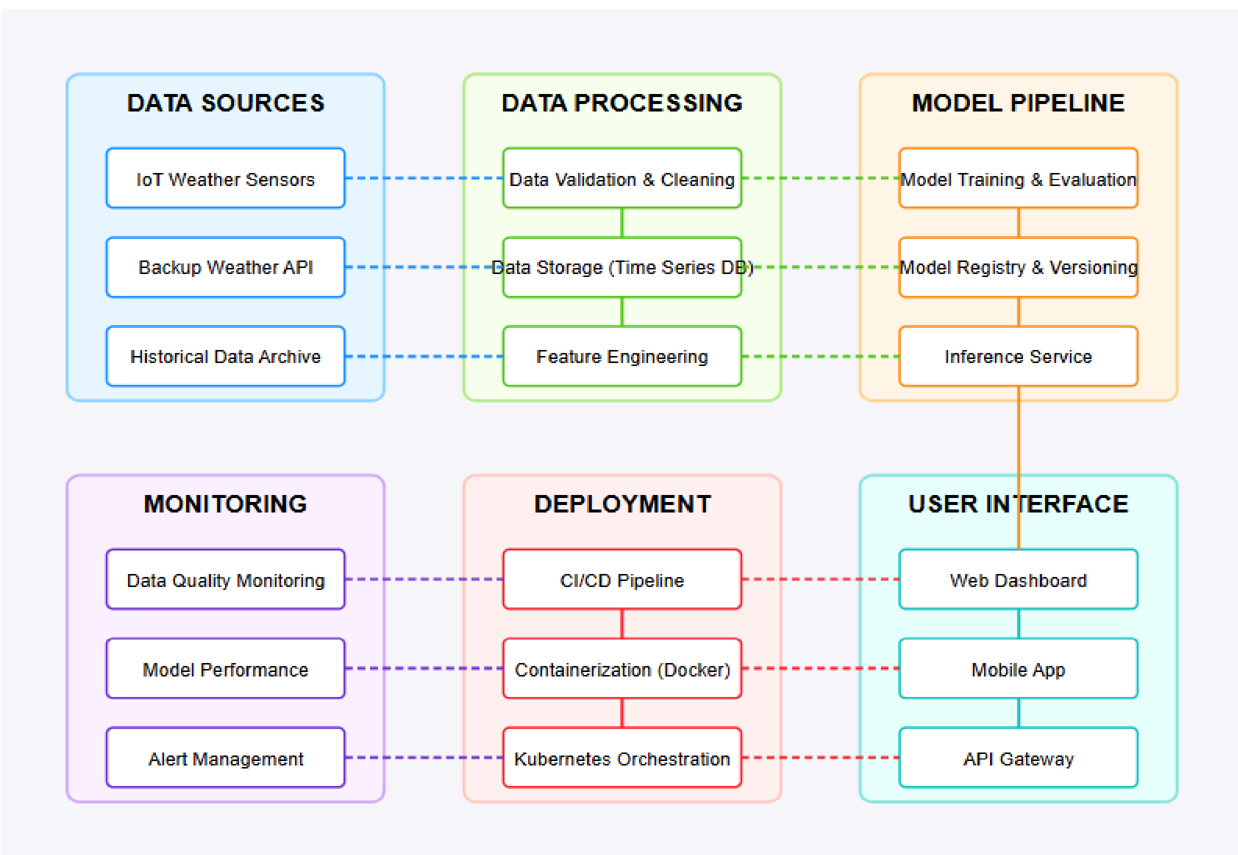
MLOps System Design for Real-Time Weather Prediction

Resilient Architecture for Smart Agriculture Applications

1. System Overview

The proposed MLOps system is designed to deliver reliable 21-day rainfall probability forecasts for agricultural decision-making using real-time IoT sensor data. The system handles data from potentially malfunctioning devices, ensures high availability, and maintains prediction quality through continuous monitoring and model updates.

2. System Architecture Diagram



3. Component Description

3.1 Data Sources

IoT Weather Sensors

- Function: Collect real-time weather data (temperature, humidity, pressure, wind speed, cloud cover)
- Frequency: Data transmitted every 1 minute
- Resilience: Multiple sensors deployed in proximity for redundancy
- Challenges: Handles intermittent connectivity and sensor malfunctions

Backup Weather API

- Function: Secondary data source when IoT sensors fail
- Operation: Automatically activated when IoT data quality falls below threshold
- Sources: Integration with national/regional weather services
- Limitation: Less granular than on-site sensors but ensures continuity

Historical Data Archive

- Function: Stores long-term weather patterns and seasonal trends
- Usage: Provides baseline data for model training and validation
- Structure: Indexed by location and date for efficient retrieval
- Value: Enables detection of anomalous readings from IoT sensors

3.2 Data Processing

Data Validation & Cleaning

- Function: Identifies and handles anomalous sensor readings
- Methods:
  - Statistical outlier detection (Z-score, IQR)
  - Cross-sensor validation
  - Physical constraint checking (e.g., humidity cannot exceed 100%)
- Actions: Flags, corrects, or removes erroneous data
- Output: Clean, consistent data stream

Data Storage

- Function: Persistently stores all weather data
- Implementation: Time-series database (InfluxDB)
- Features:
  - High write throughput for 1-minute interval data
  - Efficient time-based queries
  - Built-in downsampling for historical analysis
- Retention: Raw data retained for 90 days, aggregated data indefinitely

Feature Engineering

- Function: Transforms raw data into model-ready features
- Operations:
  - Temporal feature extraction (time of day, day of year, seasonality)
  - Rolling window statistics (24-hour averages, min/max)
  - Interaction term calculation
  - Feature normalization
- Schedule: Real-time for inference, daily batch for model retraining

3.3 Model Pipeline

Model Training & Evaluation

- Function: Creates and validates prediction models
- Frequency: Weekly retraining with daily evaluation
- Methods:
  - Automated hyperparameter tuning
  - Cross-validation for robustness
  - Multi-metric evaluation (accuracy, F1, ROC-AUC)
- Output: Candidate models for potential deployment

Model Registry & Versioning

- Function: Manages model versions and deployment history
- Features:
  - Model metadata tracking
  - Performance metrics logging
  - A/B testing capability
  - Rollback functionality
- Security: Access controls and audit logging
- Integration: Connected to CI/CD pipeline for automated deployment

Inference Service

- Function: Generates 21-day rainfall probabilities
- Operation: RESTful API endpoint
- Performance:
  - Low latency (< 500ms response time)
  - Horizontal scaling for peak demands
  - Batch processing for 21-day forecasts
- Output: Daily rain probabilities with confidence intervals

3.4 Monitoring

Data Quality Monitoring

- Function: Continuously assesses sensor data integrity
- Metrics:
  - Missing value frequency
  - Value distribution drift
  - Inter-sensor consistency
- Actions: Triggers alerts and activates backup data sources when needed

Model Performance Monitoring

- Function: Tracks prediction accuracy over time
- Methods:
  - Ground truth comparison (when rain data becomes available)
  - Prediction drift detection
  - Feature importance stability
- Threshold: Triggers retraining when accuracy drops below predefined threshold

Alert Management

- Function: Notifies appropriate personnel of system issues
- Channels: Email, SMS, in-app notifications
- Severity Levels:
  - Critical: System down, multiple sensor failures
  - Warning: Performance degradation, single sensor issues
  - Informational: Routine maintenance, model updates
- Escalation: Automated escalation for unacknowledged critical alerts

3.5 Deployment

CI/CD Pipeline

- Function: Automates testing and deployment
- Triggers:
  - Code commits to repository
  - Scheduled model retraining
  - Manual deployment requests
- Stages: Build, test, staging deployment, production deployment
- Safety: Automated rollback on performance degradation

Containerization (Docker)

- Function: Ensures consistent environments
- Components:
  - Separate containers for data processing, model training, and inference
  - Environment-specific configurations
  - Resource optimization
- Benefits: Isolated dependencies and scalable deployment

Kubernetes Orchestration

- Function: Manages container deployment and scaling
- Features:
  - Auto-scaling based on load
  - Self-healing on container failures
  - Resource allocation optimization
  - High availability configuration
- Deployment: Multi-zone for disaster recovery

3.6 User Interface

Web Dashboard

- Function: Primary interface for farmers and agricultural planners
- Features:
  - Visual 21-day forecast with probability indicators
  - Historical accuracy reporting
  - Custom alert thresholds configuration
  - Data visualization tools
- Access: Role-based permissions for different stakeholders

Mobile App

- Function: On-the-go access to forecasts
- Features:
  - Push notifications for rain alerts
  - Offline capability for remote areas
  - Simplified forecast view
  - GPS integration for location-specific forecasts
- Platforms: iOS and Android

API Gateway

- Function: Secure access point for all services
- Features:
  - Authentication and authorization
  - Rate limiting
  - Request validation
  - Response caching
  - Documentation (Swagger/OpenAPI)
- Integration: Enterprise systems and third-party applications

4. Conclusion

This MLOps system design provides a resilient architecture for delivering accurate, timely rainfall predictions for agricultural planning. The system's key strengths include:

1. **Data Resilience**: Multiple data sources and robust validation ensure prediction continuity despite sensor malfunctions
2. **Continuous Improvement**: Automated monitoring, evaluation, and retraining maintain forecast accuracy over time
3. **Scalability**: Containerized deployment with Kubernetes enables handling seasonal demand fluctuations
4. **Accessibility**: Multiple interfaces ensure farmers can access predictions regardless of technical expertise

With this architecture, the smart agriculture startup can provide reliable, hyper-local weather forecasts that significantly improve upon traditional weather services, enabling more efficient resource utilization and improved crop yields.