DA24M016
DA24M017

High-Level Design (HLD) – ALL Classifier

---

1 Purpose

Provide a bird-eye view of the Web-based Acute Lymphoblastic Leukemia (ALL) image-classification system, the major technology blocks, and the interactions among them.

2 Key Components & Responsibilities

Component	Technology	Responsibility
UI	Streamlit	image upload, prediction call, feedback form, progress indication
API	FastAPI	exposes `/predict`, `/feedback`, Prometheus `/metrics`
Model	TensorFlow-Keras CNN	classify WBC images into *Benign/Early/Pre/Pro*
Tracking	MLflow	record params, metrics, artefacts, promote model

Orchestration	Airflow	daily ETL ➜ train ➜ test ➜ register
Monitoring & Ops	Prometheus + Grafana	API latency, error-rate, CPU/mem via Node-Exporter

3 Data Flow (chronological)

1. User uploads image ▸ UI issues `POST /predict`.
2. Backend preprocesses → CNN predicts class → returns JSON.
3. UI renders class & optionally sends feedback via `POST /feedback` (stored in SQLite `/app/data/feedback.db`).
4. Prometheus scrapes `/metrics` (HTTP counters, latency histograms) & Node-Exporter system metrics.
5. Grafana dashboards visualise real-time metrics.
6. Airflow DAG (daily) reads *feedback.db*; if ≥ 100 new rows it appends images & labels to dataset, retrains via MLflow-tracked run, registers new model.

4 Non-Functional Requirements

- **Scalability** – stateless backend, can scale via docker-compose replicas.
- **Portability** – single `docker-compose` starts all services; optional separation into customer-facing (frontend + backend) vs MLOps stack.

- **Observability** – out-of-the-box API & host metrics, alert rules definable in Grafana.

- **Security** – CORS locked to specific domain for production; internal services on private network.

5 Assumptions & Constraints

- GPU not required; TensorFlow CPU build used.

- Training dataset fits in container memory (< 2 GB after preprocessing).

- Feedback retraining threshold set to 100 but is configurable via ENV `FB_THRESH`.