Архитектура системы ONLANTA AI HUB

ONLANTA AI HUB построен на микросервисной архитектуре с использованием Docker контейнеров. Система состоит из 10 основных сервисов, объединенных в единую сеть.

Основные компоненты:

- **Backend API** - Основной FastAPI сервер

- **Frontend** - React веб-приложение

- **RAG Service** - Сервис обработки документов

- **Transcription Service** - Сервис транскрибации аудио/видео

- **OCR Service** - Сервис распознавания текста из изображений

- **Celery Workers** - Асинхронная обработка задач

- **PostgreSQL** - Основная реляционная база данных

- **Redis** - Кэш и брокер сообщений

- **Neo4j** - Графовая база данных для RAG

- **Nginx** - Обратный прокси и SSL терминация

Backend сервисы

1. **Backend API (Основной сервер)**

Язык: Python

Фреймворк:

- FastAPI - асинхронный веб-фреймворк

- Uvicorn - ASGI сервер

- Gunicorn - WSGI сервер для продакшена

База данных и ORM:

- SQLAlchemy - ORM для работы с PostgreSQL

- Alembic - Система миграций

- Psycopg2 - PostgreSQL адаптер

Аутентификация и безопасность:

- python-jose - JWT токены

- passlib с bcrypt - Хэширование паролей

- bcrypt - Криптографическая библиотека

Кэширование:

- Redis - In-memory кэш и брокер задач

- Hiredis - Высокопроизводительный парсер Redis протокола

Обработка документов:

- python-docx - Работа с Word документами

- python-pptx - Работа с PowerPoint

- openpyxl - Работа с Excel

- pdfplumber - Извлечение данных из PDF

- pypdf - Работа с PDF

- PyMuPDF - Быстрая обработка PDF

- Docling – Извлечение данных из документов

OCR и компьютерное зрение:

- Pillow - Обработка изображений

- pytesseract - OCR движок

- opencv-python-headless - Компьютерное зрение

Аудио/Видео обработка:

- transformers - Hugging Face трансформеры

- peft - Parameter-Efficient Fine-Tuning

- ffmpeg-python - Работа с медиа файлами

- soundfile - Чтение/запись аудио

- librosa - Анализ аудио

- nemo_toolkit[asr] - NVIDIA NeMo для ASR

RAG и векторные базы:

- llama-index-core - Фреймворк для RAG

- llama-index-llms-ollama - LLM интеграция

- llama-index-embeddings-ollama - Embeddings через Ollama

- llama-index-vector-stores-chroma - ChromaDB интеграция

- chromadb - Векторная база данных

- sentence-transformers - Модели для embeddings

LangChain и AI агенты:

- langgraph - Граф-агенты

- langchain - Фреймворк для LLM приложений

- langchain-community - Дополнительные интеграции

- langchain-core - Ядро LangChain

Графовые базы данных:

- neo4j - Драйвер для Neo4j

Обработка текста:

- nltk - Natural Language Toolkit

- beautifulsoup4 - Парсинг HTML

- lxml - Быстрый XML/HTML парсер

Поиск и веб-скрейпинг:

- ddgs - DuckDuckGo поиск

- wikipedia - API для Википедии

Асинхронная обработка:

- Celery - Распределенная очередь задач

- Kombu - Библиотека для обмена сообщениями

HTTP клиенты:

- httpx - Асинхронный HTTP клиент

- aiohttp - Асинхронный HTTP клиент/сервер

Научные вычисления:

- numpy - Числовые вычисления
- scipy - Научные вычисления
- matplotlib - Визуализация данных
- networkx - Анализ графов

Мониторинг и логирование:

- prometheus-client - Метрики для Prometheus
- structlog - Структурированное логирование

Утилиты:

- pydantic - Валидация данных
- pydantic-settings - Настройки приложения
- python-dotenv - Загрузка переменных окружения
- python-magic - Определение типов файлов
- aiofiles - Асинхронная работа с файлами
- omegaconf - Управление конфигурацией
- reportlab - Генерация PDF

Разработка и тестирование:

- pytest - Фреймворк для тестирования
- pytest-asyncio - Тестирование асинхронного кода
- black - Форматирование кода
- flake8 - Линтер

2. RAG Service

Назначение: Продвинутый RAG с использованием графовой базы данных для сложных запросов

Технологии:

- FastAPI + Uvicorn
- Neo4j - Графовая база данных
- HTTPx - Интеграция с Ollama

- Pydantic - Валидация данных

Особенности:

- NER (Named Entity Recognition) через DeepPavlov
- Relation Extraction через Babelscape/mREBEL-large
- Reranking через Qwen3-Reranker-4B
- Embeddings через Qwen3-Embedding-4B
- Граф знаний в Neo4j

3. Transcription Service (Транскрибация аудио/видео)

Назначение: Преобразование аудио и видео в текст с диаризацией

Технологии:

- FastAPI + Uvicorn
- PyTorch + TorchAudio
- Transformers - Whisper модели
- Accelerate - Оптимизация GPU
- NeMo Toolkit - NVIDIA ASR модели
- Librosa - Аудио анализ
- FFmpeg-Python - Конвертация форматов

4. OCR Service (Распознавание текста)

Назначение: Извлечение текста из изображений и PDF с помощью OCR

Технологии:

- FastAPI + Uvicorn
- Docling - Продвинутая обработка документов
- Tesseract - OCR движок (rus+eng)
- tesserocr - Python биндинги для Tesseract
- pytesseract - Альтернативный интерфейс
- PyMuPDF - Обработка PDF
- pdf2image - Конвертация PDF в изображения
- Pillow - Обработка изображений
- PyTorch - Для ML моделей

5. Celery Workers

Назначение: Асинхронная обработка тяжелых задач

Компоненты:

- backend-celery - Основной worker для протоколов и RAG

- backend-queue - Worker для транскрибации

Технологии:

- Celery

- Redis - Брокер и бэкенд результатов

Frontend

Фреймворк: React

Основные библиотеки:

- react-dom - Рендеринг React

- react-router-dom - Маршрутизация

- react-scripts - Create React App

Стилизация:

- styled-components - CSS-in-JS

- tailwindcss - Utility-first CSS

- postcss - CSS трансформации

- autoprefixer - Автопрефиксы CSS

UI компоненты:

- lucide-react - Иконки

- react-dropzone - Drag & Drop загрузка файлов

- react-toastify - Уведомления

- react-beautiful-dnd - Drag & Drop интерфейс

- recharts - Графики и визуализация

Редакторы и markdown:

- react-quill - Rich text редактор

- react-markdown - Рендеринг markdown

- remark-gfm - GitHub Flavored Markdown

HTTP клиент:

- axios - API запросы

Базы данных

1. PostgreSQL 15

Назначение: Основное хранилище данных

2. Redis 7 Alpine

Назначение: Кэш, сессии, брокер сообщений

3. Neo4j

Назначение: Графовая база данных для RAG

4. ChromaDB

Назначение: Векторная база данных для semantic search

Инфраструктура

Docker

NVIDIA GPU

Nginx (Reverse Proxy)