

Описание технической архитектуры
программного обеспечения
«Интеллектуальная корпоративная
система интеграционного менеджмента
информации "ИКСИМИ:ДОКУМЕНТЫ 5.0"»

Наименование программного обеспечения: «Интеллектуальная корпоративная система интеграционного менеджмента информации "ИКСИМИ:ДОКУМЕНТЫ 5.0"»

Данный документ описывает базовые принципы и подходы к реализации различных компонентов ПО «Интеллектуальная корпоративная система интеграционного менеджмента информации "ИКСИМИ:ДОКУМЕНТЫ 5.0"»

1. Общая характеристика архитектуры

Программное обеспечение Hybrid RAG Knowledge Base представляет собой модульную клиент-серверную информационную систему, реализованную на основе трёхуровневой архитектуры (Presentation Layer - Application Layer - Data Layer). Архитектура построена с использованием принципов модульного проектирования и ориентирована на простоту сопровождения, прозрачность обработки данных и возможность быстрого развёртывания как в локальной, так и в защищённой корпоративной инфраструктуре.

Основной функциональной особенностью системы является реализация архитектурного подхода Hybrid Retrieval-Augmented Generation (Hybrid RAG), сочетающего методы семантического поиска по векторным представлениям документов и поиска по ключевым словам с последующей генерацией ответа языковой моделью.

При проектировании системы были приняты следующие архитектурные решения:

- реализация собственного пайплайна обработки данных на базе низкоуровневого AI ML стека PyTorch и Transformers;
- модульная организация компонентов с минимальным количеством промежуточных абстракций;
- использование гибридного механизма поиска (векторный поиск, поиск по ключевым словам и фильтрация по метаданным);
- поддержка работы в локальном контуре организации без передачи документов во внешние сервисы;
- возможность использования локальных языковых моделей.

2. Структурные элементы системы и их интерфейсы

2.1. Уровень представления (Presentation Layer)

Уровень представления реализует взаимодействие пользователя с системой посредством веб-интерфейса.

Основной компонент

Web User Interface (Main UI)

Основные функции

- интерфейс общения с AI-ассистентом;
- потоковое отображение генерируемых ответов;
- просмотр документов PDF с подсветкой найденных фрагментов по координатам;
- управление базами знаний;
- загрузка и удаление документов;
- администрирование пользователей;

Интерфейсы взаимодействия

- HTTP/REST API;
- WebSocket (для потоковой передачи ответов языковой модели).

Стек реализации

- HTML
- JS
- CSS

2.2. Уровень приложения (Application Layer)

Уровень приложения реализует основную бизнес-логику системы.

Основной компонент

Python Backend

В состав Backend входят следующие функциональные модули:

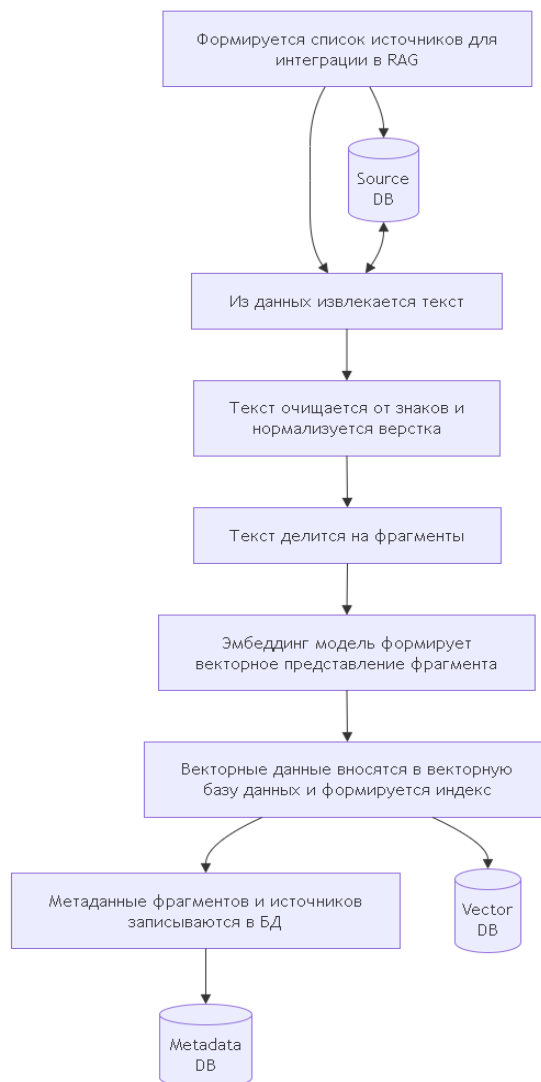
Пайплайн наполнения баз

Назначение:

- импорт документов;
- извлечение текста;
- фрагментация текста;
- векторизация ;
- формирование векторного индекса.

Используемые библиотеки:

- pdfplumber;
- spaCy;
- sentence-transformers.



Гибридный поиск и извлечение

Назначение:

- векторный поиск;
- поиск по ключевым словам;
- фильтрация результатов по метаданным;
- объединение и ранжирование результатов поиска.

Используемые технологии:

- FAISS;
- cosine similarity;
- TF-IDF (scikit-learn).

Предобработка запроса

Назначение:

- предварительная обработка пользовательского запроса;
- реформулирование запроса;
- нормализация поискового выражения.

Генерация

Назначение:

- построение итогового промпта;
- вызов языковой модели;
- генерация ответа на основании найденного контекста.

Стек:

- Локальный сервер Ollama через OpenAI API;

Администрирование и авторизация

Назначение:

- управление пользователями;
- управление ролями;
- управление базами знаний;
- контроль прав доступа;

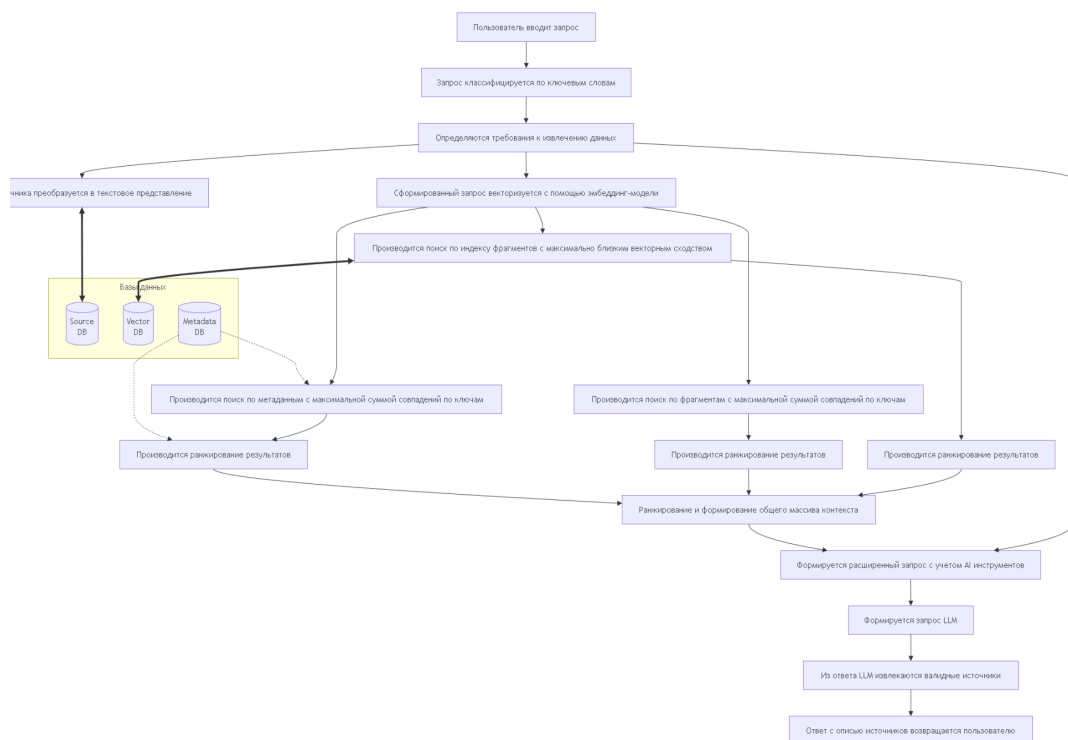
Логирование и отладка

Назначение:

- журналирование всех этапов обработки;
- диагностика работы пайплайна;
- контроль производительности;
- трассировка ошибок.

Интерфейсы уровня приложения

- внутренние Python-модули;
- REST API;
- интеграционные программные интерфейсы.



2.3. Уровень данных (Data Layer)

Уровень данных обеспечивает хранение документов, индексов и служебной информации.

Хранилище документов

Содержит оригинальные файлы:

- PDF

Векторное хранилище

Используется индекс FAISS, содержащий эмбединги текстовых фрагментов.

Хранилище метаданных

Содержит сведения о документах, может включать:

- автор;
- дата создания;
- версия;
- тип документа;
- принадлежность базе знаний;
- права доступа;
- служебные атрибуты.

Возможные реализации:

- JSON файловое хранение;

Интерфейсы

Доступ к данным осуществляется непосредственно из Backend посредством программных библиотек.

3. Взаимодействие компонентов системы

Взаимодействие между уровнями осуществляется следующим образом.

1. Пользователь формирует запрос через веб-интерфейс.
2. Запрос передаётся в Backend посредством REST API
3. Backend выполняет:
 - предобработку запроса;
 - гибридный поиск;
 - формирование контекста.
4. Контекст передаётся языковой модели.
5. Полученный ответ возвращается пользователю.
6. При необходимости пользователь открывает документ, а система отображает стартовый использованный фрагмент с использованием координат, полученных на этапе обработки PDF.

Пайплайн обработки документов включает следующие стадии:

Extraction → Chunking → Embedding → Indexing → Retrieval → Ranking → Generation

Все стадии являются независимыми и могут заменяться без изменения архитектуры остальных компонентов.

4. Архитектурный стиль

Основу архитектуры составляют следующие архитектурные подходы:

- Слои архитектуры (трёхуровневая архитектура);
- Модули архитектуры (модульная организация компонентов);
- Пайплайны (конвейер обработки документов).

При проектировании соблюдаются следующие принципы:

- Разделение процессов импорта документов, поиска и генерации ответа;
- Прозрачность всех этапов обработки данных;
- Минимизация внешних зависимостей и архитектурных абстракций;
- Совместное использование различных методов поиска;
- Возможность быстрого развёртывания в закрытом контуре организации.

5. Основные технические решения

В системе приняты следующие технические решения.

Компонент	Выбранное решение	Обоснование
Извлечение текста	pdfplumber	Получение координат текста для подсветки найденных фрагментов
Сегментация	spaCy	Лингвистически корректное разделение текста на смысловые блоки
Построение эмбеддингов	sentence-transformers (multilingual-e5-base)	Высокое качество семантического поиска для многоязычных документов
Векторный индекс	FAISS	Высокая скорость поиска и низкие требования к ресурсам

Поиск по ключевым словам	TF-IDF (scikit-learn)	Повышение точности поиска специализированной терминологии
Генерация ответов	Локальный сервер Ollama через OpenAI API	Возможность использования локальных языковых моделей
Хранение метаданных	файловое JSON хранилище	Простота сопровождения и отсутствие зависимостей

Отказ от использования специализированных RAG-фреймворков обеспечивает полный контроль над процессами обработки данных, уменьшает количество внешних зависимостей и повышает прозрачность работы системы.

Модульная структура позволяет заменять отдельные компоненты (например, модель эмбеддингов, механизм поиска или языковую модель) без изменения архитектуры остальных подсистем.

Использование локальных индексов и поддержка автономной работы обеспечивают возможность эксплуатации программного обеспечения в корпоративных информационных системах с повышенными требованиями к защите информации.

Сторонние компоненты и лицензии

Список использованных сторонних компонентов с указанием их правообладателей и лицензий их распространения

Компонент / Библиотека	Правообладатель	Лицензия	Ссылка на лицензию
Flask	Pallets Project	BSD-3-Clause	https://github.com/pallets/flask/blob/main/LICENSE.txt
Flask-CORS	Cory Dolphin	MIT	https://github.com/corydolphin/flask-cors/blob/main/LICENSE

Компонент / Библиотека	Правообладатель	Лицензия	Ссылка на лицензию
Flask-JWT-Extended	Violet B. (vroland)	MIT	https://github.com/vimalloc/flask-jwt-extended/blob/master/LICENSE
psycopg2	Psycopg Team	LGPL-3.0	https://github.com/psycopg/psycopg2/blob/master/LICENSE
requests	Kenneth Reitz	Apache-2.0	https://github.com/psf/requests/blob/main/LICENSE
os	Eric	MIT	https://opensource.org/licenses/mit
json	Python Software Foundation	Zero Clause BSD	https://docs.python.org/3/library/json.html
math	Pranav Singh	MIT	https://opensource.org/licenses/mit
uuid	Python Software Foundation	Zero Clause BSD	https://docs.python.org/3/library/uuid.html
Werkzeug	Pallets	BSD-3-Clause	https://github.com/pallets/werkzeug/blob/main/LICENSE.txt
datetime	Free Software Foundation	На основании открытой лицензии Zope Public License (ZPL)	https://github.com/zopefoundation/DateTime/blob/master/LICENSE.txt
markdown	Andreas Renberg	MIT	https://github.com/IQAndreas/markdown-licenses/blob/master/mit.md
scikit-learn (sklearn)	scikit-learn developers	BSD-3-Clause	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
pandas	NumFOCUS, Inc.	BSD-3-Clause	https://github.com/pandas-dev/pandas/blob/main/LICENSE

Компонент / Библиотека	Правообладатель	Лицензия	Ссылка на лицензию
pdfminer.six	pdfminer.six community	MIT	https://github.com/pdfminer/pdfminer.six/blob/master/LICENSE
pdfplumber	Jeremy Singer-Vine	MIT	https://github.com/jsvine/pdfplumber/blob/master/LICENSE.txt
Pillow	Secret Labs AB Fredrik Lundh and contributors	HPND / MIT-подобная	https://github.com/python-pillow/Pillow/blob/main/LICENSE
PyPDF2	Mathieu Fenniak	BSD 3-Clause	https://github.com/py-pdf/pypdf/blob/main/LICENSE
spaCy	ExplosionAI GmbH	MIT	https://github.com/explosion/spaCy/blob/master/LICENSE
pymorphy2	pymorphy2 developers	MIT	https://github.com/kmike/pymorphy2/blob/master/LICENSE.txt
pymorphy3	pymorphy3 developers	MIT	https://github.com/no-plagiarism/pymorphy3/blob/master/LICENSE.txt
bcrypt	OpenBSD Project	Apache-2.0	https://github.com/pyca/bcrypt/blob/main/LICENSE
OpenAI *Python Library	OpenAI, LLC	MIT	https://github.com/openai/openai-python/blob/main/LICENSE

*Примечание: Библиотека OpenAI Python Library применяется исключительно как клиентская прослойка для взаимодействия с локальным сервером Ollama по протоколу, совместимому с OpenAI API. Все вызовы направляются на локальный эндпоинт (<http://localhost:11434/v1>), передача данных во внешние облачные сервисы OpenAI не производится. Данное решение принято для обеспечения гибкости при смене бэкенда генерации без изменения кода приложения.

Используемые компоненты библиотеки, реализующие технологии искусственного интеллекта с указанием их правообладателей и лицензий их распространения

Компонент / Библиотека	Правообладатель	Лицензия	Ссылка на лицензию
Torch (Pytorch)	Linux Foundation	BSD-3-Clause	https://github.com/pytorch/pytorch/blob/main/LICENSE
torchvision	PyTorch Core Team Soumith Chintala	BSD-3-Clause	https://github.com/pytorch/vision/blob/main/LICENSE
torchaudio	Facebook Inc.	BSD-2-Clause	https://github.com/pytorch/audio/blob/main/LICENSE
transformers	The Hugging Face team	Apache-2.0	https://github.com/huggingface/transformers/blob/main/LICENSE
sentence-transformers	UKPLab	Apache-2.0	https://github.com/UKPLab/sentence-transformers/blob/master/LICENSE
FAISS	Meta Platforms, Inc.	MIT	https://github.com/facebookresearch/faiss/blob/main/LICENSE

Используемые модели машинного обучения с указанием их правообладателей и лицензий их распространения

Модель	Правообладатель	Лицензия	Ссылка на лицензию
ru_core_news_sm	ExplosionAI GmbH	MIT	https://github.com/explosion/spacy-models/releases/tag/ru_core_news_sm-3.7.0
en_core_web_sm	ExplosionAI GmbH	MIT	https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-3.7.1
multilingual-e5-base	Microsoft Research	MIT	https://huggingface.co/float/multilingual-e5-base

Модель	Правообладатель	Лицензия	Ссылка на лицензию
gemma 4:4b	Google	Apache 2.0	https://www.apache.org/licenses/LICENSE-2.0

Системные зависимости с указанием их правообладателей и лицензий их распространения

Компонент	Правообладатель	Лицензия	Ссылка на лицензию
Python 3.11	Python Software Foundation	PSF License Agreement	https://docs.python.org/3.11/license.html
Ollama	Ollama	MIT License	https://github.com/ollama/ollama/blob/main/LICENSE
PostgreSQL 14	The PostgreSQL Global Development Group	MIT-подобная	https://opensource.org/licenses/postgresql/
JavaScript	Microsoft	Apache-2.0	https://github.com/microsoft/TypeScript/blob/main/LICENSE.txt

6. Требования к аппаратному обеспечению (рекомендованные значения)

- ЦП: x86_64 архитектура, от 2 ядер
- ОЗУ: от 8 ГБ (зависит от объема индексируемой базы знаний и размера загруженной языковой модели).
- GPU: Опционально, рекомендуется NVIDIA с поддержкой CUDA 12+
- Дисковая подсистема: SSD от 25 ГБ (зависит от объема индексируемой базы знаний и размера загруженной языковой модели).
- ОС: Рекомендуется Astra Linux, Основа, или любой совместимый дистрибутив Linux

Архитектура программного обеспечения «Интеллектуальная корпоративная система интеграционного менеджмента информации "ИКСИМИ:ДОКУМЕНТЫ 5.0"» обеспечивает модульность, расширяемость и высокую производительность системы при сохранении простоты сопровождения и прозрачности обработки данных. Реализованный подход соответствует современным принципам построения корпоративных систем класса Retrieval-Augmented Generation и позволяет эффективно использовать программное обеспечение как в небольших организациях, так и в крупных корпоративных инфраструктурах с повышенными требованиями к безопасности и локальному хранению данных.