MLOps Project Summary

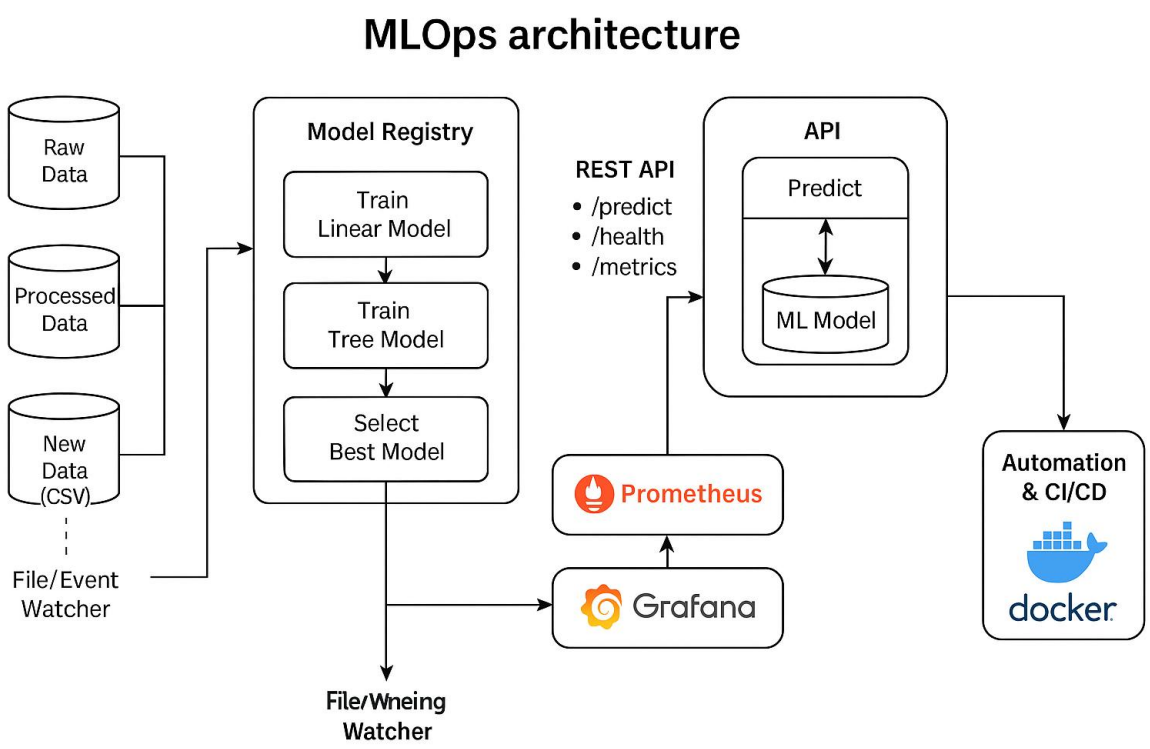
End-to-End MLOps Pipeline for California Housing Regression

Objective

Build a reproducible, automated, and observable MLOps pipeline that trains, tracks, packages, deploys, and monitors a machine learning model using the California Housing dataset.

High-Level Architecture

1. **Data Layer**
   - **Raw Data**: Fetched using fetch_data.py and stored in data/raw/
   - **Processed Data**: Cleaned and split using preprocess.py into data/processed/
   - **New Data**: Dropped into data/new/ to trigger retraining
2. **Model Training & Tracking**
   - **Training Scripts**: train_linear.py, train_tree.py
   - **Experiment Tracking**: MLflow logs metrics, parameters, and artifacts
   - **Model Selection**: select_best_and_register.py picks best model and registers it
3. **API Layer**
   - **FastAPI App**: Serves predictions via /predict endpoint
   - **Swagger & ReDoc**: Auto-generated interactive docs
   - **Health & Metrics**: /health and /metrics endpoints for observability
4. **Automation & Deployment**
   - **Docker Compose**: Orchestrates services locally
   - **GitHub Actions**: CI/CD pipeline for build, test, and deploy
   - **DVC**: Tracks and versions raw data
5. **Monitoring & Observability**
   - **Prometheus**: Scrapes metrics from FastAPI
   - **Grafana**: Visualizes request counts, latency, error rates
   - **Custom Dashboards**: FastAPI observability dashboard (ID: 16110)
6. **Retraining Trigger**
   - **File Watcher**: Monitors data/new/ using PollingObserver
   - **Trigger Logic**: On new .csv, runs full pipeline via run_training_pipeline.py



Tools & Technologies

Category	**Tools Used**
Language	Python 3.8+
API Framework	FastAPI
Experiment Tracking	MLflow
Containerization	Docker, Docker Compose
CI/CD	GitHub Actions
Monitoring	Prometheus, Grafana
Data Versioning	DVC
Logging	Custom logger via utils/logger.py

Key URLs (Local Deployment)

Service	**URL**

FastAPI	http://localhost:8000
Swagger UI	http://localhost:8000/docs
MLflow UI	http://localhost:5000
Prometheus	http://localhost:9090
Grafana	http://localhost:3000

Outcomes

- Modular, reproducible ML pipeline
- Automated retraining on new data
- Real-time monitoring and metrics
- Scalable deployment via Docker
- CI/CD integration for production readiness