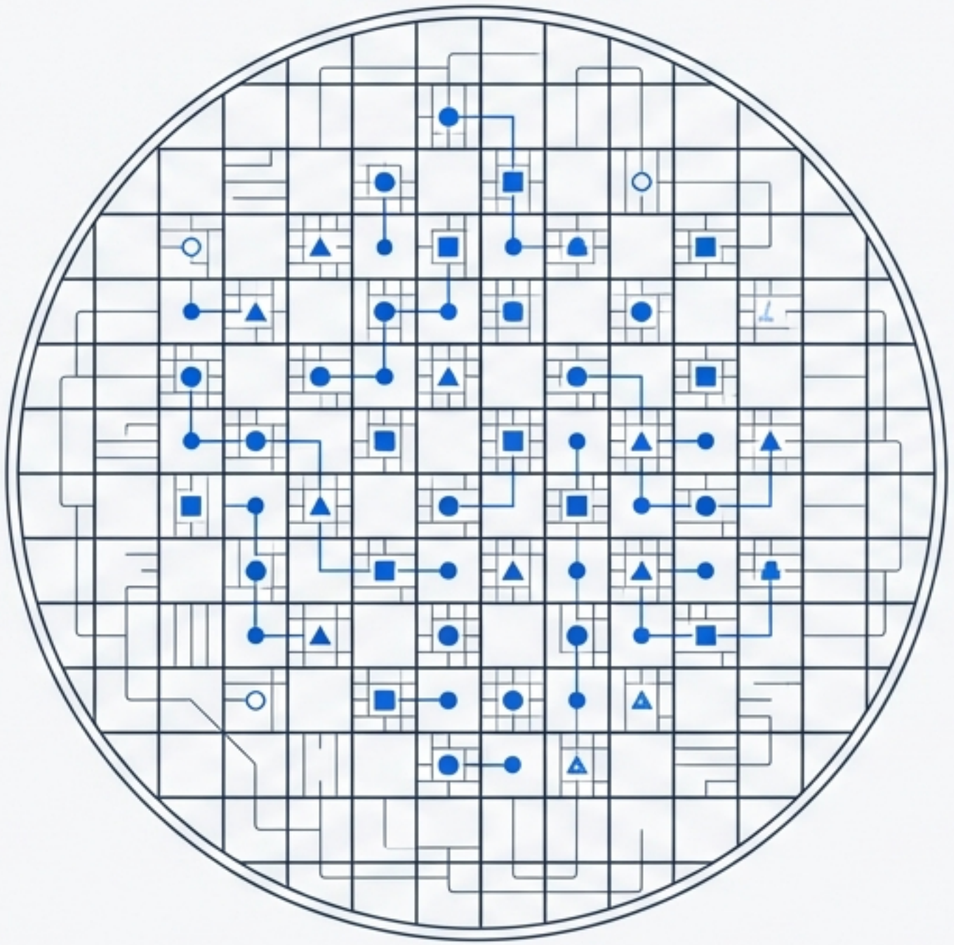


Wafer Defect Analysis System

Wafer Defect Analysis System

Detailed Design Specification | Version 2.0

AUTHOR :: Manus AI
DATE :: January 17, 2026
STATUS :: Final

Living Documentation

All diagrams in this specification are rendered using Mermaid syntax. This ensures the architecture remains version-controlled and easily updatable alongside the codebase.

Engineering Blueprint & Design Principles

This specification translates functional requirements into concrete implementation details for the AI-driven Wafer Defect Analysis System. Designed for a high-volume semiconductor manufacturing environment, the architecture prioritizes three core pillars:

Scalability

Handling fluctuations in wafer throughput without latency spikes.

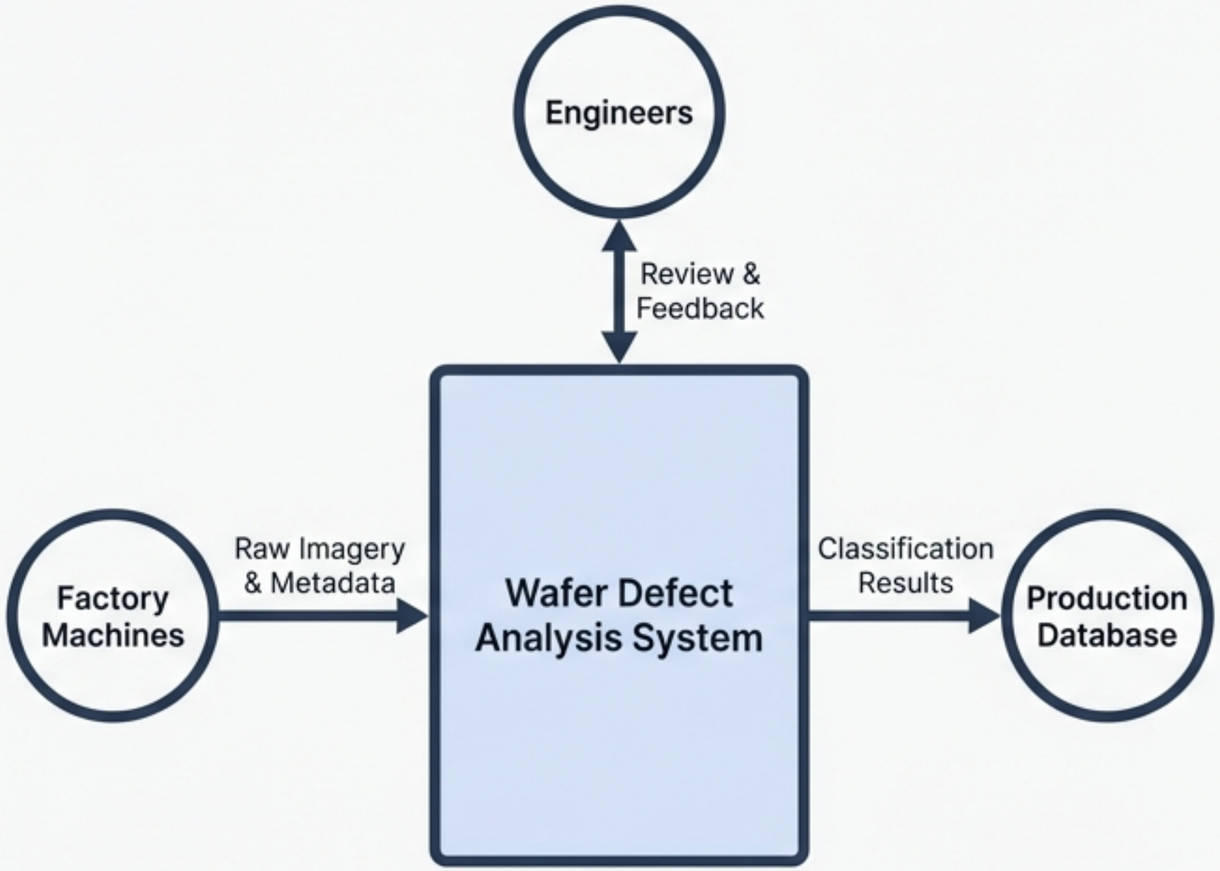
Resilience

Ensuring zero data loss and high availability during component failures.

Maintainability

Decoupling services to allow independent updates and version control.

Mini-Map
Level 0: Context



System Context (Level 0)

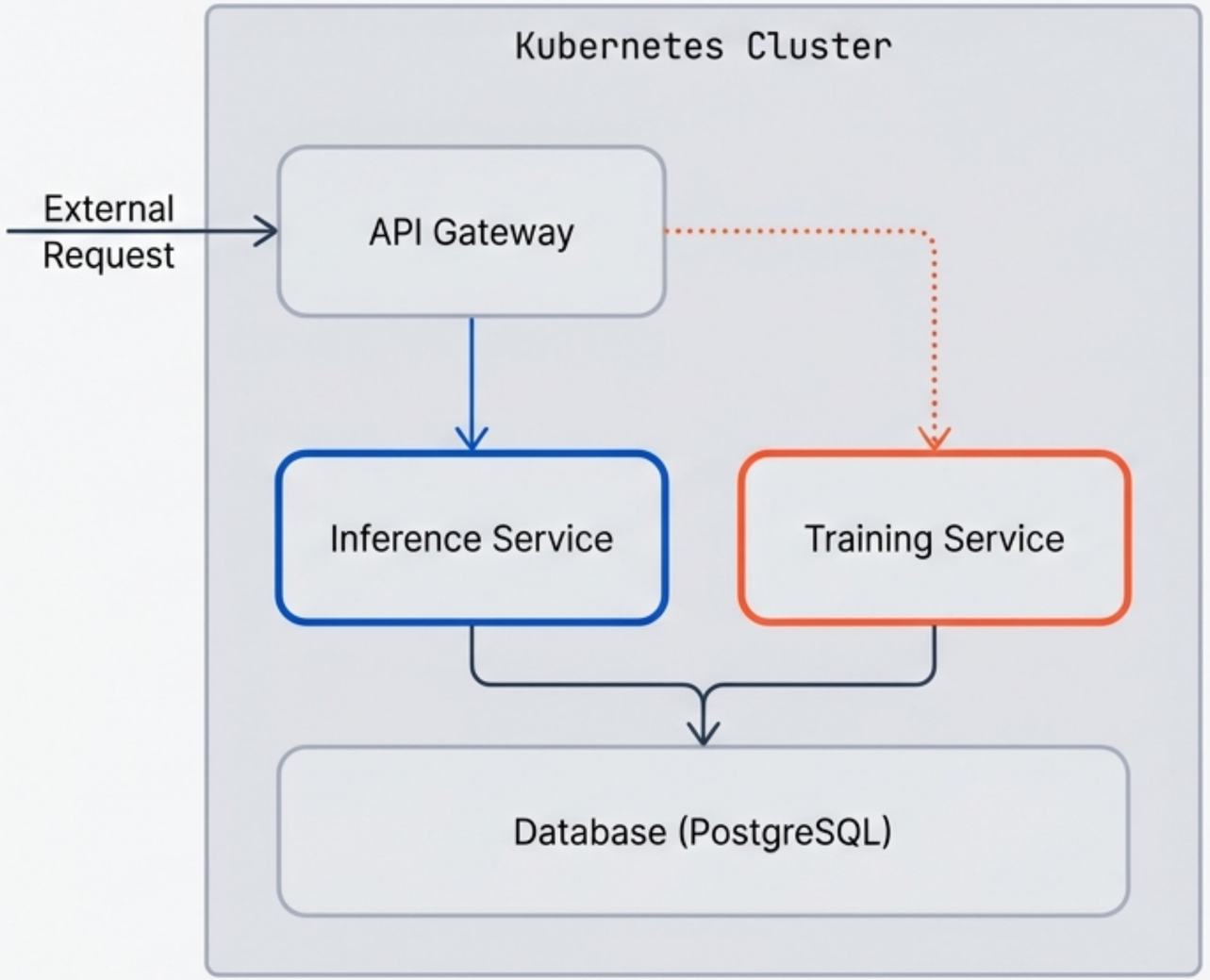
Ref: `D-DFD-01`

System Boundary: The system operates as a centralized analysis hub, abstracting the complexity of inference from the factory floor.

Primary Interfaces:

- **Ingress**: Accepts raw imagery and metadata from Factory Machines.
- **Egress**: Pushes classification results to the downstream Production Database.
- **Interaction**: Provides interfaces for Engineers to review anomalies and provide ground-truth feedback.

Component Architecture Diagram (The "City Map")



Component Architecture (Level 2)

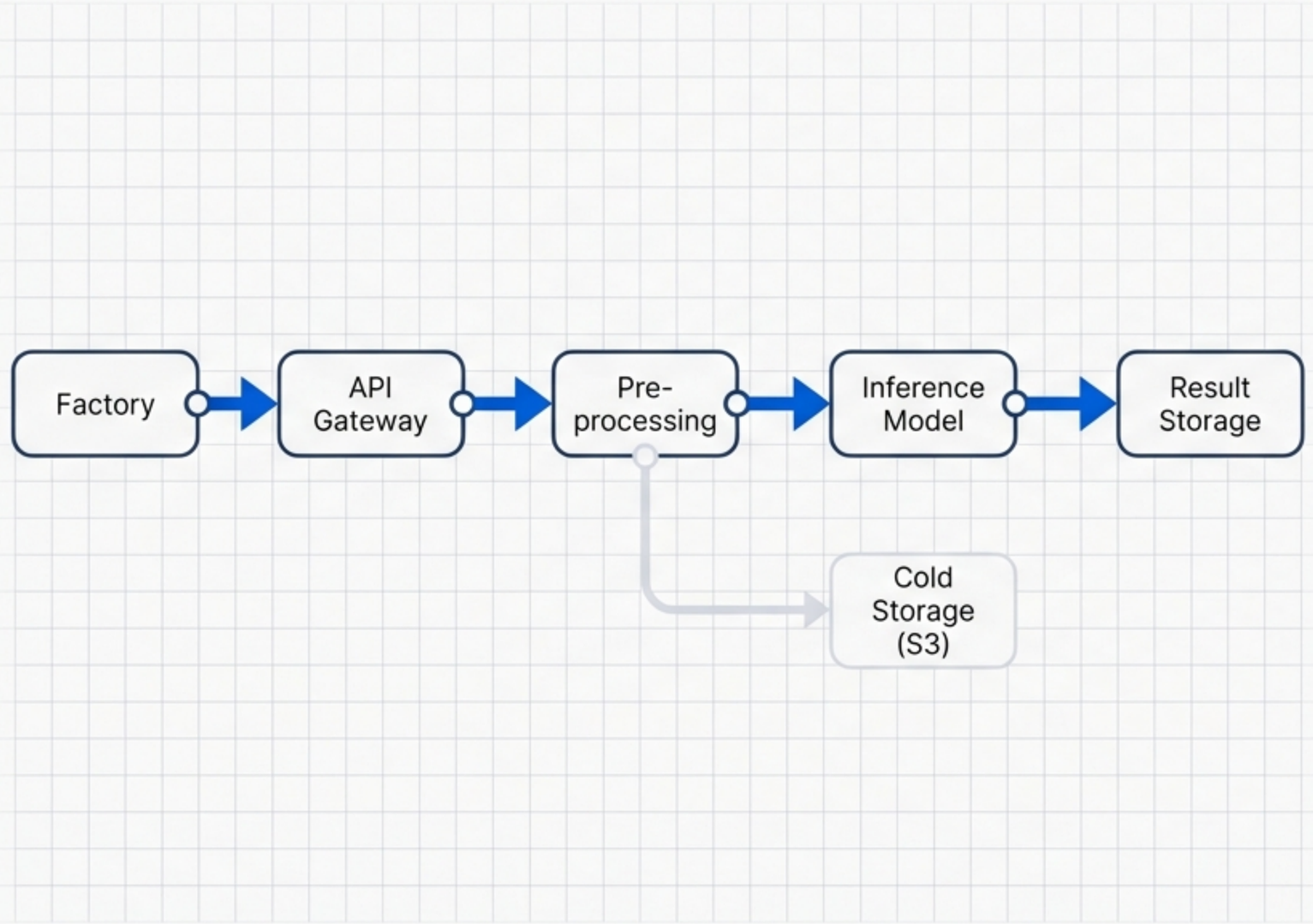
Ref: D-ARCH-01

Microservices Pattern: The system is decomposed into self-contained services, each capable of independent deployment.

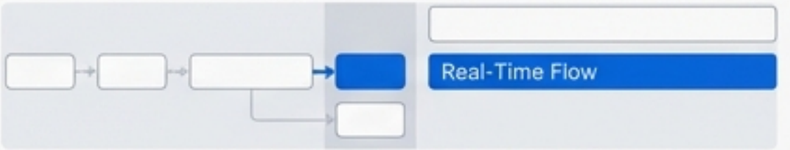
Key Components:

- **API Gateway**: The single entry point routing traffic and enforcing security.
- **Inference Engine**: Dedicated service for executing model predictions (Blue Cluster).
- **Training Module**: Asynchronous service for model retraining (Orange Cluster).

Communication: Services interact via lightweight REST/gRPC APIs, ensuring loose coupling.



Mini-Map



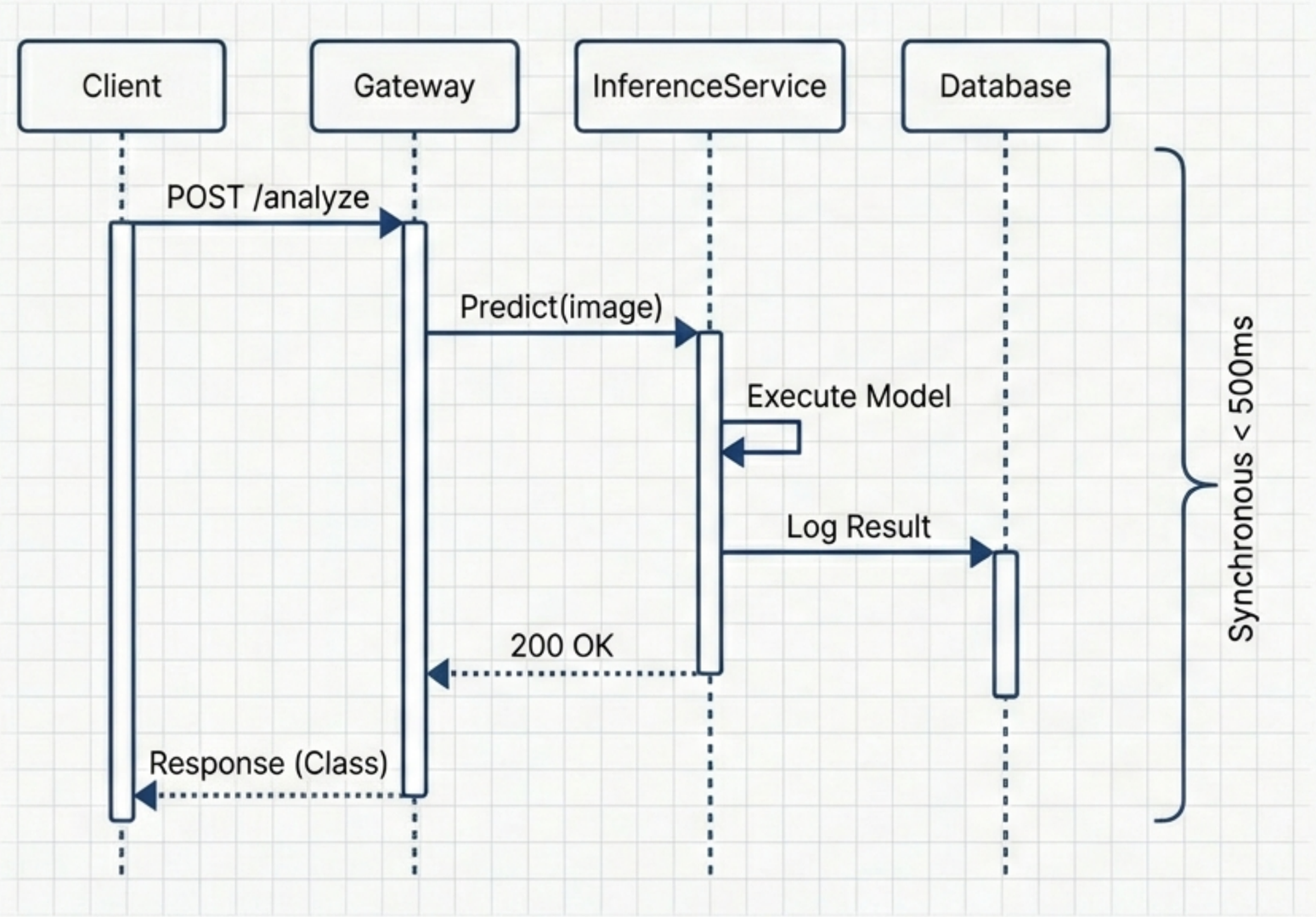
Real-Time Data Flow

Ref: D-DFD-02

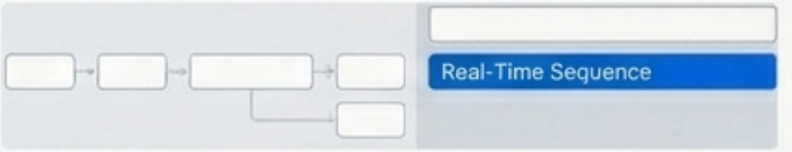
**The Hot Path*: Maximizing throughput for real-time analysis.

Data Lifecycle (Inference):

1. **Ingestion**: Raw data enters via the API Gateway.
2. **Processing**: Images are normalized and batched.
3. **Analysis**: The Inference Service queries the loaded model version.
4. **Persistence**: Results are immediately committed to the operational store, while raw data moves to cold storage for future training.



Mini-Map



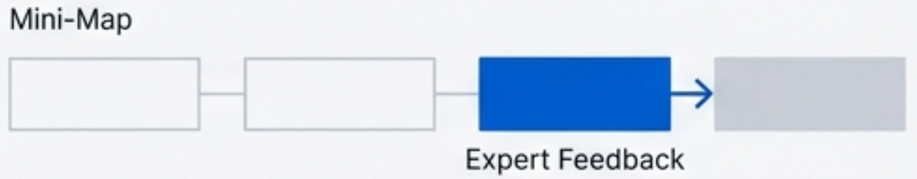
Sequence of Analysis

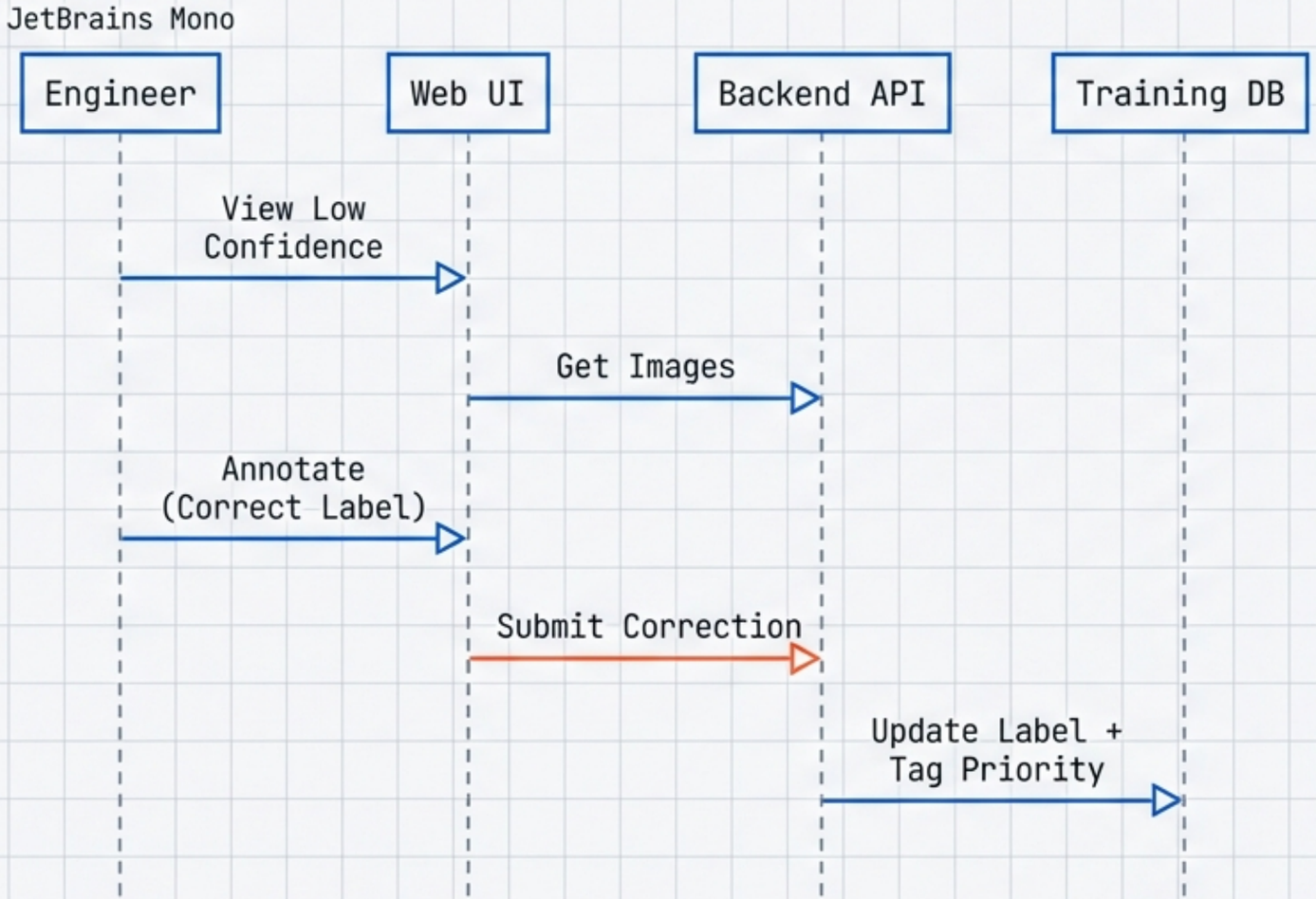
Ref: D-SEQ-01

Synchronous Execution: The analysis pipeline is designed for low-latency response.

Step-by-Step:

- **Request**: Wafer submission triggers an immediate validation check.
- **Query**: The Inference Service requests a prediction from the active model.
- **Logging**: Metadata and results are logged transactionally.
- **Response**: The classification is returned to the factory machine to trigger sorting mechanisms.





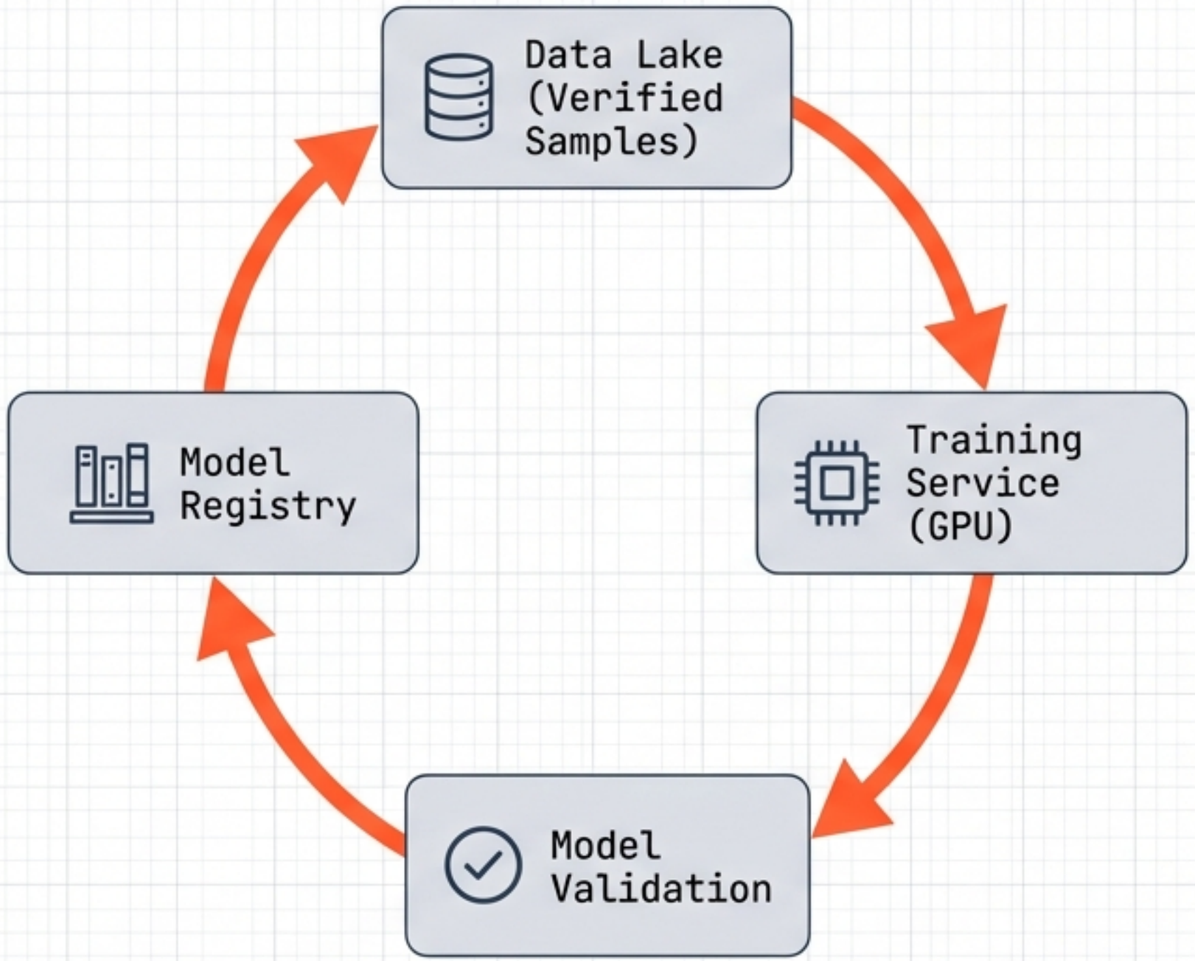
Human-in-the-Loop

Ref: D-SEQ-02

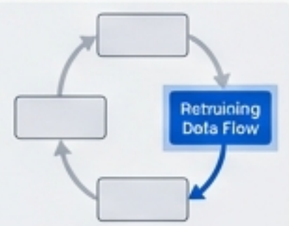
Active Learning: The system improves by leveraging edge cases and low-confidence predictions.

The Feedback Mechanism:

- **Review**: Engineers query specific wafers or review flagged anomalies.
- **Annotation**: Corrections (e.g., re-labeling a 'Scratch' as a 'Crack') are submitted back to the system.
- **Tagging**: These verified samples are tagged as high-priority training data for the next model iteration.



Mini-Map



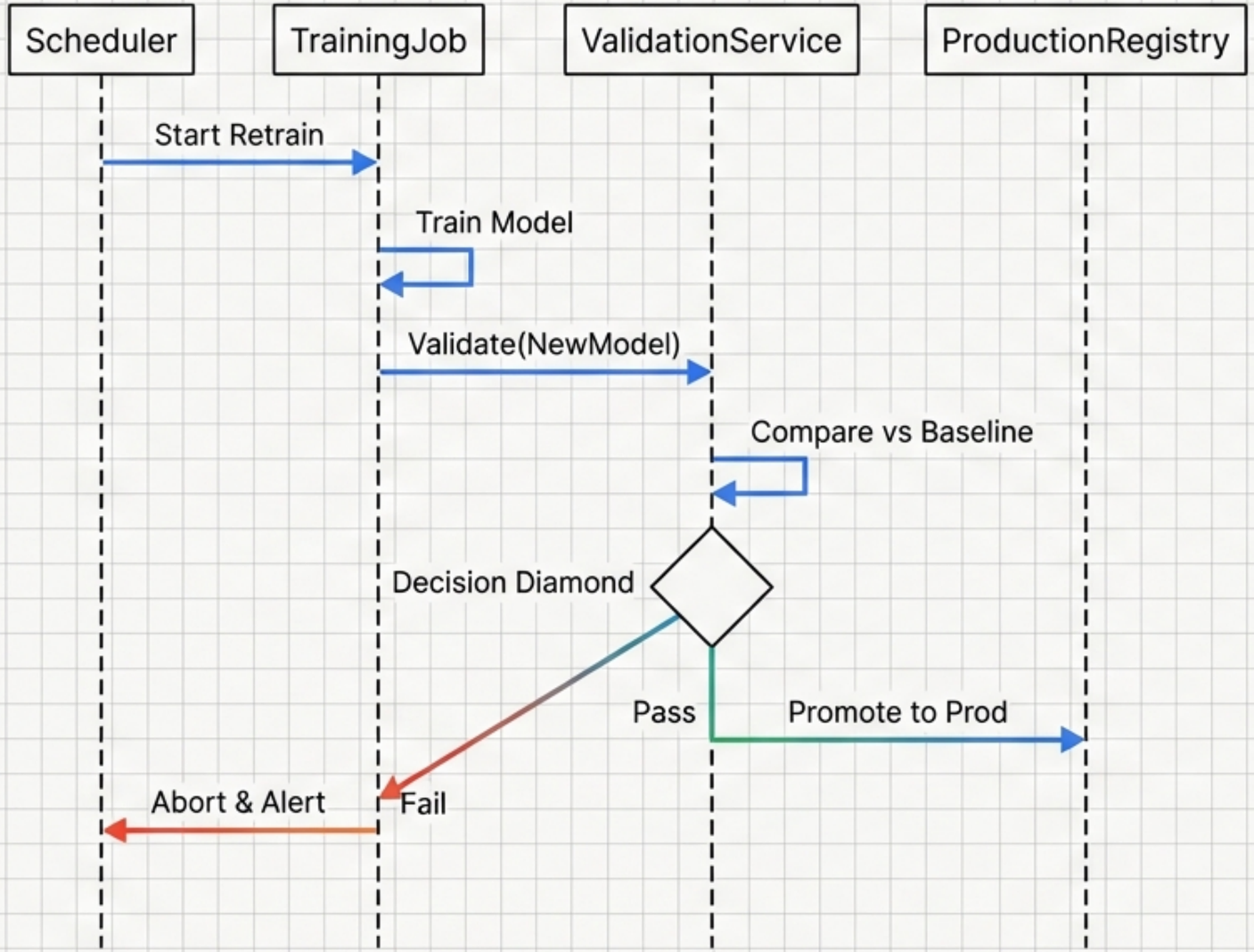
The MLOps Pipeline

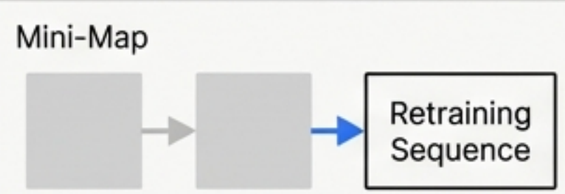
Ref: D-DFD-03

***Automated Evolution** *: Closing the loop between production data and model improvements.

****Pipeline Stages****:

1. ****Extraction****: Aggregating verified feedback and new raw data.
2. ****Training****: The Training Service consumes GPU resources to update model weights.
3. ****Versioning****: New models are versioned and stored in the Model Registry, ready for validation.





Automated Deployment

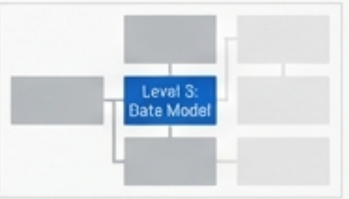
Ref: D-SEQ-03

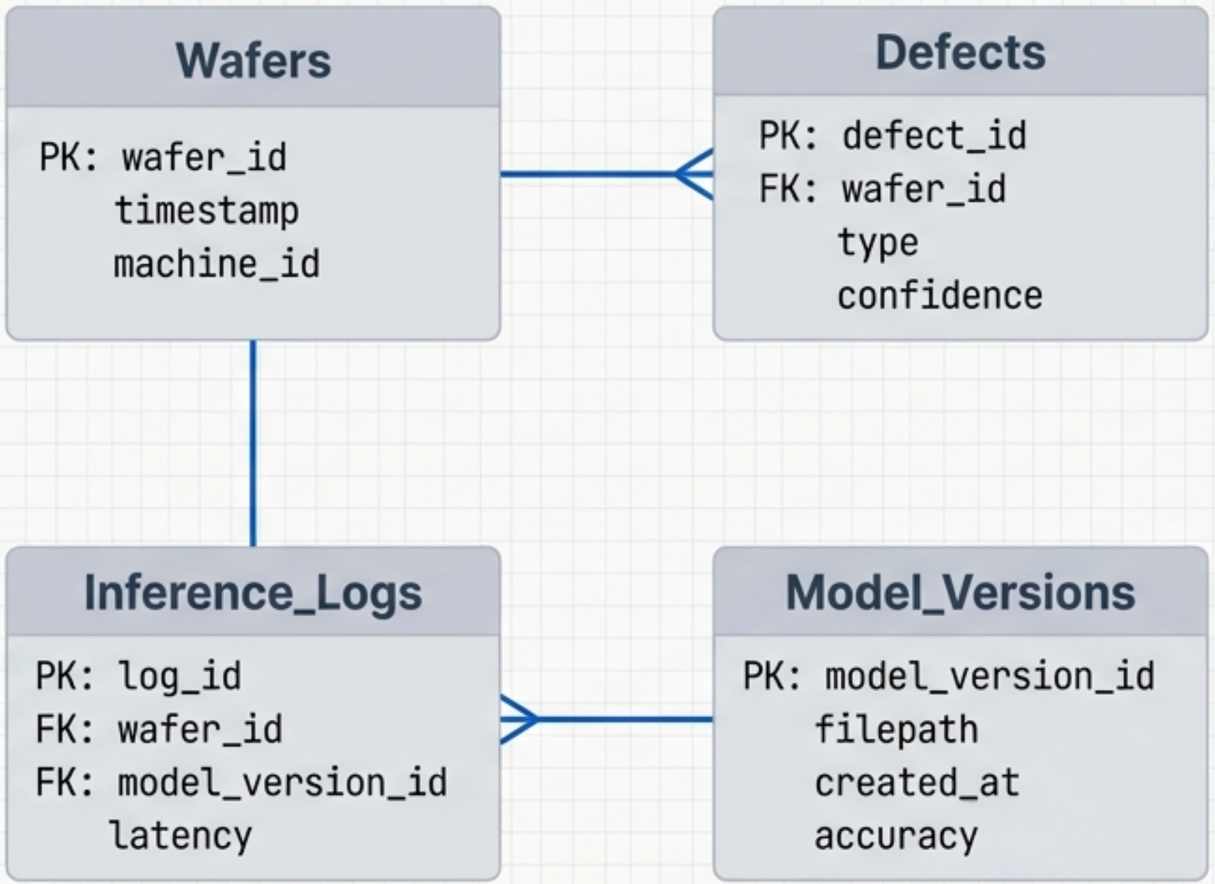
Safety & CI/CD: Ensuring new models meet performance benchmarks before production release.

Validation Gate: The system automatically runs the new model against a 'Gold Standard' test set.

Deployment:
- If performance improves: The model is promoted to the production registry.
- If performance degrades: The roll-out is aborted, and alerts are sent to engineering.

Mini-Map





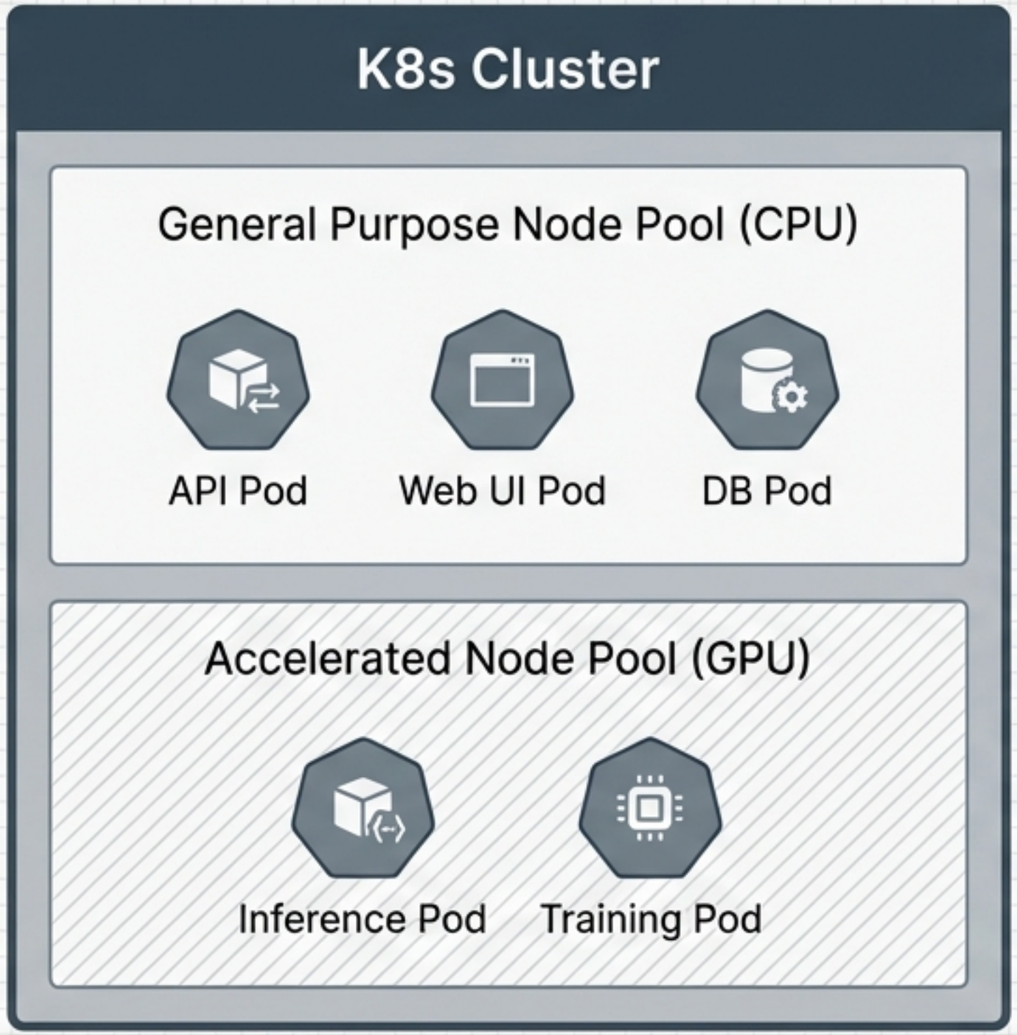
Data Model (ERD)

Ref: D-ERD-01

Schema Design: A normalized PostgreSQL schema ensures data consistency and efficient querying.

Core Relationships:

- ***Wafer <-> Defect***: One-to-many relationship tracking multiple defects per unit.
- ***Inference <-> Model Version***: Strict auditing of which model version generated a specific result.
- ***Feedback <-> Training***: Linking human inputs directly to the datasets used for retraining.



Infrastructure (Kubernetes)

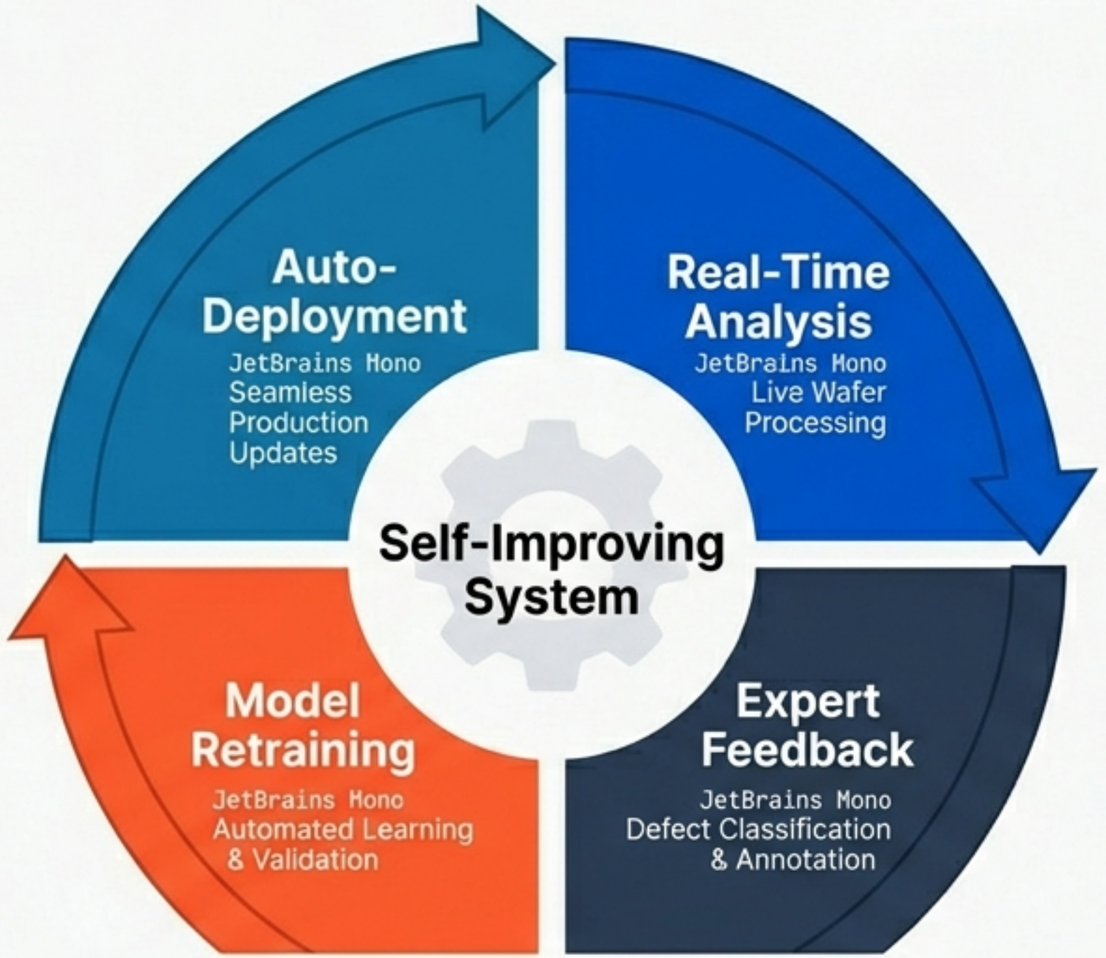
Ref: D-DEPLOY-01

Hybrid Resource Management: Strategic allocation of compute resources.

Node Pools:

- **CPU Nodes**: Host the API Gateway, Web UI, and Database (Postgres) for general-purpose logic.
- **GPU Nodes**: Dedicated expensive hardware reserved strictly for the Inference Engine and Training Service pods to maximize utilization.
- **Scalability**: Pods auto-scale horizontally based on queue depth and CPU/GPU metrics.

Mini-Map
System Overview



Traceability

Every defect is linked to a model version.

Agility

Microservices allow rapid iteration of individual components.

Intelligence

The integrated MLOps pipeline ensures the system becomes smarter with every wafer processed.

A complete architectural foundation for development, validation, and scale.