

Client Document: “AI Digital School”

Executive Summary

- Services: Admin Core, School Node, Control Plane, Admin Mobile BFF, AI-Tracker, Proxy (Nginx), Landing.
- Personal data of students/teachers stays only in School Node. Admin Core stores metadata, policies, audit and aggregated analytics.
- Control Plane centralizes module registry, policies and commands. Proxy routes and isolates traffic.

Value

- Platform management: schools, modules, policies, audit, reports.
- Reliability: observability, alerts, metrics, tracing.
- Scalability: multi-tenant, modular, resource profiles.

Architecture (diagram)

```
flowchart LR
    subgraph Internet
        U["User\nAdmin/Teacher/Student"]
    end
    U -->|HTTPS| PX["Proxy Nginx"]
    PX -->|/admin, /api/admin| AC["Admin Core (FastAPI)"]
    PX -->|/school, /api/cabinet| SN["School Node (FastAPI)"]
    PX -->|/api/control| CP["Control Plane (FastAPI)"]
    PX -->|/api/mobile-admin| MB["Admin Mobile BFF (FastAPI)"]
    PX -->|/status/ai| AI["AI-Tracker (FastAPI+AI)"]
    PX -->|/| LD["Landing (Nginx)"]

    AC --- DB["Postgres"]
    SN --- TS["TimescaleDB"]
    AC --- RD["Redis"]
    SN --- RD

    subgraph Observability
        PRM["Prometheus"]
        GFN["Grafana"]
        LKI["Loki/Promtail"]
    end
    AC -- /metrics --> PRM
    SN -- /metrics --> PRM
    CP -- /metrics --> PRM
    MB -- /metrics --> PRM
    AI -- /metrics --> PRM
    PX -- nginx_status --> PRM
    PX == logs ==> LKI
```

Services and Ports

Service	Internal port	Proxy routes	Purpose
Admin Core	9000	/admin, /api/admin/*	Admin UI, metadata, policies
School Node	8100	/school, /api/cabinet/*, 2026	School cabinet, learning data
Control Plane	8000	/api/control/*	Central control
Admin Mobile BFF	8000	/api/mobile-admin/*	Mobile admin backend
AI-Tracker	9000	/status/ai*	Activity/AI events
Proxy	80/443	9300/9443	Reverse proxy, TLS
Landing	80	/	Public static site
Postgres	5432	—	Metadata DB
TimescaleDB	5432	—	Metrics/events DB
Redis	6379	—	Cache/broker

Data Boundaries

- Cabinet Data (School Node): profiles, schedules, progress, module events — strictly within each school.
- Management Data (Admin Core): schools, module registry, policies, statuses, audit, aggregates.
- Cross-data: only identifiers and aggregates in Admin Core; no personal fields from cabinets.

Components

Admin Core

- Purpose: admin operations, metadata, policies, audit, analytics.
- Tech: FastAPI, SQLAlchemy, Redis, PyJWT, Prometheus.
- Ports/routes: 9000 ; via Proxy /admin , /api/admin/* .
- Paths: services/admin-core/ (backend: backend/app/main.py).

School Node (Template)

- Purpose: school cabinet APIs for students/teachers; module integration.
- Tech: FastAPI, SQLAlchemy, Alembic, Redis, Prometheus, OpenCV.
- Ports/routes: 8100 ; Proxy /school , /api/cabinet/* .
- Paths: services/school-node-template/ .

Control Plane

- Purpose: registry, policies/versioning, commands, statuses.
- Tech: FastAPI, Prometheus.
- Ports/routes: 8000 ; Proxy /api/control/* .
- Paths: services/control-plane/ .

Admin Mobile BFF

- Purpose: simplified endpoints for mobile admin.
- Tech: FastAPI, Prometheus.
- Ports/routes: 8000 ; Proxy /api/mobile-admin/* .

- Paths: `services/admin-mobile-bff/` .

AI-Tracker

- Purpose: video/image processing, activity events; integration with School Node.
- Tech: FastAPI, OpenCV, Prometheus.
- Ports/routes: `9000` ; Proxy `/status/ai` , `/status/ai-ready` .
- Paths: `services/ai-tracker/` .

Proxy (Nginx)

- Purpose: routing, TLS, isolation, rate limiting.
- Ports: `9300` (HTTP), `9443` (HTTPS), dedicated UI/API ports.
- Config: `services/proxy/nginx.conf` .

Landing

- Purpose: public static site.
- Port: `80` ; proxied to `/` .
- Paths: `services/landing/` .

API Overview (samples)

- Admin Core: auth, users, schools, modules, policies, audit, `/health` , `/metrics` .
- School Node: auth (OTP/JWT), cabinet, students, activities, camera/AI, `/health` , `/metrics` .
- Control Plane: `/v1/registry/*` , `/v1/control/*` , `/v1/status/*` , `/metrics` .
- Admin Mobile BFF: `/v1/mobile/*` , `/metrics` .
- AI-Tracker: `/status` , `/control/*` , `/camera/*` , `/metrics` .

Security

- Authentication: JWT (HS256), roles/policies (RBAC), admin audit.
- Isolation: routing by domain/path, cabinet data segmentation within School Node.
- Nginx: HTTPS/TLS, security headers, rate limiting for `/api` and `/auth` .

Observability

- Metrics: `GET /metrics` in all services, Prometheus scraping.
- Logs: Loki/Promtail collect backend/nginx/system/docker logs.
- Dashboards: Grafana for performance, error rates, resource consumption.

Deployment

- Orchestration: `docker-compose.yml` , `docker-compose.ssl.yml` .
- Networks/profiles: `ai-school-net` , profiles `db` , `celery` , `ssl` .
- Environment: global `.env` and local service `.env.s`.

Flow Diagram (example operation)

```
sequenceDiagram
    participant Admin
    participant AdminCore
    participant ControlPlane
    participant SchoolNode
```

```
participant Module
Admin->>>AdminCore: Enable module {school_id, module_key}
AdminCore->>>ControlPlane: Command issue
ControlPlane->>>SchoolNode: Apply module config
SchoolNode->>>Module: Start/Configure
Module-->>>SchoolNode: Status OK
SchoolNode-->>>ControlPlane: Module status
ControlPlane-->>>AdminCore: Ack + status
AdminCore-->>>Admin: Operation success
```

Risks & Mitigations

- Data leaks: strict cabinet isolation, audits, Nginx security.
- Load spikes: scale School Node, queues, caching, resource profiling.
- AI modules complexity: resource caps, CPU fallback, GPU profiles.

Roadmap

- Phase 1: API stabilization, full Swagger/Redoc.
- Phase 2: expand activity modules and mobile features.
- Phase 3: richer reporting and school auto-provisioning.

Appendices

A. Key File Paths

- `docker-compose.yml` , `docker-compose.ssl.yml`
- `services/proxy/nginx.conf`
- `services/admin-core/backend/app/main.py`
- `services/school-node-template/backend/app/main.py`
- `services/control-plane/backend/app/main.py`
- `services/admin-mobile-bff/backend/app/main.py`
- `services/ai-tracker/app/main.py`

B. Environment Variables (overview)

Variable	Purpose
ADMIN_CORE_DB_URL	Admin Core DB connection
SCHOOL_NODE_DB_URL	School Node DB connection
JWT_SECRET/ADMIN_CORE_JWT_SECRET	token secret
REDIS_URL	broker/cache
CORS_ORIGINS	CORS domains
PROMETHEUS_SCRAPE_INTERVAL	metrics scrape interval

C. Proxy Routes (overview)

Path	Upstream
------	----------

/admin/, /api/admin/*	admin-core:8000
/school/, /api/cabinet/*	school-template:8100
(not in current configuration)	—
/status/ai*	ai-tracker:8001
/	landing:8080

D. Glossary

- School: isolated instance with its own School Node.
- Module: attachable service (e.g., AI-Tracker) to a school.
- Policy: access rules and settings distributed by Control Plane.
- Audit: log of administrative operations.
- Aggregate: metrics without personal data.