

Клиентский документ: «AI Цифровая Школа»

Резюме

- Система разделена на сервисы: Admin Core, School Node, Control Plane, Admin Mobile BFF, AI-Tracker, Proxy (Nginx), Landing.
- Персональные данные учеников/учителей хранятся только в School Node. В Admin Core — метаданные, политики, аудит и агрегированная аналитика.
- Control Plane централизует реестр модулей, политики и команды. Proxy маршрутизирует и изолирует трафик.

Ценность для клиента

- Управление платформой: школы, модули, политики, аудит, отчёты.
- Надёжность: наблюдаемость, алерты, метрики, трассировка.
- Масштабируемость: многотенантность, модульная архитектура, профили нагрузки.

Архитектура (схема)

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

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

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

Таблица сервисов и портов

Сервис	Внутренний порт	Прoxy маршруты	Назначение
Admin Core	9000	/admin, /api/admin/*	Админка, метаданные, политики
School Node	8100	/school, /api/cabinet/*, 2026	Кабинет школы, учебные данные
Control Plane	8000	/api/control/*	Управление школами/модулями
Admin Mobile BFF	8000	/api/mobile-admin/*	Мобильная админка
AI-Tracker	9000	/status/ai*	Активности/AI события
Proxy	80/443	9300/9443	Реверс-прокси, SSL
Landing	80	/	Публичная статика
Postgres	5432	—	БД метаданных
TimescaleDB	5432	—	БД метрик/событий
Redis	6379	—	Кэш/брокер

Границы данных и приватность

- Cabinet Data (в School Node): профили, расписания, прогресс, события модулей — персональные данные строго внутри школы.
- Management Data (в Admin Core): школы, реестр модулей, политики, статусы, аудит, агрегаты.
- Кросс-данные: только идентификаторы и агрегированные метрики в Admin Core; никаких личных полей из кабинетов.

Компоненты

Admin Core

- Назначение: веб-админка и административные операции.
- Основное: пользователи, школы, реестр модулей, роли/политики, аудит, отчёты, метрики.
- Технологии: FastAPI, SQLAlchemy, Redis, PyJWT, Prometheus.
- Порты/маршруты: 9000 ; через Proxy /admin , /api/admin/* .
- Пути: services/admin-core/ (backend: backend/app/main.py , конфиг: backend/app/core/config.py).

School Node (Template)

- Назначение: API внутри школы для учеников/учителей, интеграция с модулями.
- Основное: аутентификация (OTP/JWT), студенты/кабинет, расписания, прогресс, AI/камеры, метрики.
- Технологии: FastAPI, SQLAlchemy, Alembic, Redis, Prometheus, OpenCV.
- Порты/маршруты: 8100 ; Proxy /school , /api/cabinet/* .
- Пути: services/school-node-template/ (backend: backend/app/main.py).

Control Plane

- Назначение: реестр и команды управления для школ/модулей.
- Основное: `registry` , `control` , `status` , метрики.
- Технологии: FastAPI, Prometheus.
- Порты/маршруты: `8000` ; Proxy `/api/control/*` .
- Пути: `services/control-plane/` .

Admin Mobile BFF

- Назначение: упрощённые эндпоинты под мобильную админку.
- Основное: мобильная панель, алерты, прокси-команды.
- Технологии: FastAPI, Prometheus.
- Порты/маршруты: `8000` ; Proxy `/api/mobile-admin/*` .
- Пути: `services/admin-mobile-bff/` .

AI-Tracker

- Назначение: специализированные сервисы активности (видео/изображения), интеграция со School Node.
- Основное: статус/метрики, управление, камеры, детекция.
- Технологии: FastAPI, OpenCV, Prometheus.
- Порты/маршруты: `9000` ; Proxy `/status/ai` , `/status/ai-ready` .
- Пути: `services/ai-tracker/` .

Proxy (Nginx)

- Назначение: маршрутизация, SSL, изоляция, rate limiting.
- Порты: `9300` (HTTP), `9443` (HTTPS), выделенные порты для UI/API.
- Конфиг: `services/proxy/nginx.conf` .

Landing

- Назначение: публичная статика.
- Порт: `80` (внутренний), проксируется на `/` .
- Пути: `services/landing/` .

API (обзор)

Admin Core (пример)

- Аутентификация: `POST /v1/auth/login` , `GET /v1/auth/me` , `POST /v1/auth/logout` .
- Пользователи: `GET/POST /v1/users` , `GET/PUT/DELETE /v1/users/{id}` .
- Школы: `GET/POST /v1/schools` , `GET/PUT/DELETE /v1/schools/{id}` , `POST /v1/schools/{id}/heartbeat` .
- Модули: `GET/POST /v1/modules` , управление установками/версиями.
- Политики/аудит: `GET/POST /v1/policies` , `GET /v1/audit` .
- Системные: `GET /health` , `GET /metrics` .

School Node (пример)

- Аутентификация: `POST /v1/auth/request_otp` , `POST /v1/auth/verify_otp` , `GET /v1/auth/me` .
- Кабинет: `GET /v1/cabinet/profile` , `GET /v1/cabinet/schedule` .
- Студенты/активности: `GET/POST /v1/students` , `POST /v1/students/{id}/activity` .
- Камеры/AI: `GET /v1/camera/status` , `POST /v1/camera/start` , `POST /v1/ai/detect` .

- Системные: `GET /health` , `GET /metrics` .

Control Plane / Admin Mobile BFF / AI-Tracker

- Control Plane: `GET/POST /v1/registry/modules|schools` , `POST /v1/control/schools/{school_id}/modules/{module_key}/start|stop` , `GET /metrics` .
- Admin Mobile BFF: `GET /v1/mobile/dashboard` , `GET /v1/mobile/alerts` , `POST /v1/mobile/control/*` , `GET /metrics` .
- AI-Tracker: `GET /status|/health|/metrics` , `POST /control/start|stop` , `GET /camera/*` .

Безопасность

- Аутентификация: JWT (HS256), роли/политики (RBAC), аудит админ-операций.
- Изоляция данных: маршрутизация по доменам/путям, сегментация cabinet-данных в School Node.
- Nginx: HTTPS/TLS, security headers, rate limiting для `/api` и `/auth` .

Наблюдаемость

- Метрики: `GET /metrics` во всех сервисах, сбор Prometheus.
- Логи: Loki/Promtail собирают backend/nginx/system/docker.
- Дашборды: Grafana для производительности, ошибок, потребления ресурсов.

Развёртывание

- Оркестрация: `docker-compose.yml` , `docker-compose.ssl.yml` .
- Сети/профили: `ai-school-net` , профили `db` , `celery` , `ssl` .
- Окружение: `.env` глобальный и локальные `.env` в сервисах.

Диаграмма потока (пример операции)

```
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
```

Риски и меры

- Утечки данных: строгая изоляция cabinet-данных, аудиты, безопасность Nginx.
- Нагрузки: масштабирование School Node, очереди, кэширование, профили ресурсов.
- Модули AI: управление ресурсами, fallback на CPU, профили GPU.

Дорожная карта

- Этап 1: стабилизация API, полная документация Swagger/Redoc.
- Этап 2: развитие модулей активности и мобильной админки.
- Этап 3: расширение отчётности и автопроежирования школ.

Приложения

А. Пути к ключевым файлам

- `docker-compose.yml` , `docker-compose.ssl.yml`
- `services/proxy/nginx.conf`
- `services/admin-core/backend/app/main.py` , `backend/app/core/config.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`

В. Переменные окружения (обзор)

Переменная	Назначение
ADMIN_CORE_DB_URL	строка подключения Admin Core к БД
SCHOOL_NODE_DB_URL	строка подключения School Node
JWT_SECRET/ADMIN_CORE_JWT_SECRET	секрет для токенов
REDIS_URL	брокер/кэш
CORS_ORIGINS	домены для CORS
PROMETHEUS_SCRAPY_INTERVAL	период опроса метрик

С. Маршруты Proху (обзор)

Путь	Апстрим
<code>/admin/</code> , <code>/api/admin/*</code>	<code>admin-core:8000</code>
<code>/school/</code> , <code>/api/cabinet/*</code>	<code>school-template:8100</code>
(нет в текущей конфигурации)	—
<code>/status/ai*</code>	<code>ai-tracker:8001</code>
<code>/</code>	<code>landing:8080</code>

Д. CLO/SLA и алертинг (обзор)

- Доступность: 99.5% для Admin Core и School Node.
- Время ответа p95: < 400мс.
- Ошибки: < 1%.

Е. Глоссарий

- Школа: изолированный инстанс с собственным School Node.
- Модуль: подключаемый сервис (например, AI-Tracker) к школе.
- Политика: правила прав доступа и настройки, распространяемые Control Plane.
- Аудит: журнал административных действий.
- Агрегат: метрики без персональных данных.

Г. Детализация эндпоинтов и файлов

Сервис	Эндпоинт (пример)	Файл/директория
Admin Core	/v1/auth/login	services/admin-core/backend/app/main.py, backend/app/api/v1/auth.py
Admin Core	/v1/schools	services/admin-core/backend/app/api/v1/schools.py
Admin Core	/metrics	services/admin-core/backend/app/main.py
School Node	/v1/auth/request_otp	services/school-node-template/backend/app/api/v1/auth_api.py
School Node	/v1/cabinet/profile	services/school-node-template/backend/app/api/v1/diary_api.py
School Node	/v1/camera/start	services/school-node- template/backend/app/api/v1/functional_api.py
Control Plane	/v1/registry/modules	services/control-plane/backend/app/main.py
Admin Mobile BFF	/v1/mobile/dashboard	services/admin-mobile-bff/backend/app/main.py
AI-Tracker	/control/start	services/ai-tracker/app/main.py

Примечание: точные пути роутеров могут отличаться по файлам *_api.py в backend/app/api/v1/ (School Node) и соответствующих модулях в Admin Core; в документе зафиксированы ключевые точки входа.