., recommendation-v2), and metadata (hyperparameters, data hashes) is logged for lineage.

Updates and Rollbacks
New model versions deploy via canary rollouts using Cloud Deploy. A small percentage of traffic tests the new version; if stable, traffic shifts fully. If issues arise, traffic reverts to the previous version. Versioned endpoints (e.g., /recommendation/v2) allow rollbacks without downtime.

Lineage and Reproducibility
Vertex AI Pipelines track lineage (code, data, configurations) automatically. All pipeline components are stored in Git, and training environments are containerized (Docker) with fixed library versions for reproducibility.

3. Monitoring and Maintenance

Key Metrics

Demand Forecasting: MAE, MAPE, RMSE, data drift (KL-divergence), latency, throughput.
Customer Segmentation: Silhouette score, cluster distribution, feature drift, latency.
Product Recommendation: Hit rate, click-through rate, revenue uplift, latency, error rate.

Alerting System

Prometheus collects metrics (via Managed Service for Prometheus), visualized in Grafana dashboards. Alerts trigger on thresholds (e.g., accuracy drop >5%) via email/Slack. The architecture diagram now explicitly shows the retraining loop triggered by alerts.

Retraining Strategy

Models retrain on a schedule (weekly/seasonal) or when alerts (e.g., data drift) fire. A Cloud Function triggers a Vertex AI Pipeline to retrain, evaluate, and deploy the new model via canary rollout if performance improves. Underperforming models are discarded.

Sample Monitoring Dashboard Mockup

Top Row: Model Accuracy and Inference Latency over time.

Bottom Row: Data Drift (input distribution shift) and Anomaly Rate (drift detector flags). Alerts are set on thresholds (e.g., accuracy drop, high drift).

4. Documentation and Governance

Model Documentation
Each model has a Model Card detailing purpose, data schema, training data, performance, limitations, and fairness assessments. Documentation is stored in a wiki and updated per version.

Governance Framework
A review board (data science lead, privacy officer, dev-ops) approves new models/versions. CI/CD pipelines enforce approval gates before deployment. The architecture diagram now includes a "Governance Approval" step before deployment.

Compliance with Standards
GDPR compliance is ensured via data minimization, encryption, and consent logging. Data subjects' rights (access, deletion) are supported via a service for data requests.

Transparency and Auditing

Model releases are tracked with change logs in Git. Access logs (Cloud Audit Logs) provide audit trails. Regular reviews reassess compliance, and privacy notices inform users of profiling logic.