

**Описание функциональных характеристик
Программного обеспечения (программы для
ЭВМ) «Мимир: локальный RAG-поиск по
НСИ и конфиденциальным данным с
искусственным интеллектом»**

2026 г.

Содержание

Термины и определения	3
1. Описание системы	5
2. Перечень реализуемых функций системы.....	5
2.1. Пользовательские функции	5
2.2. Административные функции	5
2.3. Интеграционные функции	5
3. Описание частей системы	5
3.1. Подсистема аутентификации и управления доступом	5
3.2. Подсистема обработки пользовательских запросов	6
3.3. Подсистема поиска и извлечения информации (Retrieval).....	6
3.4. Подсистема формирования ответа (Generation)	6
3.5. Подсистема управления базой знаний.....	6
3.6. Подсистема журналирования и мониторинга.....	7
3.7. Подсистема интеграционного взаимодействия	7
4. Протокол взаимодействия, формат данных	8
4.1. Взаимодействие пользовательского интерфейса с серверной частью	8
4.2. Взаимодействие серверной части с компонентом обработки естественного (русского) языка.....	8
4.3. Общие требования к протоколам и форматам.....	8

Термины и определения

Администратор системы — пользователь, обладающий расширенными правами доступа и осуществляющий управление параметрами функционирования системы, источниками данных, ролями пользователей и настройками обработки запросов.

Аутентификация — процедура установления подлинности пользователя посредством проверки его учётных данных перед предоставлением доступа к функциональным возможностям системы.

База знаний — совокупность внутренних документов организации (регламентов, инструкций, методических материалов и иных текстовых источников), используемых системой в качестве источника данных для поиска и формирования ответов.

Веб-интерфейс системы — программно реализованный пользовательский интерфейс, обеспечивающий взаимодействие пользователя с системой через веб-браузер.

Версионирование API — механизм управления изменениями программного интерфейса, предусматривающий поддержку нескольких версий API для обеспечения совместимости интеграций.

Генерация ответа — автоматизированный процесс формирования текстового результата на основе анализа запроса пользователя и релевантных фрагментов документов базы знаний.

Документ базы знаний — структурированная или неструктурированная единица информации (файл, текстовый материал, запись), включённая в состав базы знаний и подлежащая индексации.

Индексирование — процесс автоматизированной обработки документов базы знаний с целью обеспечения их последующего поиска и использования при формировании ответа.

Информационный запрос пользователя — текстовое сообщение, вводимое пользователем в интерфейсе системы с целью получения информации из базы знаний.

Информационная безопасность — совокупность организационных и технических мер, реализованных в системе для защиты данных от несанкционированного доступа, изменения, удаления или распространения.

Контекст запроса — совокупность данных, используемых системой при формировании ответа, включая текст запроса и найденные релевантные фрагменты документов.

Модель обработки естественного (русского) языка — программный компонент системы, реализующий алгоритмы анализа текстовых данных и генерации текстового ответа.

Ответ системы — текстовый результат обработки запроса пользователя, сформированный системой на основе данных базы знаний.

Пользователь системы — физическое лицо, обладающее правом доступа к системе и осуществляющее взаимодействие с её функциональными возможностями в пределах назначенных прав.

Программное обеспечение (Система) — программа для ЭВМ «Система интеллектуального поиска по базе знаний с использованием технологии Retrieval-Augmented Generation (RAG)», предназначенная для обработки текстовых запросов пользователей и предоставления ответов на основании внутренних информационных ресурсов организации.

Ролевая модель доступа — механизм разграничения прав пользователей,

основанный на назначении ролей с определённым набором разрешённых действий.

Технология Retrieval-Augmented Generation (RAG) — метод обработки запросов, при котором перед формированием ответа осуществляется поиск релевантных фрагментов документов базы знаний, которые используются в качестве контекста генерации ответа.

API (Application Programming Interface) — программный интерфейс взаимодействия, обеспечивающий обмен данными между компонентами системы либо между системой и внешними информационными ресурсами.

HTTP (HyperText Transfer Protocol) — прикладной протокол передачи данных, используемый для взаимодействия между компонентами системы.

HTTPS (HyperText Transfer Protocol Secure) — защищённая версия протокола HTTP, обеспечивающая шифрование передаваемых данных и защиту от несанкционированного перехвата.

JSON (JavaScript Object Notation) — текстовый формат обмена данными, применяемый для передачи структурированной информации между компонентами системы.

OpenAI-совместимый API — программный интерфейс, реализующий формат и структуру запросов и ответов, совместимые со спецификацией OpenAI API, используемый для взаимодействия с компонентом обработки естественного (русского) языка.

REST (Representational State Transfer) — архитектурный стиль построения программных интерфейсов, основанный на использовании стандартных методов протокола HTTP и принципа отсутствия сессионного состояния.

Stateless-взаимодействие — принцип организации обмена данными, при котором каждый запрос к системе является независимым и содержит всю необходимую информацию для его обработки без сохранения серверного сессионного состояния.

Streaming (поточковая передача данных) — режим передачи данных, при котором ответ системы передаётся пользователю поэтапно по мере формирования результата обработки запроса.

1. Описание системы

Полное наименование программного обеспечения: «Мимир: локальный RAG-поиск по НСИ и конфиденциальным данным с искусственным интеллектом»

Программа для ЭВМ представляет собой информационную систему, предназначенную для обработки текстовых запросов пользователей и предоставления ответов на основе внутренних документов организации.

Система обеспечивает:

- Приём текстовых запросов пользователей через веб-интерфейс;
- Автоматизированный поиск релевантной информации в базе знаний;
- Формирование текстового ответа на основании найденных данных;
- Разграничение прав доступа пользователей к функциональным возможностям системы;
- Ведение журналирования действий пользователей и обработки запросов.

В качестве источника данных используется база знаний, сформированная из внутренних нормативных, регламентирующих и методических документов организации.

Система функционирует в виде веб-приложения и обеспечивает доступ к своим функциям посредством сети передачи данных с использованием стандартных протоколов взаимодействия.

Назначением системы является обеспечение сотрудников организации инструментом оперативного получения информации из внутренних документов посредством обработки запросов на естественном (русском) языке.

2. Перечень реализуемых функций системы

2.1. Пользовательские функции

- Аутентификация и авторизация пользователей.
- Приём и обработка текстовых запросов на естественном (русском) языке.
- Поиск релевантной информации в базе знаний.
- Формирование текстового ответа на основе найденных данных.

2.2. Административные функции

- Назначение и управление ролями доступа.
- Управление источниками данных базы знаний.
- Индексация и переиндексация документов базы знаний.
- Настройка параметров обработки запросов.
- Ведение журналирования действий пользователей и событий системы.

2.3. Интеграционные функции

- Интеграция с внутренними источниками данных организации.
- Предоставление программного интерфейса (API) для взаимодействия с внешними системами.
- Обеспечение обмена данными с использованием стандартных протоколов взаимодействия.
- Поддержка расширения перечня источников данных.

3. Описание частей системы

3.1. Подсистема аутентификации и управления доступом

Функциональные требования:

- Идентификация и аутентификация пользователей при входе в систему.
- Авторизация пользователей в соответствии с назначенными ролями.
- Разграничение прав доступа к функциональным возможностям системы.
- Ограничение доступа к данным базы знаний в соответствии с правами пользователя.
- Поддержка назначения и изменения ролей доступа.

Модели данных:

- Пользователь.
- Роль.
- Матрица прав доступа.

3.2. Подсистема обработки пользовательских запросов

Функциональные требования:

- Приём текстовых запросов пользователей через веб-интерфейс.
- Обработка запросов на естественном (русском) языке.
- Формирование контекста обработки запроса.
- Формирование текстового ответа на основе релевантных данных.
- Отображение результатов обработки запроса пользователю.
- Сохранение истории диалога с системой.

Модели данных:

- Запрос пользователя.
- Ответ системы.
- История диалога с системой.

3.3. Подсистема поиска и извлечения информации (Retrieval)

Функциональные требования:

- Индексация документов базы знаний.
- Поиск релевантных фрагментов документов по содержанию запроса.
- Отбор и передача релевантных данных в подсистему формирования ответа.
- Поддержка обновления и переиндексации документов.
- Обеспечение быстродействия поиска при работе с массивом документов.

Модели данных:

- Документ базы знаний.
- Индексированное представление документа.
- Фрагмент документа.

3.4. Подсистема формирования ответа (Generation)

Функциональные требования:

- Использование технологии Retrieval-Augmented Generation (RAG).
- Формирование ответа на основе текста запроса и найденных релевантных данных.
- Генерация структурированного текстового результата.
- Ограничение объёма ответа в соответствии с параметрами системы.
- Обеспечение корректности и целостности формируемого ответа.

Модели данных:

- Контекст запроса.
- Сформированный ответ.

3.5. Подсистема управления базой знаний

Функциональные требования:

- Подключение источников данных базы знаний.
- Загрузка и обновление документов.
- Управление доступностью документов для поиска.
- Контроль актуальности данных.
- Управление структурой хранения документов.

Модели данных:

- Источник данных.
- Документ.
- Метаданные документа.

3.6. Подсистема журналирования и мониторинга

Функциональные требования:

- Регистрация действий пользователей.
- Регистрация операций обработки запросов.
- Хранение журналов событий системы.
- Обеспечение возможности анализа работы системы.
- Контроль корректности функционирования системы.

Модели данных:

- Событие системы.
- Журнал действий.

3.7. Подсистема интеграционного взаимодействия

Функциональные требования:

- Предоставление программного интерфейса (API) для взаимодействия с внешними системами.
- Приём запросов от внешних информационных систем.
- Обеспечение передачи результатов обработки запросов.
- Поддержка стандартных протоколов взаимодействия.
- Обеспечение безопасного обмена данными.

Модели данных:

- Внешний запрос.
- Внешний ответ.
- Параметры интеграции.

4. Протокол взаимодействия, формат данных

Взаимодействие компонентов системы осуществляется с использованием стандартных сетевых протоколов и форматов передачи данных.

4.1. Взаимодействие пользовательского интерфейса с серверной частью

Обмен данными между веб-интерфейсом системы и серверной частью осуществляется:

- по протоколу HTTP и HTTPS;
- с использованием формата передачи данных JSON;
- с применением REST-подхода к организации программного интерфейса (API).

Запросы к серверной части направляются посредством HTTP-методов (в том числе POST и GET) к API-эндпоинтам, реализующим обработку пользовательских сообщений.

Ответы сервера передаются в формате JSON и содержат результат обработки запроса пользователя.

В системе поддерживается режим потоковой передачи (streaming) данных при формировании ответа, обеспечивающий поэтапную отправку результата обработки запроса.

4.2. Взаимодействие серверной части с компонентом обработки естественного (русского) языка

Серверная часть системы взаимодействует с компонентом обработки естественного (русского) языка посредством OpenAI-совместимого API.

Передача данных осуществляется:

- по протоколу HTTP/HTTPS;
- в формате JSON;
- без использования сессионного состояния (stateless-взаимодействие).

4.3. Общие требования к протоколам и форматам

Передача данных между компонентами системы осуществляется по защищённому протоколу HTTPS при работе в производственной среде.

- Формат данных — JSON.
- API поддерживает версионирование.
- Взаимодействие реализуется с соблюдением метода архитектуры, при котором сервер не хранит информацию о предыдущих запросах клиента (без сохранения состояний).

Изменения в API не допускаются без обеспечения обратной совместимости либо введения новой версии интерфейса.