ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«ДЖИ-ЭМ-СИ-ЭС ВЕРЭКС»

Юридический адрес: 109004, г. Москва, вн. тер. г. муниципальный округ Таганский,
пер. Товарищеский, дом 7, строение 2, этаж 2, комната 7
Фактический адрес: 119602, г. Москва, ул. Покрышкина, д. 7
ИНН 7716593315, КПП 770901001
Тел.: (495) 737 99 91
info@gmcs.ru, www.gmcs.ru

Описание технической архитектуры

Программное обеспечение «ВерЭкс ИИ»

Сведения о Программном обеспечении

Программное обеспечение «ВерЭкс ИИ» представляет собой модульную распределённую веб-платформу, предназначенную для интеллектуальной обработки и анализа корпоративной информации с применением технологий искусственного интеллекта.

Программа обеспечивает извлечение, структурирование, хранение, поиск и анализ информации, содержащейся в корпоративных документах, базах знаний, электронных таблицах, презентациях, аудио- и видеоматериалах.

Для интеллектуальной обработки информации применяются технологии полнотекстового, семантического и гибридного поиска, генерации ответов с дополнением контекста из корпоративных источников (RAG), а также многошаговой агентной обработки запросов.

Взаимодействие пользователей с функциональными возможностями Программы осуществляется через веб-интерфейс, включая диалоговый интерфейс для формирования запросов на естественном языке, и через предусмотренные программные интерфейсы интеграции.

Программа поддерживает модульное расширение функциональности, подключение различных моделей искусственного интеллекта и прикладных инструментов, а также развёртывание в инфраструктуре организации.

С технической точки зрения, программное обеспечение «ВерЭкс ИИ» представляет собой распределённую веб-платформу, реализованную на языках C#, TypeScript и Python.

Платформа предназначена для корпоративной работы с генеративными моделями искусственного интеллекта, базами знаний и документами. Серверные компоненты поставляются в виде контейнеров Docker и взаимодействуют по HTTP API, протоколу MCP, через очередь сообщений и внутренние сетевые соединения.

Техническая архитектура «ВерЭкс ИИ» включает следующие основные компоненты:

1. Веб-интерфейс пользователя

Клиентская часть реализована на Angular и предоставляется пользователю через браузер. Веб-интерфейс обеспечивает:

- работу с чатами, базами знаний, файлами и настройками доступных моделей ИИ;
- передачу запросов и файлов в сервер приложений через REST API;
- получение потоковых ответов и статусов длительных операций через SignalR.

2. Сервер приложений и API

Центральный серверный компонент реализован на ASP.NET Core. Он выполняет бизнес-логику системы и координирует работу остальных сервисов:

- аутентифицирует пользователей, проверяет права доступа и управляет пользовательскими сессиями;
- обрабатывает запросы чатов, баз знаний, документов и административных функций;
- маршрутизирует обращения к моделям ИИ и внутренним сервисам;
- публикует события о ходе операций и результатах обработки.

3. Подсистема хранения данных

Для хранения различных типов информации используются специализированные компоненты:

- PostgreSQL — структурированные данные, настройки, пользователи, права доступа и метаданные;
- MinIO — исходные и сформированные файлы, документы и результаты мультимедийной обработки;
- OpenSearch — полнотекстовые и векторные индексы, используемые при поиске по базам знаний.

4. Сервис обработки документов и RAG

Сервис на Python подготавливает документы для поиска с дополнением контекста (RAG). В его задачи входят:

- получение файлов из объектного хранилища и извлечение содержимого;
- разбиение текста на фрагменты, построение векторных представлений и индексирование;
- гибридный поиск по полнотекстовым и векторным индексам OpenSearch;
- передача серверу приложений состояния и результатов обработки через защищённый внутренний API.

5. Сервис прикладных инструментов Python Tools

Изолированный Python-сервис предоставляет операции обработки файлов и табличных данных. Он получает файлы из MinIO, выполняет требуемую операцию и возвращает структурированный результат серверу приложений. Обработка PDF и распознавание текста выполняются поддерживаемыми серверными средствами, включая альтернативные способы обработки PDF-файлов.

Сервис MCP

Сервис реализует Model Context Protocol и предоставляет моделям ИИ контролируемый доступ к прикладным инструментам. Взаимодействие выполняется по HTTP и JSON-RPC; инструменты разделены на конфигурируемые пулы, доступ к которым защищается API-ключами. Сервис предоставляет проверки работоспособности и готовности.

7. Подсистема обработки аудио и видео

Мультимедийная обработка вынесена в специализированные сервисы. Используется программный комплекс Whisper-UI, реализованный на Python. Обеспечивает очистку аудио/видео файлов от пустых блоков и музыки, транскрибацию речи, разбивку на спикеров. И отдельный сервис на Python для выделения из видеофайлов кадров:

- маршрутизатор распознавания речи выбирает настроенный способ транскрибации;
- локальный сервис транскрибации выполняет асинхронные очистку, распознавание речи, разбиение на спикеров специализированными открытыми моделями, скачиваемыми с https://huggingface.co/ и использует системные исполняемые файлы FFmpeg/ffprobe для подготовки аудио;
- сервис обработки видео выделяет ключевые кадры, сохраняет результаты в MinIO и ведёт состояния задач в SQLite.

8. Очереди и фоновые процессы

RabbitMQ используется для передачи длительных и ресурсоёмких заданий между компонентами. Фоновая обработка не блокирует пользовательские HTTP-запросы. Состояние операции сохраняется сервером, а изменение статуса передаётся веб-интерфейсу через SignalR или получается периодическим опросом API.

9. Интеграция с моделями искусственного интеллекта

Сервер приложений обращается к настроенным поставщикам языковых, векторных и мультимодальных моделей через унифицированные программные интерфейсы. Конкретные модели, адреса сервисов и ключи доступа задаются конфигурацией развёртывания. При локальном размещении моделей вычислительные требования определяются выбранной моделью и не входят в базовую конфигурацию платформы.

Логическое взаимодействие компонентов

Пользователь работает в браузере с веб-интерфейсом, который передаёт запросы серверу приложений.

Сервер приложений проверяет полномочия, выполняет бизнес-логику и обращается к PostgreSQL, MinIO и OpenSearch.

Длительные операции передаются через RabbitMQ в профильные сервисы; сведения о ходе выполнения возвращаются серверу приложений.

Для формирования ответа сервер приложений вызывает настроенную модель ИИ, RAG-поиск и, при необходимости, инструменты MCP или изолированную среду исполнения кода.

Результат и события выполнения передаются веб-интерфейсу по REST API и SignalR.

Развёртывание, безопасность и эксплуатация

Компоненты развёртываются в контейнерах Docker в изолированной сети.

Постоянные данные размещаются в томах PostgreSQL, MinIO, OpenSearch, RabbitMQ и локальных хранилищах специализированных сервисов.

Внешняя публикация адресов и защита транспортного канала определяются инфраструктурой заказчика.

- пользовательский доступ защищается механизмами аутентификации и авторизации сервера приложений;
- внутренние API сервисов защищаются отдельными API-ключами;
- секреты, адреса сервисов и параметры моделей передаются через конфигурацию среды развёртывания;
- для компонентов предусмотрены журналы событий, проверки работоспособности и диагностика доступности инфраструктуры;
- сборка и поставка контейнеров автоматизируются конвейерами CI/CD.

Системные требования Программного обеспечения «ВерЭкс ИИ»

Системное программное обеспечение

- Операционная система сервера: Linux (Debian, Ubuntu, Astra Linux либо совместимая ОС).

- Среда контейнеризации: Docker с поддержкой Docker Compose.
- СУБД: PostgreSQL 16; объектное хранилище MinIO; поисковая система OpenSearch; брокер сообщений RabbitMQ.
- Клиент: современный браузер на базе Chromium; клиентская операционная система — любая, поддерживающая такой браузер.
- Сетевое взаимодействие: доступ по HTTP/HTTPS к опубликованному веб-интерфейсу и доступ контейнеров к внутренним сервисам и настроенным моделям ИИ.

Ниже приведена ориентировочная конфигурация для серверного развёртывания без локального размещения ресурсоёмких языковых моделей.

Фактические параметры зависят от числа пользователей, объёма документов, интенсивности индексации и требований к отказоустойчивости.

Сервер приложений и сервисов

- Доступное дисковое пространство: от 60 Гб для контейнеров, журналов и временных данных.
- Процессор: от 12 вычислительных ядер.
- Оперативная память: от 64 Гб.
- Сетевой интерфейс: не менее 1 Гбит/с для связи с хранилищами и внешними сервисами.

Сервер данных и поисковых индексов

- Доступное дисковое пространство: от 300 Гб с возможностью расширения.
- Процессор: от 12 вычислительных ядер.
- Оперативная память: от 64 Гб.
- Сетевой интерфейс: не менее 1 Гбит/с для связи с сервером приложений.

Дополнительные требования при локальном размещении моделей ИИ

Тип и объём оперативной памяти, видеопамяти и дискового пространства определяются выбранными языковыми, векторными и мультимодальными моделями. Эти ресурсы рассчитываются отдельно после утверждения состава моделей и требований к производительности.