Модуль управления ИИ-агентами

STC-S9013.01

Руководство администратора

НЦДА.01367-01 91 01

ЦРТ инновации

ОГЛАВЛЕНИЕ

Оглавление 1
Аннотация 4
Термины и определения 5
1. Введение 7
1.1. Общие сведения 7
1.1.1. Сведения о продукте и изготовителе 7
1.1.2. Товарные знаки 7
1.1.3. Служба технической поддержки 7
1.2. Назначение решения 8
1.3. Уровень подготовки персонала 9
2. Основные архитектурные решения Модуля 10
2.1. Технологический стек 10
2.2. Структура проекта Модуль 12
2.3. Микросервисы 12
2.4. Схема взаимодействия сервисов 13
2.5. Архитектура AgentService 15
2.6. Архитектура MeetingService 16
2.7. Архитектура AgentsRegistry 17
2.8. Основные потоки данных 17
2.9. Уровни зависимостей библиотек. Shared Libraries (слои) 20
2.10. Контекстная диаграмма 20
2.11. Контейнерная диаграмма 21
2.12. Модель данных 21
2.12.1. Хранилище реляционных данных PostgreSQL (DbService) 22
2.12.2. Хранилище документо-ориентированных данных чатов OpenSearch (ContextStorage) 25
2.12.3. Хранилище S3 файлов и архивов агентов/скиллов S3 (Объектное хранилище) 27
2.12.4. Общая схема взаимодействия хранилищ 28
2.13. Взаимодействие Модуля и системы «AgentNavigator» 28

3. Описание API Модуля 29
3.1. Сводная таблица микросервисов API 29
3.2. AgentService API 29
3.3. ContextStorage API 32
3.4. MeetingService API 33
3.5. DbService API 36
3.6. Authorization API 39
3.7. AgentsRegistry API 40
3.8. InterfaceCompositor API 45
3.9. ShureControl API 46
3.10. UserManagement API 46
3.11. PlatformKafka 47
4. Взаимодействие администратора/пользователя с API Модуля 48
4.1. Аутентификация 48
4.2. Формат ответов 48
4.3. Коды HTTP 48
5. Системные требования 50
5.1. Требования к аппаратному обеспечению Модуля 50
5.2. Требования к программному обеспечению Модуля 50
5.3. Требования к клиентской компоненте Модуля 51
6. Установка программного обеспечения Модуля 52
6.1. Шаги установки 52
7. Настройка компонентов Модуля. 53
7.1. Рабочее пространство «Настройки» 53
7.2. Управление агентами 54
7.2.1. Реестр агентов 54
7.2.2. Карточка агентов 55
7.2.3. Создание агента 56
7.2.4. Редактирование агента 58
7.2.5. Подключение навыков к агенту 58
7.2.6. Присвоение версий агенту 58

7.2.7. Архивация агента ... 59
7.3. Управление навыками ... 59
7.3.1. Реестр навыков ... 60
7.3.2. Карточка навыка ... 61
7.3.3. Добавление навыка ... 62
7.3.4. Редактирование навыка ... 63
7.3.5. Подключение инструментов к навыку ... 64
7.3.6. Присвоение версий навыку ... 64
7.4. Управление инструментами ... 64
7.4.1. Реестр инструментов ... 65
7.4.2. Карточка инструмента ... 66
8. Управление учетными записями пользователей ... 68
8.1. Раздел «Доступ и управление» ... 68
8.1.1. Подраздел «Пользователи» ... 68
8.1.2. Карточка пользователя ... 69
8.1.2.1. Основная информация о пользователе ... 70
8.1.2.2. Роли и доступы ... 70
8.1.2.3. Сессии ... 70
9. Эксплуатационно-техническое обслуживание Модуля ... 72
9.1. Проверка состояния компонентов Модуля ... 73
9.2. Настройка диагностики ... 74
9.3. Запуск диагностического модуля ... 75
Приложение 1. Перечень сторонних компонентов Модуля ... 77
Основной проект C# библиотеки ... 77
Используемые в архитектуре сторонние компоненты ... 82
Сторонние Frontend компоненты ... 83

АННОТАЦИЯ

Настоящий документ является руководством администратора Модуля управления ИИ-агентами (далее – Модуль).

Документ описывает основные сведения об архитектуре Модуля, требования к программно-аппаратной платформе, на которой будет развертываться Модуль, сценарии установки и настройки Модуля, а также порядок эксплуатационно-технического обслуживания Модуля.

Документ предназначен для администраторов (системных администраторов, представителей IT-службы организаций, ответственных за сопровождение и обслуживание) Модуля.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Табл. 1. Перечень терминов, сокращений и обозначений

Термин, сокращение, обозначение	Описание, определение, расшифровка
DOCX	(от англ. Microsoft Word Open XML Document) – формат файла, используемый для хранения документов, созданных в Microsoft Word
LLM	(от англ. Large Language Model) — большая языковая модель
ZIP	Формат архивных файлов
Агент	Специализированный программный компонент на базе ИИ, предназначенный для выполнения определенного типа задач пользователя в рамках диалога
Бакет	От англ. Buket-«ведро», «корзина». В облачных технологиях и системах хранения – базовая логическая единица для хранения файлов, данных и медиа
БД	База данных
Билд	Сборка программы и компиляция исходного кода программы в исполняемый файл или приложение
Деплой	Размещение готовой версии ПО на платформе, доступной для пользователей
ИИ	Искусственный интеллект
Инстанс	Запущенный процесс или изолированная среда сервера СУБД, которая работает с оперативной памятью и процессорным временем для обработки запросов
Инструмент	Программный компонент, подключаемый к навыку агента и расширяющий его функциональные возможности
Канал	Диалог, в котором пользователь обращается к менеджеру агентов, конкретному узконаправленному агенту/агентам, а также подключает пользователя/пользователей по необходимости

Термин, сокращение, обозначение	Описание, определение, расшифровка
Менеджер агентов	ИИ-агент, который на основании запроса пользователя подбирает подходящего агента из доступных на платформе либо сообщает об отсутствии подходящего агента
Модуль	Модуль управления ИИ-агентами
Навык	Программный компонент, подключаемый к агенту и определяющий его поведение при выполнении задач
Пагинация	Способ разделения большого объема данных или длинного списка на отдельные последовательные страницы при выдаче ответов клиенту
Пинговать	Посылать запросы для проверки доступности сервера или сайта и качества связи
ПО	Программное обеспечение
Резолвед	Решенный/разрешенный (о проблеме или вопросе)
Скилл	Навык
Скролл, скроллинг	Процесс перемещения содержимого (текста, картинок, страниц) вверх, вниз, влево или вправо на экране устройства.
Сообщение	Минимальная единица взаимодействия внутри диалога, содержащая текстовую информацию, отправленную пользователем, LLM или агентом
СУБД	Система управления базами данных
Токен	Минимальный фрагмент информации (слово, часть слова, символ), на которые нейросеть разбивает текст при обработке(токенизиция)
ЦП	Центральный процессор
Чат	Диалог, в котором происходит общение пользователя с LLM-моделью (ИИ-помощник)
Эндпойнт	Конечная точка, точка входа или точка взаимодействия

1. ВВЕДЕНИЕ

1.1. Общие сведения

1.1.1. Сведения о продукте и изготовителе

Наименование: Модуля управления ИИ-агентами

Изготовитель: Общество с ограниченной ответственностью «ЦРТ-инновации»

Адрес: 194044, Санкт-Петербург, ул. Гельсингфорсская, д. 3, к. 11 лит. Д, пом. 229

Телефон: +7 (812) 325-88-48

Факс: +7 (812) 327-92-97

1.1.2. Товарные знаки

Названия компаний и названия продуктов, упомянутые в документе, являются собственностью их соответствующих владельцев.

Иллюстративный материал, приведённый в документе, может содержать вымышленные имена, фамилии, номера телефонов, адреса и тому подобные сведения. Любое их совпадение с данными реальных личностей следует считать случайностью.

Интерфейс на рисунках может незначительно отличаться от интерфейса продукта в связи с его постоянной модернизацией.

Ни одна из частей этого издания не подлежит воспроизведению, передаче, хранению в поисковой системе или переводу на какой-либо язык в любой форме, любыми средствами без письменного разрешения общества с ограниченной ответственностью «ЦРТ-инновации».

1.1.3. Служба технической поддержки

Адрес службы сервисного обслуживания и технической поддержки в интернете:

Электронная почта: support@speechpro.com

Адрес в сети Интернет: http://www.speechpro.ru/support

При обращении в службу технической поддержки необходимо представить четкое описание возникшей проблемы.

Предварительно подготовьте следующую информацию:

- – сведения о конфигурации серверов;

- – название и версии используемых операционных систем;
- – версия программного обеспечения.

1.2. Назначение решения

Модуль представляет собой микросервисную платформу для организации работы с ИИ-агентами, записи и обработки материалов совещаний.

Модуль предназначен для:

- управления интеллектуальными агентами;
- управления навыками и библиотекой знаний;
- организации взаимодействия сотрудников с агентами;
- формирования ответов, документов и аналитических материалов;
- поддержки сценариев совместной работы;
- записи, распознавания и анализа совещаний;
- интеграции с внешними системами и сервисами;
- обеспечения аудита и контроля качества функционирования Системы.

Целями разработки и внедрения Модуля являются:

- повышение эффективности корпоративных процессов;
- сокращение времени на поиск, подготовку и обработку информации;
- повышение качества подготовки ответов и документов;
- обеспечение прозрачности исполнения агентных сценариев;
- централизация управления знаниями, агентами и правами доступа;
- повышение уровня автоматизации и стандартизации взаимодействий.

Модуль обеспечивает:

- Поддержку множества LLM-провайдеров (Ollama, GigaChat, vLLM, Cloud.ru Evolution).
- Управление агентами и скиллами.
- Cowork-сессии (несколько участников).
- Интеграцию с Kafka.
- Управление микрофонами Shure.
- Корпоративную аутентификацию через LDAP.

1.3. Уровень подготовки персонала

Администраторы должны знать техническую и эксплуатационную документацию Модуля и обладать квалификацией, позволяющей выполнять следующие функции:

- вносить изменения в настройки Модуля и создавать самостоятельные настройки в соответствии с требованиями пользователей;
- уметь выполнять типовые операции всех функций и служб Модуля;
- уметь моделировать варианты использования Модуля в тех или иных бизнес–процессах;
- вести инструктаж новых пользователей;
- настраивать роли и права доступа пользователей Модуля;
- владеть навыками работы с актуальными версиями СУБД, обеспечивающих работу Модуля;
- осуществлять контроль функционирования Модуля и устранение возникающих отказов и последствий сбоев;
- проведение регламентных работ в соответствии с технической документацией Модуля.

2. ОСНОВНЫЕ АРХИТЕКТУРНЫЕ РЕШЕНИЯ МОДУЛЯ

1. Сервис-ориентированная архитектура – каждый сервис отвечает за свою доменную область.

2. REST API – все сервисы общаются через HTTP/REST.

3. Consul – для service discovery и централизованной конфигурации.

4. JWT-аутентификация – токены для межсервисного взаимодействия.

5. Shared Libraries – общие библиотеки вынесены в Common/ и Data/.

2.1. Технологический стек

Перечень технологических стеков, используемых Модулем, приведен в Табл. 2

Табл. 2. Перечень технологических стеков, используемых Модулем

Компонент	Технология
Runtime	.NET 10.0 (net10.0), C#
DI-контейнер	Autofac 9.1.0
ORM	Entity Framework Core 10.0.10
PostgreSQL	Npgsql 10.0.1
Поиск	OpenSearch 1.8.0
Брокер сообщений	Confluent Kafka 2.12.0
Service Discovery	Consul 1.8.0
Логирование	NLog 6.1.2
Маппинг DTO	Mapster 10.0.11
Документирование API	Swagger (Swashbuckle) 10.2.3
Конвертация документов	DoclingSharp 8.17.9
S3-совместимое хранилище	AWS SDK S3 4.0.21.2
Docker-клиент	Docker.DotNet.Enhanced 4.3.3
LLM-клиенты	| OllamaSharp 5.4.25, Microsoft.Extensions.AI.Abstractions 10.7.0
Аутентификация	JWT Bearer, LDAP (Kerberos.NET 4.6.146)

Компонент	Технология
Аудио	NAudio 2.2.1
Тестирование	xUnit 2.9.2

2.2. Структура проекта Модуль

```
services/
├── Common/ # Общие библиотеки
│   ├── Common/ # Ядро: утилиты, шифрование, аудио
│   ├── Common.Exceptions/ # Типы исключений
│   ├── Common.Runtime/ # P/Invoke, окружение, ввод
│   ├── Consul/ # Интеграция с Consul
│   ├── Database/ # EF Core, PostgreSQL, SQLite
│   ├── DocConverter/ # Конвертация документов
│   ├── GigaChat/ # Клиент GigaChat API
│   ├── OpenSearch/ # Адаптер OpenSearch
│   ├── Rest.Base/ # REST DTO, enum и базовые типы
│   ├── Rest.Client/ # REST-клиентский фреймворк
│   └── Rest.Server/ # REST-сервер, Swagger, auth middleware
│
├── Data/ # Библиотеки данных
│   ├── Auth/ # Авторизация, CurrentUserProvider
│   ├── Mapping/ # Маппинг на основе Mapster
│   ├── Rest.Custom/ # REST endpoints сервисов
│   ├── Rest.Data/ # DTO всех сервисов
│   ├── S3/ # AWS S3-клиент
│   └── Settings/ # Глобальные настройки
│
├── AgentService/ # Микросервис AI-агентов
├── AgentsRegistry/ # Регистрация агентов
├── Authorization/ # Аутентификация
├── ContextStorage/ # Хранение контекста
├── DbService/ # API базы данных
├── InterfaceCompositor/ # Композиция пользовательского интерфейса
├── MeetingService/ # Запись и обработка совещаний
├── PlatformKafka/ # Kafka Gateway
├── ShureControl/ # Управление микрофонами
├── UserManagement/ # Управление пользователями
│
├── Directory.Build.props # Общие MSBuild-свойства
└── sln # Solution-файл
```

2.3. Микросервисы

Перечень микросервисов, входящих в состав Модуля, с указанием используемых портов приведен в Табл. 3.

Табл. 3. Перечень микросервисов, входящих в состав Модуля

Сервис	Порт	Назначение
AgentService	30021	Оркестрация AI-агентов, задачи, каналы, чаты, промпты, sandbox
AgentsRegistry	34571	Управление агентами и навыками, загрузка архивов, версионирование, загрузка архивов, привязка агентов к пользователям, доступы пользователей
Authorization	34567	Аутентификация (пароль/LDAP), выдача JWT-токенов, сервисные токены.
ContextStorage	34569	Хранение контекста сессий (чаты, каналы) в OpenSearch
DbService	34570	Единый API к PostgreSQL: чтение, сохранение и изменение данных (агенты, скиллы, пользователи, совещания).
InterfaceCompositor	3001	Композиция пользовательского интерфейса (фронтенд), bootstrap, workspace
MeetingService	5260	Запись, обработка и архив совещаний; транскрибация, диаризация, LLM-анализ, контекстные подсказки и выгрузка протокола в формате DOCX
PlatformKafka	30051	Kafka-шлюз: подписка на топики, проксирование сообщений в AgentService
ShureControl	30061	Управление микрофонами Shure (mute/unmute, состояние кнопки)
UserManagement	34568	Управление пользователями, правами и персональными данными

2.4. Схема взаимодействия сервисов

Схема взаимодействия сервисов Модуля изображена на Рис. 1

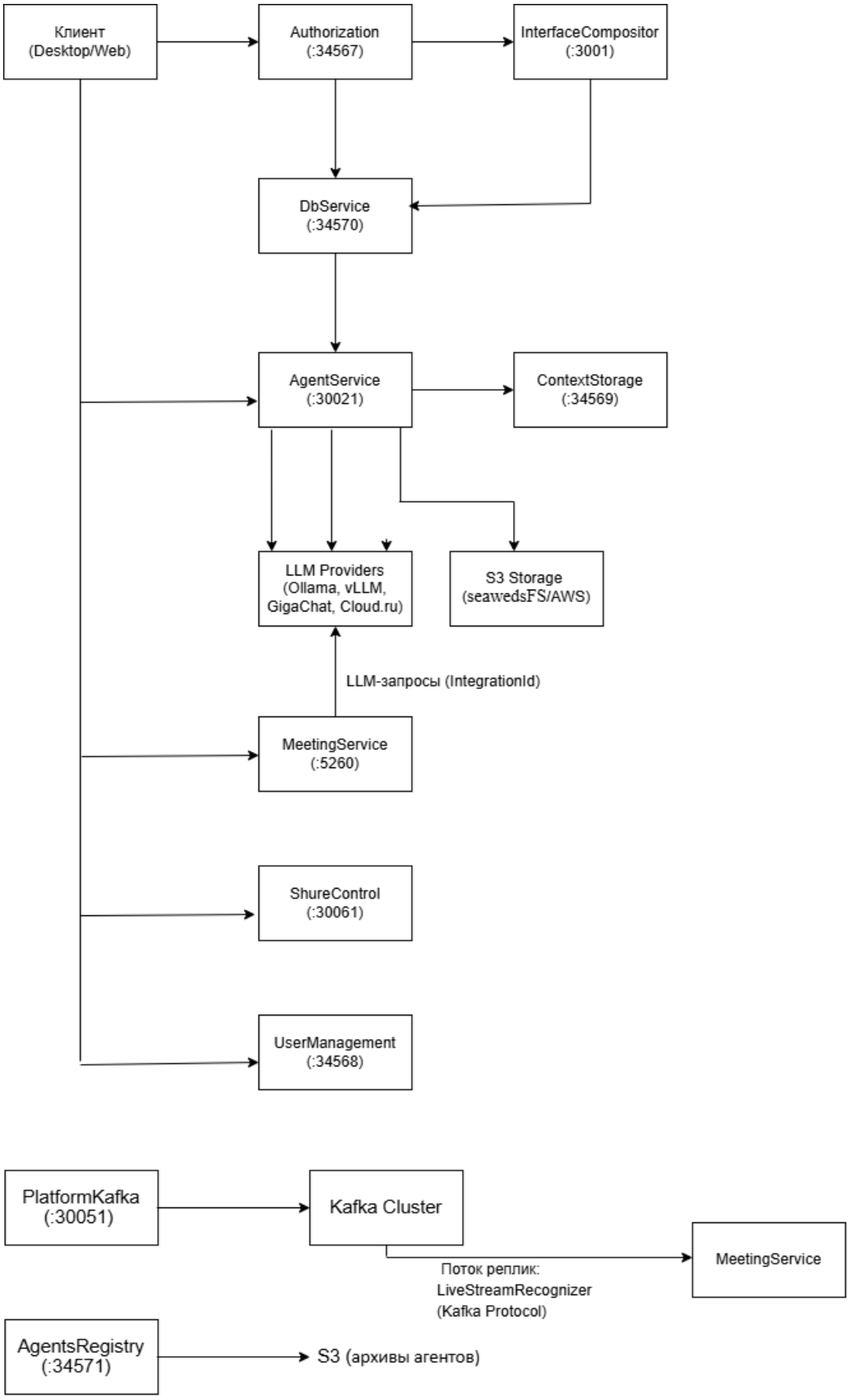


Рис. 1. Схема взаимодействия сервисов Модуля

2.5. Архитектура AgentService

```
AgentService
├── Controllers (REST API)
│   ├── ChatController
│   ├── ChannelController
│   ├── CoworkController
│   ├── DirectLlmController
│   └── AgentServiceController
├── Services
│   ├── ChatService — бизнес-логика чатов
│   ├── SessionService — управление сессиями
│   ├── HistoryProvider — история сообщений
│   ├── SandboxManager — Docker-песочницы
│   └── DirectLlmMessageService — прямые LLM-запросы
├── Agents
│   ├── AgentProvider — провайдер данных агентов
│   └── RouterAgent — маршрутизация к LLM
└── ChatClient
    ├── OllamaChatClient
    ├── GigaChatFileRehostClient
    ├── VllmChatClient
    └── CloudRuEvolutionChatClient
```

AgentService поддерживает три типа сессий:

Тип	Описание
Chat	Обычный чат: пользователь ↔ AI-агент
Channel	Канал: пользователь ↔ AI-агент (отдельная категория)
Cowork	Cowork-сессия: несколько пользователей и/или агентов

2.6. Архитектура MeetingService

```
MeetingService
├── Controllers (REST API)
│   ├── MeetingsController — жизненный цикл совещаний, задачи, транскрипты,
│   │  хинты, генерация документов (.docx)
│   ├── PersonsController — управление участниками совещаний
│   ├── FileProcessingController — загрузка аудио/видео, получение результатов
│   │  обработки, сохранение списков участников
│   ├── LlmSettingsController — управление настройками LLM
│   └── PreviewController — предпросмотр файлов (текст, повестка, участники)
│
├── Services — бизнес-логика
│   ├── MeetingService — совещания, задачи, транскрипты, хинты, протоколы
│   ├── PersonService — участники совещаний
│   ├── RecorderService — управление записью через MediaConnector
│   ├── MediaProcessingService — обработка загруженных аудио/видеофайлов
│   ├── MeetingDocGenerator — генерация документов протокола (.docx)
│   ├── RecognitionProcessingWorker — фоновый обработчик распознавания речи
│   ├── PhraseConsumer / RecognitionResultConsumer — Kafka-консюмеры
│   └── Сервисы анализа, хинтов, нормализации транскриптов
│
├── Models — API-модели
│   └── Модели совещаний, задач, участников, хинтов, транскриптов, LLM
│
├── Settings — конфигурационные секции
│   └── Настройки: Recorder, Kafka, Diarization, LLM, Retry-политики,
│  DocGenerator, Kestrel, FormOptions
│
└── MeetingService.DataAccess
    ├── MeetingServiceDbContext — EF Core DbContext (PostgreSQL)
    ├── Configurations — маппинги сущностей Meeting, MeetingAgenda, MeetingTask,
    │  MeetingTranscript, Person, MeetingPerson, хинты, задачи обработки
    ├── Entities — сущности БД
    └── Migrations — миграции EF Core
```

2.7. Архитектура AgentsRegistry

AgentsRegistry
- AgentsController
  - Управление агентами
  - Версионирование агентов
  - Привязка навыков к агентам
- SkillsController
  - Управление навыками
  - Версионирование навыков
- UserAgentsController
  - Назначение агентов пользователям
- Проверка прав доступа
- Валидация manifest.json
- Загрузка архивов в S3
- DbServiceRestPoint

└── HTTP-вызовы DbService

2.8. Основные потоки данных

1. Поток чата (Chat Prompt) изображен на Рис. 2.

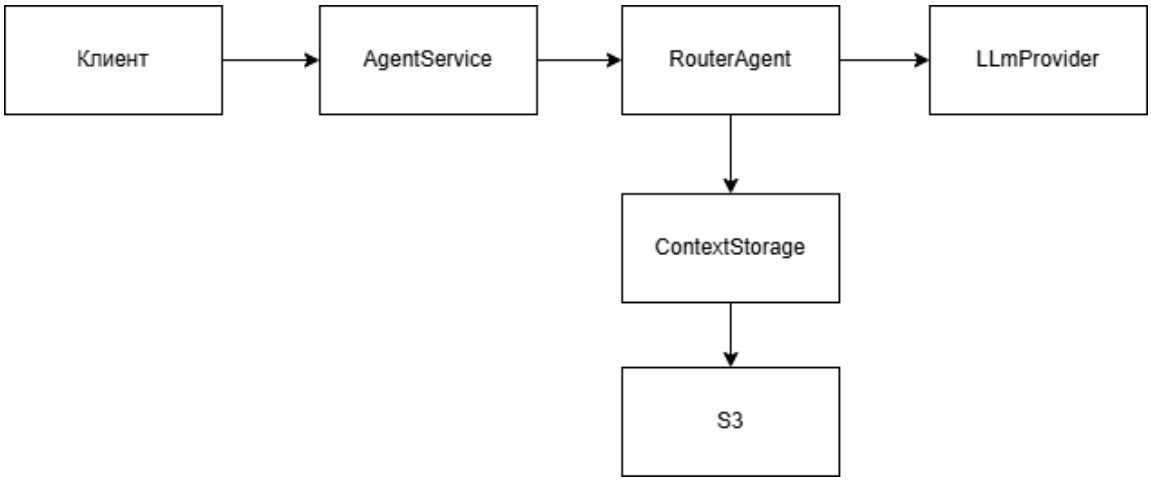


Рис. 2. Схема потоков данных чата.

2. Поток аутентификации изображен на Рис. 3.

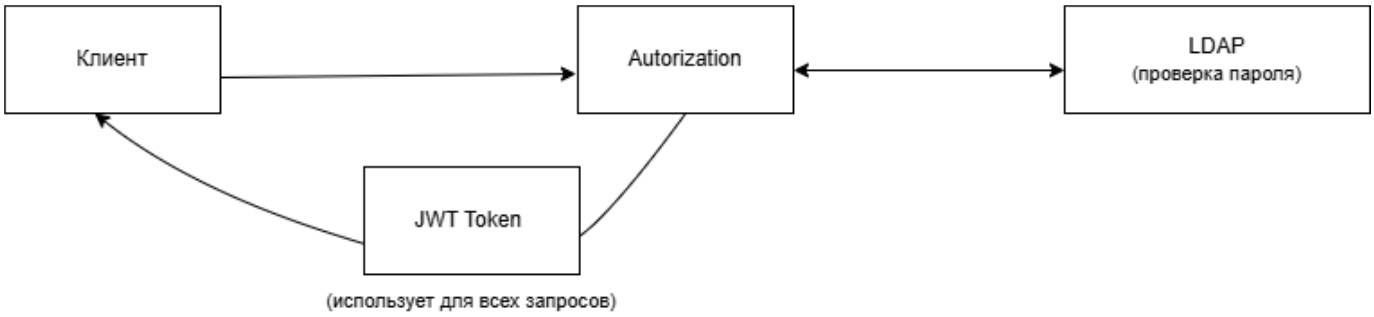


Рис. 3. Схема потоков данных аутентификации.

3. Поток управления агентами изображен на Рис. 4.

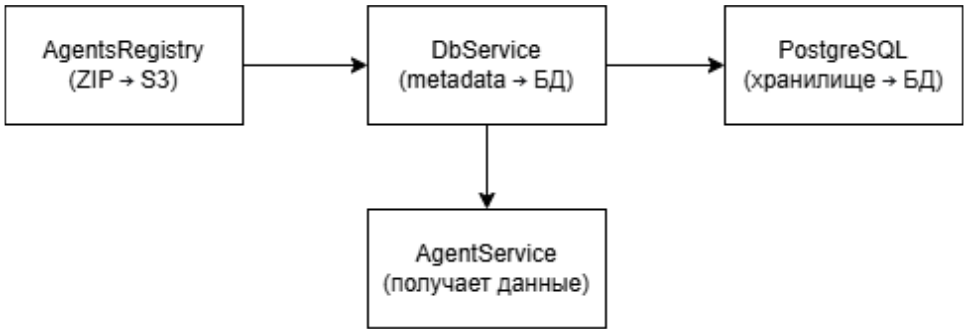


Рис. 4. Схема потоков данных управления агентами.

4. Поток обработки совещаний изображен на Рис. 5.

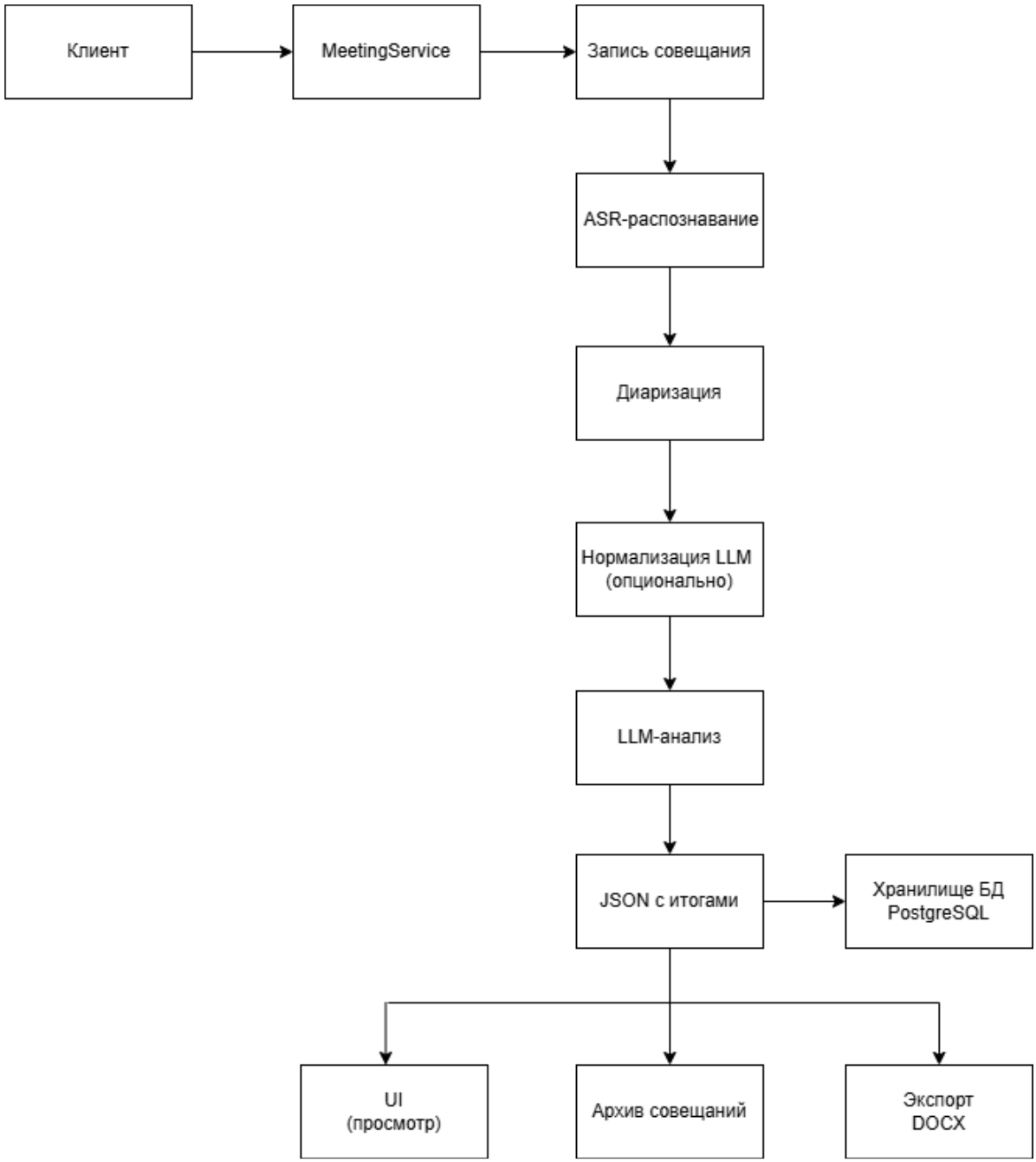


Рис. 5. Схема потоков данных обработки материалов совещаний.

5. Поток контекстных подсказок изображен на Рис. 6.

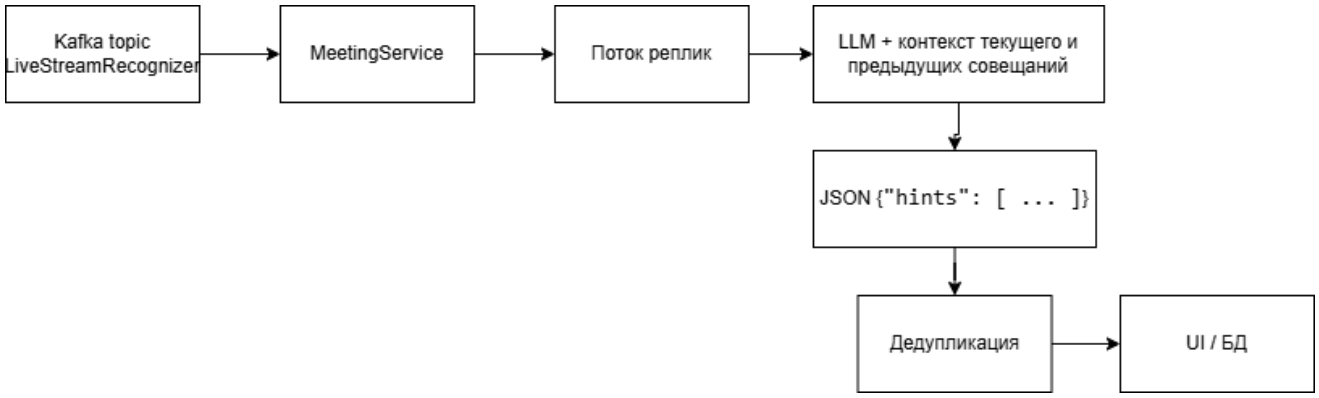


Рис. 6. Схема потоков данных контекстных подсказок.

2.9. Уровни зависимостей библиотек. Shared Libraries (слои)

Уровень 0: Common.Exceptions (без зависимостей)

Уровень 1: Common, Rest.Base

Уровень 2: Common.Runtime, Consul, Database, DocConverter, Rest.Client

Уровень 3: GigaChat, Rest.Server, Settings

Уровень 4: OpenSearch, Auth, Rest.Data, Mapping

Уровень 5: Rest.Custom, S3

2.10. Контекстная диаграмма

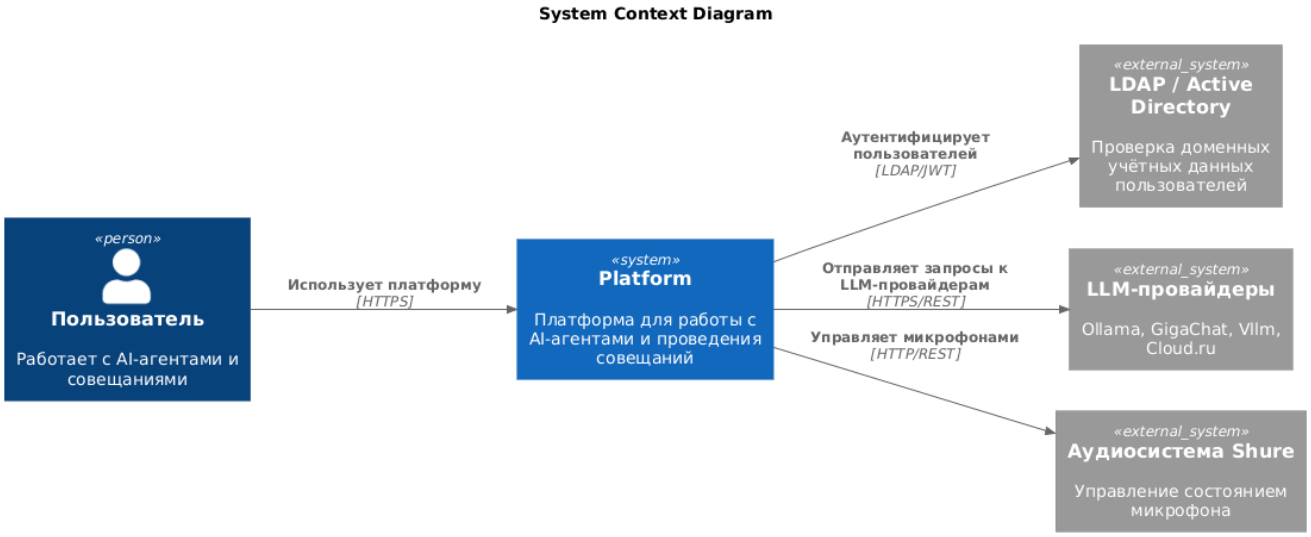


Рис 7. Контекстная диаграмма Модуля.

2.11. Контейнерная диаграмма

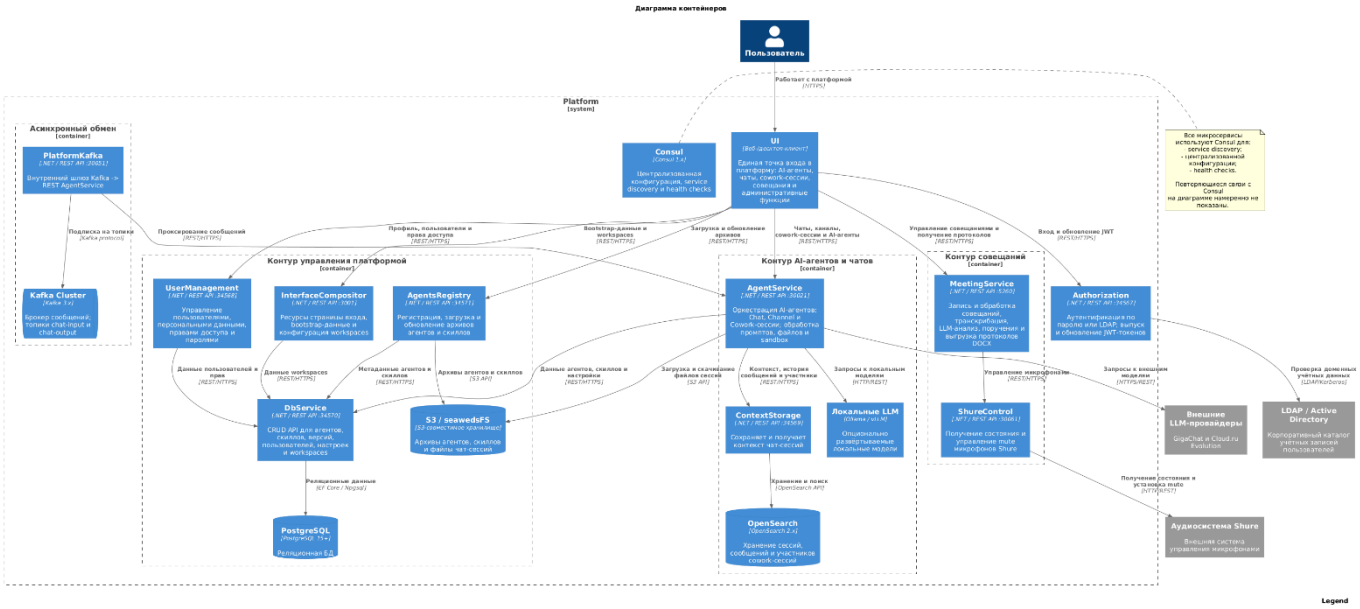


Рис. 8. Контейнерная диаграмма Модуля.

2.12. Модель данных

Модуль использует три основных хранилища:

PostgreSQL (через EF Core / DbService) – реляционные данные

OpenSearch (через ContextStorage) – документо-ориентированные данные чатов

S3 – файлы и архивы агентов/скиллов

2.12.1. Хранилище реляционных данных PostgreSQL (DbService)

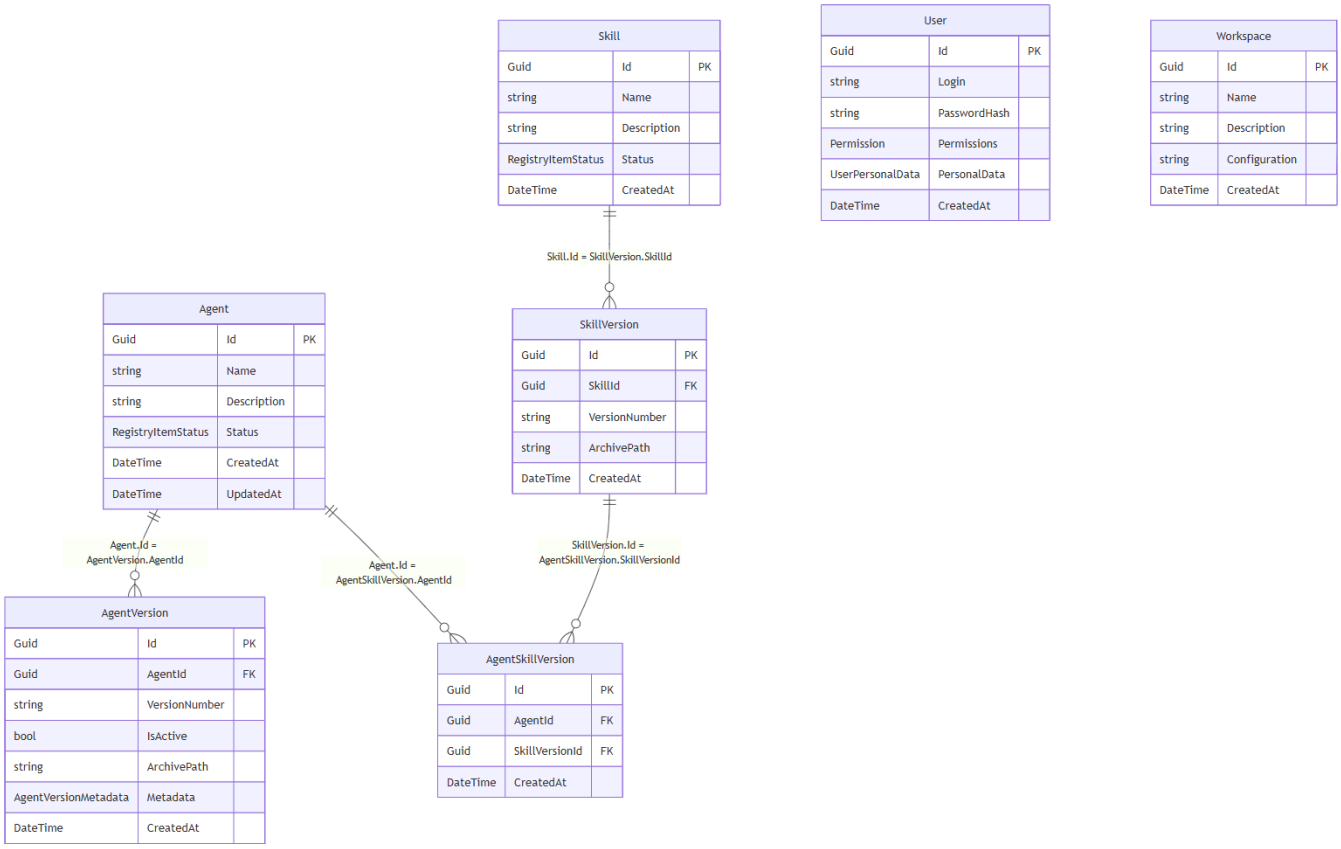


Рис. 9. Схема организации данных хранилища PostgreSQL (DbService) Модуля

Agent

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
Name	string	Название агента
Description	string	Описание
Status	RegistryItemStatus	Статус (Active/Inactive/Archived)
CreatedAt	DateTime	Дата создания
UpdatedAt	DateTime	Дата обновления

AgentVersion

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
AgentId	Guid (FK→Agent)	Родительский агент
VersionNumber	string	Номер версии (semver)
IsActive	bool	Активная ли версия
ArchivePath	string	Путь к архиву в S3
Metadata	AgentVersionMetadata	Метаданные (JSON)
CreatedAt	DateTime	Дата создания

Skill

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
Name	String	Название скилла
Description	String	Описание
Status	RegistryItemStatus	Статус
CreatedAt	DateTime	Дата создания

SkillVersion

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
SkillId	Guid (FK→Skill)	Родительский скилл
VersionNumber	String	Номер версии
ArchivePath	String	Путь к архиву в S3
CreatedAt	DateTime	Дата создания

AgentSkillVersion

Поле	Тип	Описание
Id Guid	(PK)	Уникальный идентификатор
AgentId	Guid (FK→Agent)	Агент
SkillVersionId	Guid (FK→SkillVersion)	Версия скилла
CreatedAt	DateTime	Дата привязки

User

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
Login	String	Логин
PasswordHash	String	Хеш пароля
Permissions	Permission ([Flags])	Права доступа
PersonalData	UserPersonalData	Личные данные (JSON)
CreatedAt	DateTime	Дата создания

Workspace

Поле	Тип	Описание
Id	Guid (PK)	Уникальный идентификатор
Name	string	Название
Description	string	Описание
Configuration	string (JSON)	Конфигурация workspace
CreatedAt	DateTime	Дата создания

2.12.2. Хранилище документо-ориентированных данных чатов OpenSearch (ContextStorage)

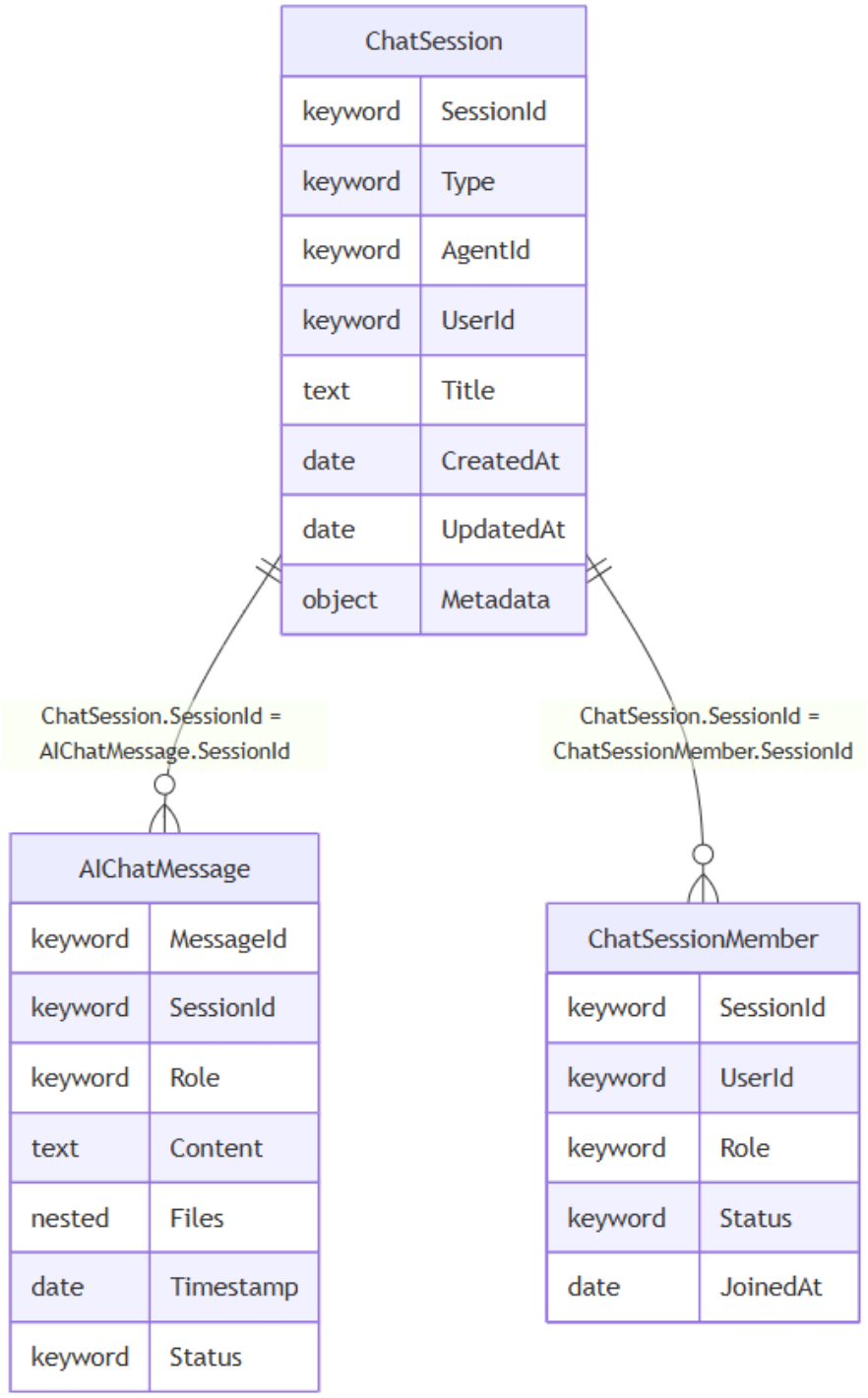


Рис. 10. Схема организации данных хранилища OpenSearch (через ContextStorage) Модуля.

ChatSession

Поле	Тип	Описание
SessionId	keyword	ID сессии
Type	keyword	chat, channel или cowork
AgentId	keyword	ID агента
UserId	keyword	ID создателя
Title text	Название сессии	
CreatedAt	Date	Дата создания
UpdatedAt	Date	Дата обновления
Metadata	object	Произвольные метаданные

AIChatMessage

Поле	Тип	Описание
MessageId	keyword	ID сообщения
SessionId	keyword	ID сессии
Role	keyword	user, assistant, system
Content	text	Текст сообщения
Files	nested[]	Прикреплённые файлы
Timestamp	date	Время отправки
Status	keyword	Статус (sending/sent/error)

ChatSessionMember

Поле	Тип	Описание
SessionId	keyword	ID сессии
UserId	keyword	ID пользователя
Role	keyword	owner, member, viewer
Status	keyword	active, removed
JoinedAt	date	Время добавления

2.12.3. Хранилище S3 файлов и архивов агентов/скиллов S3 (Объектное хранилище)

Структура ключей:

/agents/{agentId}/versions/{versionId}/archive.zip

/skills/{skillId}/versions/{versionId}/archive.zip

/files/{sessionId}/{fileId}/{filename}

Метаданные (при загрузке файлов):

- ContentType
- OriginalFileName
- SessionId
- UploadedBy
- UploadedAt

Бакеты:

- **agents** – архивы агентов
- **skills** – архивы скиллов
- **files** – файлы из чатов

2.12.4. Общая схема взаимодействия хранилищ

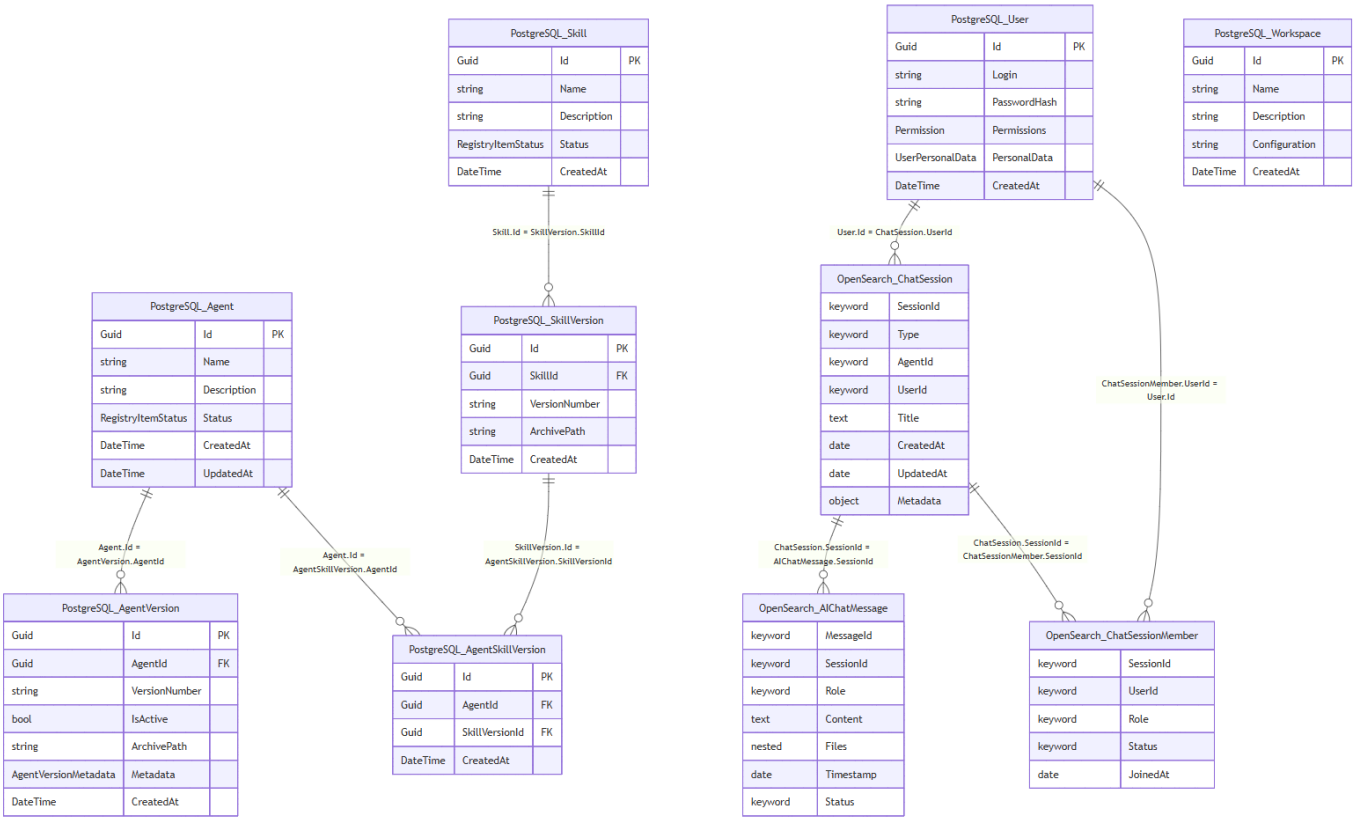


Рис. 11. Общая схема взаимодействия хранилищ Модуля.

2.13. Взаимодействие Модуля и системы «AgentNavigator»

Взаимодействие Модуля и системы "AgentNavigator" осуществляется по протоколу MCP для вызова доступных методов и получения результатов их выполнения.

3. ОПИСАНИЕ API МОДУЛЯ

3.1. Сводная таблица микросервисов API

Сводная таблица микросервисов API приведена в Табл. 4.

Табл. 4. Сводная таблица микросервисов API

Сервис	Базовый URL	Всего эндпоинтов
AgentService	:30021/api/v1	34
AgentsRegistry	:34571/api/v1	4
Authorization	:34567/api/v1	4
ContextStorage	:34569/api/v1	15
DbService	:34570/api/v1	42
InterfaceCompositor	:3001/api/v1	5
MeetingSrevice	:5260/api/v1	21
PlatformKafka	:30051	0 (Kafka gateway)
ShureControl	:30061/api/v1	3
UserManagement	:34568/api/v1	9

3.2. AgentService API

Базовый URL: http://<host>:30021/api/v1

Аутентификация: JWT Bearer Token (требуется для всех эндпоинтов, кроме #if DEBUG)

Перечень API запросов к микросервису Chat Controller приведен в Табл. 5.

Табл. 5. Перечень запросов API к микросервису Chat Controller

Метод	Путь	Описание	Permission
GET	/api/v1/GetAgents	Получить список доступных агентов	Login
POST	/api/v1/ChatPrompt	Отправить промпт в чат (multipart/form-data)	Login
POST	/api/v1/ChatExtPrompt	Отправить промпт с внешними файлами (JSON)	Login
GET	/api/v1/Chat/List	Получить список всех чат-сессий (без истории)	Login
GET	/api/v1/Chat/List/{agentId}	Получить список чат-сессий для конкретного агента	Login
PUT	/api/v1/Chat	Создать новую чат-сессию	Login
GET	/api/v1/Chat/{sessionId}	Получить чат-сессию с полной историей	Login
DELETE	/api/v1/Chat/{sessionId}	Удалить чат-сессию	Login
PATCH	/api/v1/Chat/{sessionId}/Rename	Переименовать чат-сессию	Login
POST	/api/v1/Chat/DownloadFile	Скачать файл из чат-сессии	Login

Табл. 6. Channel Controller

Метод	Путь	Описание
POST	/api/v1/ChannelPrompt	Отправить промпт в канал
POST	/api/v1/ChannelExtPrompt	Промпт с внешними файлами
GET	/api/v1/Channel/List	Список каналов
GET	/api/v1/Channel/List/{agentId}	Каналы по агенту
GET	/api/v1/Channel/List/Router	Получить/создать Router-канал
PUT	/api/v1/Channel	Создать канал
GET	/api/v1/Channel/{sessionId}	Получить канал
DELETE	/api/v1/Channel/{sessionId}	Удалить канал
PATCH	/api/v1/Channel/{sessionId}/Rename	Переименовать
POST	/api/v1/Channel/DownloadFile	Скачать файл

Табл. 7. Cowork Controller

Метод	Путь	Описание
POST	/api/v1/CoworkPrompt	Отправить промпт
POST	/api/v1/CoworkExtPrompt	Промпт с внешними файлами
GET	/api/v1/Cowork/List	Список cowork-сессий
GET	/api/v1/Cowork/List/{agentId}	Список cowork-сессий по агенту
PUT	/api/v1/Cowork	Создать cowork
GET	/api/v1/Cowork/{sessionId}	Получить
DELETE	/api/v1/Cowork/{sessionId}	Удалить
PATCH	/api/v1/Cowork/{sessionId}/Rename	Переименовать
POST	/api/v1/Cowork/DownloadFile	Скачать файл
PUT	/api/v1/Cowork/{sessionId}/Members	Добавить участника
GET	/api/v1/Cowork/{sessionId}/Members	Участники
DELETE	/api/v1/Cowork/{sessionId}/Members	Удалить участника
PUT	/api/v1/Cowork/{sessionId}/Agents	Добавить агента
GET	/api/v1/Cowork/{sessionId}/Agents	Агенты в сессии
DELETE	/api/v1/Cowork/{sessionId}/Agents	Удалить агента

Табл. 8. Direct LLM

Метод	Путь	Описание	Permission
POST	/api/v1/DirectLlmPrompt	Отправить прямой запрос к LLM без сохранения контекста	Login

Табл. 9. Debug (Только в DEBUG-сборке.)

Метод	Путь	Описание
POST	api/DEBUG/SimpleChat	Простой чат через GigaChat
POST	api/DEBUG/FileUpload	Загрузка файла в GigaChat
GET	/api/DEBUG/some	Тестовый эндпоинт

3.3. ContextStorage API

Базовый URL: http://<host>:34569/api/v

Все эндпоинты – POST, принимают JSON.

Табл. 10. Сессии

Метод	Путь	Описание	Permission
POST	/api/v1/GetUserSessions	Получить сессии пользователя по фильтру	Login
POST	/api/v1/GetSessionHistory	Получить историю сессии	Login
POST	/api/v1/GetSessionMetadata	Получить метаданные сессии (без сообщений)	Login
POST	/api/v1/CreateSession	Создать новую сессию	Login
POST	/api/v1/UpdateSession	Обновить данные сессии	Login
POST	/api/v1/DeleteSession	Удалить сессию	Login

Табл. 11. Сообщения

Метод	Путь	Описание	Permission
POST	/api/v1/StoreSessionMessages	Сохранить сообщения AI-чата	Login
POST	/api/v1/GetSessionMessages	Получить сырые сообщения	–

Табл. 12. Участники (Cowork)

Метод	Путь	Описание	Permission
POST	/api/v1/GetSessionMembers	Получить участников сессии	Login
POST	/api/v1/AddSessionMember	Добавить участника	Login
POST	/api/v1/RemoveSessionMember	Удалить участника	Login
POST	/api/v1/AddAgentToSession	Добавить агента в сессию	Login
POST	/api/v1/GetSessionAgents	Получить агентов в сессии	Login
POST	/api/v1/RemoveAgentFromSession	Удалить агента из сессии	Login

3.4. MeetingService API

Табл. 13. MeetingsController (/api/v1/meetings) — 21 эндпоинт

Метод	Путь	Описание
POST	start	Создать совещание
POST	pause	Приостановить
POST	resume	Возобновить
POST	finish	Завершить
POST	restart	Перезапустить обработку
POST	get-meeting	Получить совещание
POST	get-paged	Получить список (пагинация)
PATCH	set-agenda	Обновить повестку
GET	list-assigned-persons	Участники совещания
PUT	assign-person	Назначить участника ответственным за поручение
DELETE	unassign-person	Открепить участника от ответственности за поручение
GET	list-tasks	Список поручений
POST	create-task	Создать поручений
PATCH	update-task	Обновить поручение
DELETE	task	Удалить поручение
POST	get-hints-paged	Хинты (пагинация)
GET	get-raw-transcript	Сырой транскрипт
GET	get-normalized-transcript	Нормализованный транскрипт
POST	get-doc	Документ протокола
POST	get-doc-short	Краткий документ
POST	get-doc-transcript	Документ с транскриптом

Табл. 14. PersonsController (/api/v1/persons) — 7 эндпоинтов

Метод	Путь	Описание
POST	list	Список участников (пагинация)
POST	by-meeting	Участники по ID совещания
GET	(root)	Получить участника по ID
POST	(root)	Создать участника
PUT	import	Импорт участников из файла
PATCH	(root)	Обновить участника
DELETE	(root)	Удалить участника

Табл. 15. FileProcessingController (/api/v1/file-processing) — 4 эндпоинта

Метод	Путь	Описание
POST	upload	Загрузить аудио/видео
GET	result/{taskId}	Результат обработки
GET	{callId}/audio	Получить аудио
POST	store-persons	Сохранить файл список участников

Табл. 16. LlmSettingsController (/api/v1/llm-settings) — 2 эндпоинта

Метод	Путь	Описание
GET	(root)	Получить каталог LLM
POST	llm	Установить LLM

Табл. 17. PreviewController (/api/v1/preview) — 3 эндпоинта

Метод	Путь	Описание
POST	file-text	Предпросмотр текста файла
POST	file-agenda	Предпросмотр повестки
POST	file-persons	Предпросмотр участников

Табл. 18. Дополнительные системные эндпоинты (/) — 2 эндпоинта

Метод	Путь	Описание
GET	/version	Версия сервиса
GET	/health	Health check

Всего: 39 эндпоинтов.

PUT /api/v1/Meetings/start Body:

```
{
    "meetingId": "guid"
}
```

POST /api/v1/Meetings/list Body:

```
{
    "offset": 0,
    "limit": 50
}
```

POST /api/v1/Meetings/list Response:

```
{
    "meetings": [
        {
            "id": "guid",
            "state": "Recording | RecordingPaused | Processing | Succeeded | Failed",
            "startTime": "ISO8601",
            "finishTime": "ISO8601",
            "summary": "string",
            "error": "string"
        }
    ]
}
```

PATCH /api/v1/Meetings/update/{meetingId} Body:

```
{
    "questions": "string (optional)",
    "summary": "string (optional)"
}
```

PUT /api/v1/Meetings/create-task/{meetingId} Body:

```
{
    "description": "string",
    "assigneeId": "guid (optional)",
    "dueDate": "ISO8601 (optional)"
}
```

Табл. 19. Управление ответственными (PersonsController)

Метод	Путь	Описание
GET	/api/v1/Persons/list	Получить список ответственных
GET	/api/v1/Persons/get/{personId}	Получить данные ответственного
PUT	/api/v1/Persons/create	Создать запись ответственного
PATCH	/api/v1/Persons/update/{personId}	Обновить данные ответственного
DELETE	/api/v1/Persons/delete/{personId}	Удалить запись ответственного

PUT /api/v1/Persons/create Body:

```
{
    "lastName": "string",
    "firstName": "string",
    "middleName": "string"
}
```

3.5. DbService API

Базовый URL: http://<host>:34570/api/v1

Методы DbService предназначены для внутреннего взаимодействия сервисов платформы. Внешний API управления агентами, навыками, версиями, связями и пользовательскими доступами предоставляется сервисом AgentsRegistry.

Табл. 20. Settings

Метод	Путь	Описание	Permission
POST	/api/v1/Settings	Получить настройки по типу/ID	Login
GET	/api/v1/Settings/GetUser	Получить настройки текущего пользователя	Login
POST	/api/v1/Settings/SetUser	Установить настройки текущего пользователя	Login
GET	/api/v1/Settings/ConnectionAddresses	Получить адреса подключения	-
GET	/api/v1/Settings/Global	Получить глобальные настройки	-
POST	/api/v1/Settings/GetService	Получить сервисные настройки по типу	Login

Табл. 21. Users

Метод	Путь	Описание	Permission
POST	/api/v1/User/Create	Создать пользователя	Admin
GET	/api/v1/User/List	Список пользователей	Admin
POST	/api/v1/User/GetByIds	Получить по ID	Admin
POST	/api/v1/User/GetByLogin	Получить по логину	Admin
POST	/api/v1/User/Update	Обновить пользователя	Admin
POST	/api/v1/User/ChangePassword	Сменить пароль	Admin
GET	/api/v1/User/GetPersonalData	Личные данные	Admin
POST	/api/v1/User/SetPersonalData	Обновить личные данные	Admin
POST	/api/v1/User/Delete	Удалить пользователя	Admin
POST	/api/v1/User/Login	Аутентификация	Admin
GET	/api/v1/User/ResetAdminPassword	Сброс пароля Администратора	Login

Табл. 22. Agents

Метод	Путь	Описание	Permission
POST	/api/v1/GetAgents	Список агентов (с фильтром, пагинация)	Login
POST	/api/v1/GetAgentById	Агент по ID	Login
GET	/api/v1/GetAgentByIdWithLastVersion	Агент с последней версией	Login
POST	/api/v1/CreateAgent	Создать агента	Login
POST	/api/v1/UpdateAgent	Обновить агента	Login
POST	/api/v1/ChangeAgentStatusById	Изменить статус	Login

Табл. 23. Skills

Метод	Путь	Описание	Permission
POST	/api/v1/GetSkills	Список скиллов	Login
POST	/api/v1/GetSkillById	Скилл по ID	Login
GET	/api/v1/GetSkillByIdWithLastVersion	Скилл с последней версией	Login
POST	/api/v1/CreateSkill	Создать скилл	Login
POST	/api/v1/UpdateSkill	Обновить скилл	Login
POST	/api/v1/ChangeSkillStatusById	Изменить статус	Login

Табл. 24. Agent Versions

Метод	Путь	Описание	Permission
POST	api/v1/GetAgentVersionsByAgentId	Версии агента	Login
POST	/api/v1/GetAgentVersionByVersionId	Конкретная версия	Login
POST	/api/v1/CreateAgentVersion	Создать версию	Login
POST	/api/v1/ActivateAgentVersion	Активировать версию	Login

Табл. 25. Skill Versions

Метод	Путь	Описание	Permission
POST	/api/v1/GetSkillVersionsBySkillId	Версии скилла	Login
POST	/api/v1/GetSkillVersionByVersionId	Конкретная версия	Login
POST	/api/v1/CreateSkillVersion	Создать версию	Login

Табл. 26. Agent-Skill Binding

Метод	Путь	Описание	Permission
POST	/api/v1/GetAgentSkillVersionsByAgentId	Скиллы агента	Login
POST	/api/v1/AddSkillToAgent	Привязать скилл	Login
POST	/api/v1/DeleteSkillFromAgent	Отвязать скилл	Login
POST	/api/v1/UpdateSkillVersionInAgent	Обновить версию скилла	Login

Табл. 27. Workspaces

Метод	Путь	Описание	Permission
GET	/api/v1/Workspace/ListAll	Все workspace	Login
POST	/api/v1/Workspace/Get	Workspace по ID	Login

3.6. Authorization API

Базовый URL: http://<host>:34567/api/v1

Табл. 28. Login

Метод	Путь	Описание
POST	/api/v1/Login	Вход по паролю (без LDAP)
POST	/api/v1/LoginDomain	Вход через LDAP/домен

POST/api/v1/Login Body:

```
{
    "login": "string",
    "passwordBase64": "string"
}
```

Ответ:

```
{
    "data": {
        "accessToken": "string (JWT)",
        "refreshToken": "string",
        "expiresIn": 3600,
        "userInfo": {
            "id": "guid",
            "login": "string",
            "permissions": "Login"
        }
    }
}
```

Табл. 29. Token

Метод	Путь	Описание
POST	/api/v1/RefreshToken	Обновить JWT-токен
GET	api/v1/ServiceToken	Получить сервисный токен (только для доверенных IP)

3.7. AgentsRegistry API

Базовый URL: http://<host>:34571/api/v1

Права доступа: если в таблице указано Админ, endpoint доступен только администраторам.

Авторизованный пользователь — любой пользователь, прошедший аутентификацию.

AgentsRegistry предоставляет API для управления агентами, навыками, версиями агентов и навыков, связями между агентами и навыками, а также назначением агентов пользователям.

AgentsController

Управление агентами. Контроллер отвечает за создание, изменение, поиск и версионирование агентов.

Base path: /api/v1/agents

Табл. 30. AgentsController

Операция	HTTP	Endpoint	Доступ	Описание
Поиск агентов	POST	/api/v1/agents/search	Админ	Возвращает список агентов с пагинацией, поиском по названию и фильтрами по статусу и категории
Получить агента	GET	/api/v1/agents/by-id?agentId=	Админ	Возвращает агента по идентификатору
Получить агента с последней версией	GET	/api/v1/agents/by-id/with-lastversion?agentId=	Админ	Возвращает агента вместе с его последней версией
Создать агента	POST	/api/v1/agents/create	Админ	Создаёт агента и его первую версию из JSON-модели. Архив не используется
Обновить агента	PUT	/api/v1/agents/update	Админ	Обновляет название, описание и правила агента
Изменить статус	POST	/api/v1/agents/change-status	Админ	Изменяет статус агента: 0 — отключён, 1 — активен, 2 — в архиве
Создать из архива	POST	/api/v1/agents/create-fromarchive	Админ	Принимает ZIP до 50 МБ, читает и валидирует manifest.json, загружает архив в S3 и создаёт агента с первой версией
Обновить из архива	PUT	/api/v1/agents/update-fromarchive?agentId=	Админ	Принимает ZIP-архив и сравнивает его с текущей версией. При изменении настроек или содержимого создаётся новая версия

Логика обновления агента из архива

При вызове:

PUT /api/v1/agents/update-from-archive?agentId=

система сравнивает новый архив с текущей версией агента.

Новая версия создаётся, если изменились:

- настройки LLM;
- описание;
- содержимое архива.

Новая версия не создаётся, если изменились только:

- название;
- описание агента;
- другие поля «шапки» агента.

В последнем случае обновляется существующая запись агента.

Agents Versions

Base path: /api/v1/agents/versions

Табл. 31. Agents Versions

Операция	HTTP	Endpoint	Доступ	Описание
Получить список версий	GET	/api/v1/agents/versions?agentId=	Админ	Возвращает все версии указанного агента
Получить версию	GET	/api/v1/agents/versions/by-id?agentVersionId=	Админ	Возвращает конкретную версию агента
Создать версию	POST	/api/v1/agents/versions/create	Админ	Создаёт новую версию агента. Номер версии вычисляется автоматически на основании VersionType
Активировать версию	POST	/api/v1/agents/versions/activate	Админ	Делает указанную версию активной и записывает её в ActiveVersionId

Привязка навыков к агенту

Base path: /api/v1/agents

Табл. 32. Привязка навыков к агенту

Операция	HTTP	Endpoint	Доступ	Описание
Получить навыки агента	GET	/api/v1/agents/agent-skill-versions? agentId=	Админ	Возвращает список навыков, привязанных к агенту, включая используемые версии
Добавить навык	POST	/api/v1/agents/add-skill	Админ	Привязывает навык к агенту с указанием версии навыка
Удалить навык	POST	/api/v1/agents/delete-skill	Админ	Отвязывает навык от агента
Изменить версию навыка	PUT	/api/v1/agents/update-skill-version	Админ	Заменяет используемую версию навыка на другую

Важно: изменение навыков создаёт новую версию агента

Любое изменение состава навыков агента автоматически приводит к созданию **новой минорной версии агента**.

Это относится к операциям:

- добавление навыка;
- удаление навыка;
- изменение версии навыка.

При этом:

- создаётся новая Minor-версия агента;
- фиксируется изменение в AgentChangesHistory;
- новая версия содержит актуальный состав навыков.

Таким образом, предыдущую версию агента можно использовать для восстановления его прежнего состава навыков.

SkillsController

Управление навыками

Base path: /api/v1/skills

Табл. 33. Управление навыками

Операция	HTTP	Endpoint	Доступ	Описание
Поиск навыков	POST	/api/v1/skills/search	Админ	Возвращает список навыков с пагинацией и фильтрами
Получить навык	GET	/api/v1/skills/by-id?skillId=	Админ	Возвращает навык по идентификатору
Получить навык с последней версией	GET	/api/v1/skills/by-id/with-lastversion?skillId=	Админ	Возвращает навык вместе с его последней версией
Создать навык	POST	/api/v1/skills/create	Админ	Создаёт навык и его первую версию из JSON-модели
Обновить навык	PUT	/api/v1/skills/update	Админ	Обновляет название, описание и правила навыка
Изменить статус	POST	/api/v1/skills/change-status	Админ	Изменяет статус навыка
Создать из архива	POST	/api/v1/skills/create-from-archive	Админ	Принимает ZIP до 300 МБ, валидирует manifest.json и создаёт навык с первой версией
Обновить из архива	PUT	/api/v1/skills/update-fromarchive?skillId=	Админ	Сравнивает архив с текущей версией. Если содержимое не изменилось, новая версия не создаётся

Skills Versions

Base path: /api/v1/skills/versions

Табл. 34. Skills Versions

Операция	HTTP	Endpoint	Доступ	Описание
Получить список версий	GET	/api/v1/skills/versions?skillId=	Админ	Возвращает все версии указанного навыка
Получить версию	GET	/api/v1/skills/versions/by-id?skillVersionId=	Админ	Возвращает конкретную версию навыка
Создать версию	POST	/api/v1/skills/versions/create	Админ	Создаёт новую версию навыка

UserAgentsController

Назначение агентов пользователям

Base path: /api/v1/user-agents

Табл. 35. UserAgentsController

Операция	HTTP	Endpoint	Доступ	Описание
Мои агенты	GET	/api/v1/user-agents/available-forcurrent-user	Авторизованный пользователь	Возвращает список агентов, назначенных текущему пользователю
Агенты пользователя	GET	/api/v1/user-agents/available?userId=	Админ	Возвращает список агентов указанного пользователя
Назначить агента	POST	/api/v1/user-agents/assign	Админ	Даёт пользователю доступ к агенту. Активная версия устанавливается автоматически
Отвязать агента	DELETE	/api/v1/user-agents/unassign	Админ	Забирает у пользователя доступ к агенту
Выбрать версию	PATCH	/api/v1/user-agents/set-activeversion	Админ	Устанавливает версию агента, которую будет использовать пользователь
Добавить в избранное	PATCH	/api/v1/user-agents/add-favorite?agentId=	Авторизованный пользователь	Добавляет агента в избранное
Удалить из избранного	PATCH	/api/v1/user-agents/removefavorite?agentId=	Авторизованный пользователь	Убирает агента из избранного

3.8. InterfaceCompositor API

Базовый URL: http://<host>:3001/api/v1

Табл. 36. InterfaceCompositor API

Метод	Путь	Описание	Permission
GET	/api/v1/LoginResources	Ресурсы для страницы логина	-
GET	/api/v1/BootStrap	Bootstrap-данные для UI	Login
GET	api/v1/Workspaces	Список workspace	Login
POST	/api/v1/Workspace	Получить workspace по ID	Login
GET	/api/v1/Reload	Принудительная перезагрузка композиции	Login

3.9. ShureControl API

Базовый URL: http://<host>:30061/api/v1

Без аутентификации.

Табл. 37. ShureControl API

Метод	Путь	Описание
GET	/api/v1/GetButtonPressState	Состояние кнопки Shure
POST	/api/v1/SetMute	Установить mute микрофона
GET	/api/v1/GetButtonAvailableState	Доступность кнопки

POST /api/v1/SetMute Body:

```
{
    "mute": true
}
```

3.10. UserManagement API

Базовый URL: http://<host>:34568/api/v1

Табл. 38. UserManagement API

Метод	Путь	Описание	Permission
POST	/api/v1/CreateUser	Создать пользователя	Admin
GET	/api/v1/ListUsers	Список пользователей	Login
POST	/api/v1/GetUsers	Получить по ID	Login
POST	/api/v1/GetPersonalData	Личные данные (свои или userId)	Login
POST	/api/v1/SetPersonalData	Обновить личные данные	Login
POST	/api/v1/ChangeUserPermissions	Изменить права	Admin
POST	/api/v1/ChangeUserLogin	Сменить логин	Admin
POST	/api/v1/ChangeSelfPassword	Сменить свой пароль	Login
POST	/api/v1/ChangeOtherPassword	Сменить чужой пароль	Admin

3.11. PlatformKafka

Базовый URL: http://<host>:30051

Не имеет собственных REST-эндпоинтов. Работает как Kafka-шлюз:

- Подписывается на Kafka-топики
- Проксирует сообщения в AgentService через REST
- Использует AgentServiceRestPoint для отправки ChatExtPrompt.

4. ВЗАИМОДЕЙСТВИЕ АДМИНИСТРАТОРА/ПОЛЬЗОВАТЕЛЯ С API МОДУЛЯ

4.1. Аутентификация

Большинство эндпоинтов защищены JWT-токеном:

Authorization: Bearer <token>

Уровни доступа (enum Permission):

- **None** – нет никаких прав
- **Login** – любой аутентифицированный пользователь
- **Permission1–Permission5** – дополнительные уровни
- **Admin** – администратор
- **All** – все права

4.2. Формат ответов

Все REST-ответы следуют единому формату:

```
{
"data": { ... },
"error": {
    "text": "Сообщение",
    "debugText": "Детали (только debug)"
    }
}
```

Используется DTO: BaseDto<T> из Rest.Base.

4.3. Коды HTTP

Ответы Модуля в формате HTTP на запросы API приведены Табл. 39.

Табл. 39. Ответы Модуля в формате HTTP на запросы API

Код	Описание
200	Успешный ответ
400	Неверные параметры
401	Отсутствует или недействителен JWT
403	Недостаточно прав
404	Ресурс не найден
500	Внутренняя ошибка сервера

5. СИСТЕМНЫЕ ТРЕБОВАНИЯ

5.1. Требования к аппаратному обеспечению Модуля

Требования к аппаратному обеспечению для 1000 одновременных активных пользователей:

- Операционная система - AstraLinux SE 1.8 Воронеж или выше.
- Количество инстансов – 69.
- Необходимое количество ядер ЦП – 568.
- Рабочий объём оперативной памяти – 2 392 Гб.
- Тип хранилища – SSD.
- Рабочий объём хранилища – 17 880 Гб.

5.2. Требования к программному обеспечению Модуля

Технические компоненты в составе Модуля перечислены в Табл. 40. Полный перечень сторонних компонентов ПО Модуля приведен в Приложении 1.

Табл. 40. Технические компоненты в составе Модуля

Компонент	Версия	Назначение
.NET SDK	10.0	Runtime для микросервисов
PostgreSQL	15+	Реляционная БД для DbService
OpenSearch	2.x	Поисковый движок для ContextStorage
Consul	1.x	Service Discovery и конфигурация
Kafka	3.x	Брокер сообщений
S3 (MinIO / AWS S3)	-	Объектное хранилище
Docker	24+	Песочницы в AgentService (опционально)
Ollama / vLLM	-	Локальные LLM (опционально)

Перечень портов, используемых микросервисами и окружением Модуля приведен в Табл. 41.

Табл. 41. Порты, используемые микросервисами и окружением Модуля

Сервис	Порт	Протокол
AgentService	30021	HTTP/REST
AgentsRegistry	34571	HTTP/REST
Authorization	34567	HTTP/REST
ContextStorage	34569	HTTP/REST
DbService	34570	HTTP/REST
InterfaceCompositor	3001	HTTP/REST
MeetingService	5260	HTTP/REST
PlatformKafka	30051	HTTP/REST
ShureControl	30061	HTTP/REST
UserManagement	34568	HTTP/REST

5.3. Требования к клиентской компоненте Модуля

Операционная система клиентской части. Одна из перечисленных:

- Windows версии 10 и выше;
- Linux (операционные системы, основанные на Debian, с ядром версии 2.6 и выше).

Веб-браузер клиентской части. Один из перечисленных:

- Google Chrome версии 91.0. и выше;
- Edge версии 92 и выше;
- Mozilla Firefox версии 113.0.2 и выше;
- Яндекс Браузер версии 24.1 и выше.

6. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ

Перед установкой программного обеспечения Модуля необходимо убедиться, что аппаратные и программные средств, на которые планируется установка, отвечают требованиям изложенным в подразделах 5.1 и 5.2 данного документа.

6.1. Шаги установки

Проверить наличие:

- Установленной программы Docker (проверить: docker --version)
- Установленной программы Docker Compose (при использовании, проверить: docker compose version)
- Прав на выполнение Docker-команд (членство в группе docker или sudo)

Шаги запуска:

- Перейти в директорию с дистрибутивом
- Выдать права на выполнение скриптов

```
chmod +x ./*.sh
```

- Загрузить Docker-образы

```
# ./load_img.sh
```

! Дождитесь успешного завершения загрузки. При необходимости проверьте наличие образов командой docker images.

- Запустить контейнеры

```
# ./run.sh
```

- Проверить доступность сервисов. Откройте в браузере https://localhost:8087 или используйте имя хоста, на котором было выполнено развёртывание.

Примечание: при использовании самоподписанного SSL-сертификата браузер может выдать предупреждение о безопасности — это ожидаемо, подтвердите переход.

7. НАСТРОЙКА КОМПОНЕНТОВ МОДУЛЯ.

7.1. Рабочее пространство «Настройки»

Рабочее пространство «Настройки» предусматривает управление реестрами агентов, навыками и инструментами, в том числе создание, изменение, просмотр и удаление, а также формирование их состава путем подключения навыков к агентам и инструментов к навыкам, централизованного управления настройками платформы, пользователями, рабочими пространствами, агентами и их навыками и обеспечивает настройку прав доступа.

Интерфейс рабочего пространства «Настройки» представлен на Рис. 12.

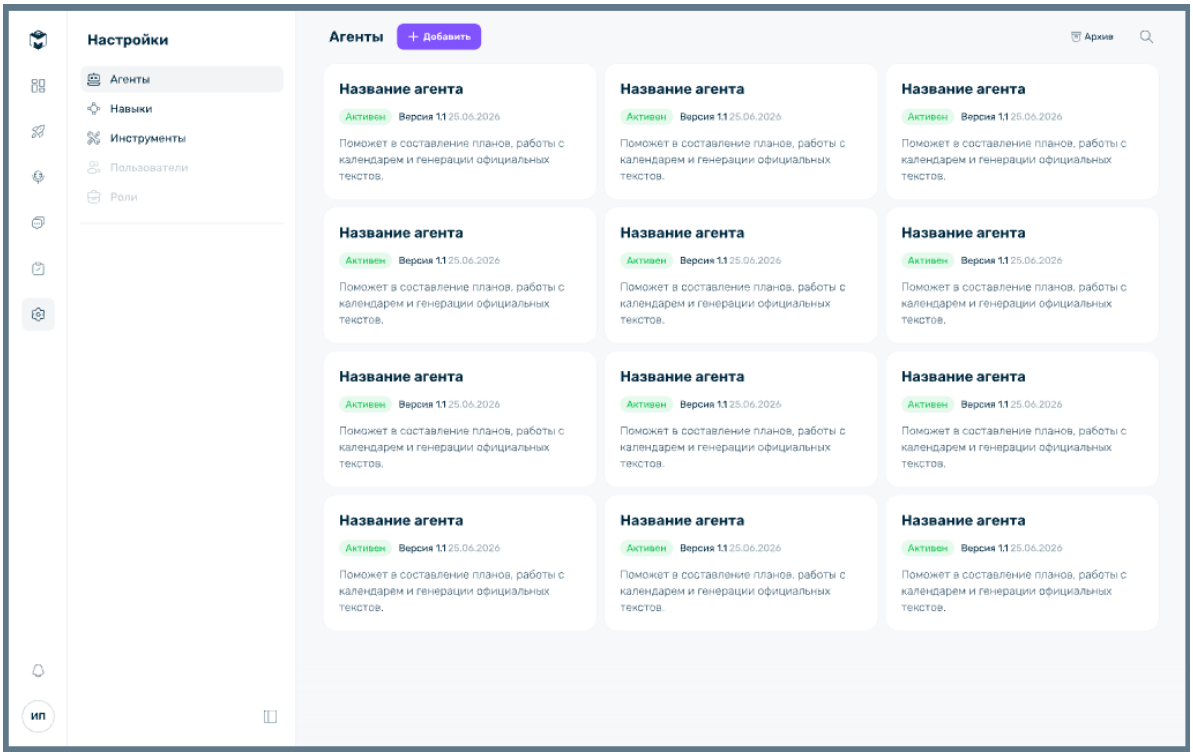


Рис. 12. Интерфейс рабочего пространства «Настройки»

Левое меню рабочего пространства «Настройки» содержит блоки для настройки:

- агентов;
- навыков;
- инструментов;
- пользователей и ролей.

7.2. Управление агентами

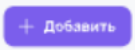
Раздел «Агенты» предназначен для централизованного управления ИИ-агентами Модуля.

Управление агентами включает:

- просмотр реестра агентов;
- создание нового агента путем загрузки пакета;
- редактирование агента с созданием новых версий;
- подключение навыков к агенту;
- публикацию версий агента;
- просмотр и переключение между версиями;
- архивацию агента.

7.2.1. Реестр агентов

Страница реестра агентов содержит:

- верхнее меню, которое содержит:
  - название раздела «Агенты»;
  - кнопку «

» для загрузки пакета агента в формате zip;
  - папку «Архив» для просмотра архивированных агентов;
  - кнопку «», предназначенную для поиска агента.
- основную область, которая содержит реестр агентов в виде мини-карточек, состоящих из:
  - названия агента;
  - статуса агента;
  - номера последней опубликованной версии с датой;
  - описания агента.

7.2.2. Карточка агентов

При нажатии на мини-карточку открывается подробная карточка агента, содержащая:

- статус агента;
- версию агента;
- дату;
- кнопку «

», предназначенную для перезагрузки пакета агента;
- отметку о рекомендованной версии;
- наименование агента;
- описание;
- инструкцию;
- список подключенных навыков;
- кнопку «
», предназначенную для добавления навыка;
- кнопку «
», расположенную рядом с наименованием навыка и предназначенную для удаления навыка;
- кнопку «
», предназначенную для переноса агента в архив;
- кнопку «Отменить», предназначенную для отмены внесенных изменений;
- кнопку «Сохранить», предназначенную для публикации новой версии агента. Кнопка становится активной только при наличии изменений;
- кнопку «
», предназначенную для закрытия окна.

Интерфейс подробной карточки агента представлен на Рис. 13.

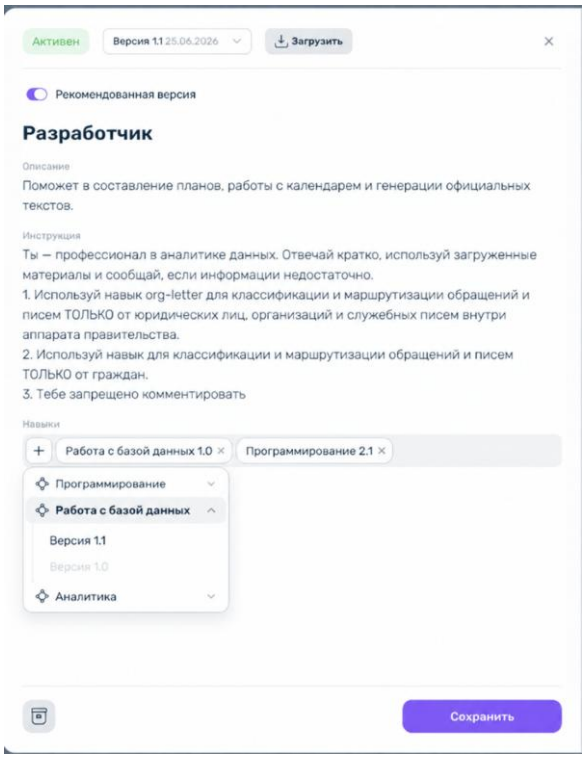


Рис. 13. Подробная карточка агента

7.2.3. Создание агента

Предусмотрена загрузка нового агента из пакета в формате zip, содержащего:

- файл с основными параметрами и конфигурацией агента;
- файл с инструкцией для языковой модели.

Интерфейс формы добавления агента представлен на Рис. 14.

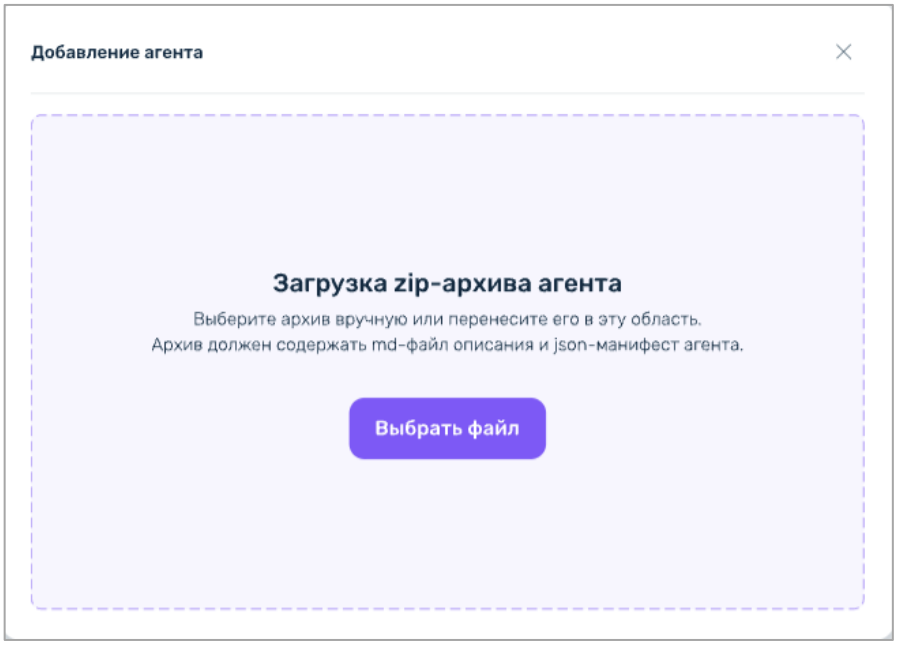


Рис. 14. Форма добавления агента

Для загрузки пакета следует выполнить следующие действия:

- В верхнем меню нажать кнопку « + Добавить », предназначенную для открытия окна загрузки.
- В окне загрузки доступна загрузка файла по кнопке «Выбрать файл» или перетаскиванием. Загрузить файл.
- В процессе обработки пакета отображается индикатор загрузки.

Предусмотрена возможность закрытия окна загрузки без сохранения внесенных изменений по кнопке « × ».

В процесс обработки также входит проверка на уникальность имени агента и тестирование работоспособности загружаемого агента.

При успешной загрузке, проверке и тестировании выдается сообщение и отображается кнопка «Перейти к агенту», предназначенная для перехода в карточку агента.

При загрузке с ошибкой выдается сообщение об ошибке с возможностью повторить загрузку, нажав кнопку «Загрузить заново».

Повторная загрузка пакета полностью заменяет содержимое текущей версии данными из пакета. Подключенные вручную навыки сохраняются.

7.2.4. Редактирование агента

Для редактирования агента необходимо открыть карточку и выбрать необходимую версию для редактирования из выпадающего списка.

Создание новой версии доступно двумя способами:

- Добавление или удаление подключенных навыков. Версия обновляется после нажатия кнопки «Сохранить».
- Перезагрузка нового пакета. Новая версия создается после загрузки, проверки и тестирования.

Для отмены всех внесенных изменений предусмотрена кнопка "Отменить", для переноса версии в архив предусмотрена кнопка «  », а для публикации предусмотрена кнопка «Сохранить».

После публикации сохраняется новая версия агента с датой, равной дате сохранения.

7.2.5. Подключение навыков к агенту

Подключение навыков доступно при первом создании и при редактировании агента, при условии их наличия и доступности.

Добавление навыков происходит по кнопке « » в карточке агента в блоке «Навыки». При нажатии появляется список доступных навыков и их версий. Если список пуст, отображается текст «Нет доступных навыков».

К агенту можно подключить только одну версию одного навыка. Количество подключенных навыков не ограничено.

7.2.6. Присвоение версий агенту

Новая версия агента присваивается при загрузке пакета или при сохранении изменений по кнопке «Сохранить». Отображение версий предусмотрено в карточке агента.

Для просмотра и редактирования предыдущей версии предусмотрен выпадающий список с возможностью переключения по версиям.

Переключатель "Рекомендованная версия" предназначен для проставления соответствующей отметки агенту. При наличии у агента данной отметки предусмотрено предложение рекомендованной версии пользователю при создании диалога с агентом.

7.2.7. Архивация агента

Удаление агента выполняется путем переноса в Архив.

Архивированный агент:

- перестает отображаться при создании новых диалогов;
- существующие диалоги сохраняют возможность просмотра истории;
- отправка новых сообщений этому агенту невозможна.

7.3. Управление навыками

Раздел «Навыки» предназначен для централизованного управления навыками агентов.

Интерфейс раздела «Навыки» представлен на рисунке Рис. 15.

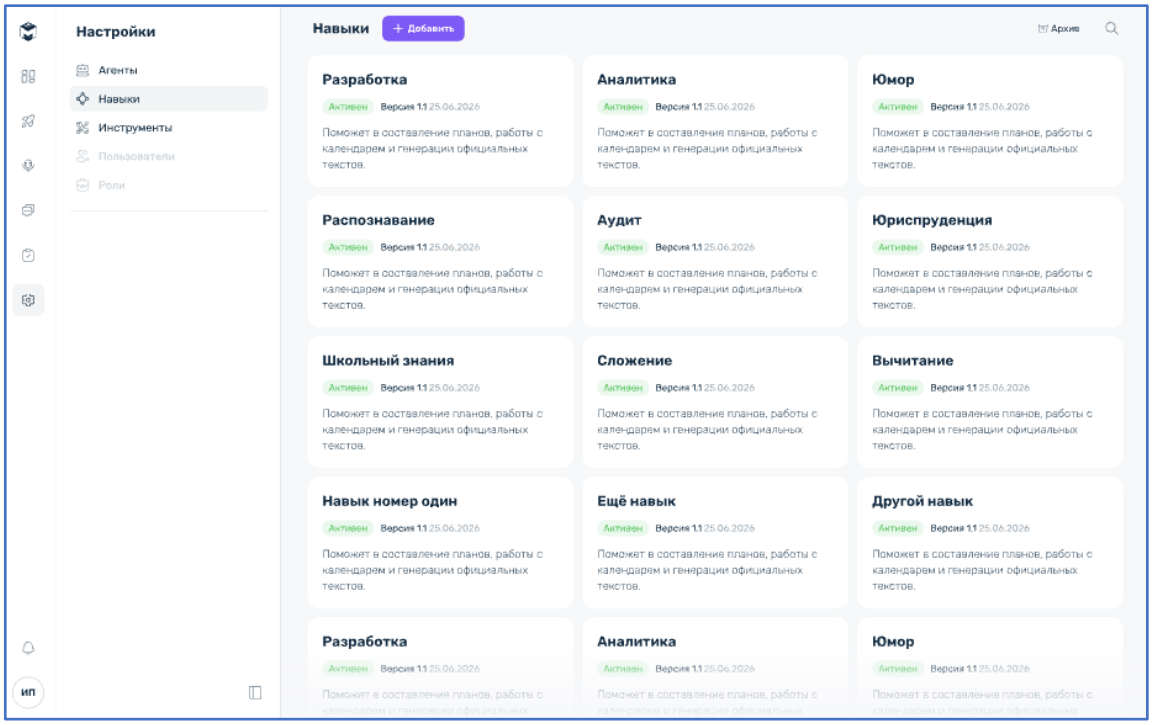


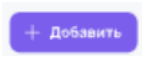
Рис. 15. Интерфейс раздела «Навыки»

Управление навыками включает:

- просмотр реестра навыков;
- создание нового навыка путем загрузки пакета формата zip;
- редактирование навыка путем создания новых версий;
- архивирование навыка;
- подключение инструментов к навыку;
- просмотр списка агентов, в которых используется навык;
- публикацию версий навыка.

7.3.1. Реестр навыков

Страница реестра навыков содержит:

- верхнее меню, которое содержит:
  - название раздела «Навыки»;
  - кнопку «

» для загрузки пакета навыка в формате zip;
  - папку «Архив» для просмотра архивированных навыков;
  - кнопку «», предназначенную для поиска навыка.
- основную область, которая содержит реестр навыков в виде мини-карточек, состоящих из:
  - названия навыка;
  - статуса навыка;
  - версии навыка с датой;
  - описания навыка.

7.3.2. Карточка навыка

При нажатии на мини-карточку открывается подробная карточка навыка, содержащая:

- версию;
- дату;
- кнопку «Загрузить», предназначенную для перезагрузки пакета навыка в формате zip;
- наименование;
- описание;
- инструкцию;
- список подключенных инструментов;
- кнопку «+», предназначенную для добавления инструмента;
- кнопку «×», расположенную рядом с наименованием инструмента и предназначенную для удаления инструмента;
- список агентов, в которых есть навык;
- кнопку «Отменить», предназначенную для отмены несохраненных изменений;
- кнопку «», предназначенную для переноса навыка в архив;
- кнопку «Сохранить», предназначенную для публикации новой версии навыка;
- кнопку закрытия окна.

Интерфейс подробной карточки навыка представлен на Рис. 16.

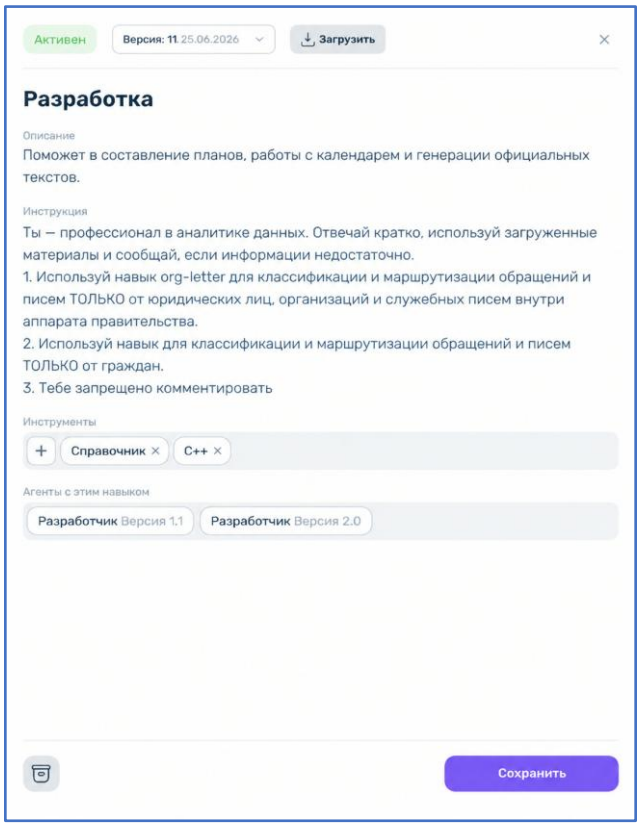


Рис. 16. Подробная карточка навыка

7.3.3. Добавление навыка

Предусмотрена загрузка нового навыка из пакета, содержащего:

- файл с основными параметрами и конфигурацией агента;
- файл с инструкцией для языковой модели.

Интерфейс формы добавления агента представлен на Рис. 17.

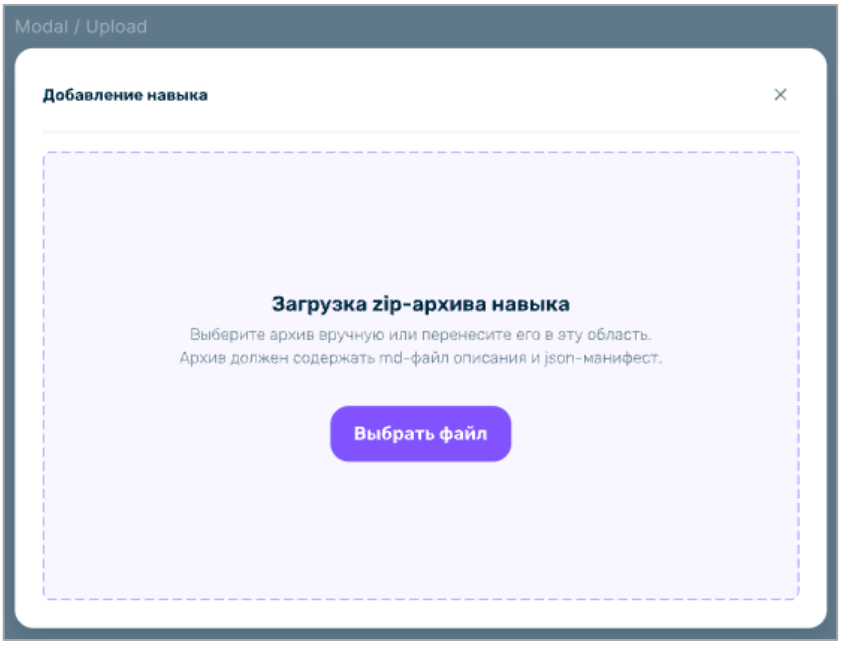


Рис. 17. Форма добавления навыка

Для загрузки пакета следует выполнить следующие действия:

- В верхнем меню нажать кнопку « + Добавить », предназначенную для открытия окна загрузки.
- В окне загрузки доступна загрузка файла по кнопке «Выбрать файл» или перетаскиванием. Загрузить файл.
- В процессе обработки пакета отображается индикатор загрузки.

Предусмотрена возможность закрытия окна загрузки без сохранения внесенных изменений по кнопке « × ».

Если файл не был загружен, то отображается сообщение об ошибке с возможностью повторить загрузку путем нажатия на кнопку «Загрузить заново».

При успешной загрузке выдается сообщение и отображается кнопка «Перейти к навыку», предназначенная для перехода в карточку навыка.

7.3.4. Редактирование навыка

Для редактирования навыка необходимо открыть карточку этого навыка и выбрать необходимую версию для редактирования из выпадающего списка.

Создание новой версии навыка доступно двумя способами:

- Добавление или удаление подключенных инструментов. Версия обновляется после нажатия кнопки «Сохранить».
- Перезагрузка нового пакета. Новая версия создается после загрузки и проверки.

Для отмены всех внесенных изменений предусмотрена кнопка «Отменить», для переноса версии в архив предусмотрена кнопка « », а для публикации предусмотрена кнопка «Сохранить».

После внесения изменений и публикации сохраняется новая версия навыка с датой, равной дате сохранения.

7.3.5. Подключение инструментов к навыку

Подключение инструментов доступно при первом создании и при редактировании навыка, при условии их наличия и доступности.

Добавление инструментов происходит по кнопке « ». При нажатии появляется список доступных инструментов. Если список пуст, то отображается текст «Нет доступных инструментов».

7.3.6. Присвоение версий навыку

Новая версия навыка присваивается при загрузке пакета или при сохранении изменений по кнопке «Сохранить». Отображение версий предусмотрено в карточке навыка.

Для просмотра и редактирования предыдущей версии предусмотрен выпадающий список с возможностью переключения по версиям.

7.4. Управление инструментами

Раздел «Инструменты» предназначен для управления инструментами навыков агентов.

Интерфейс раздела «Инструменты» представлен на Рис. 18.

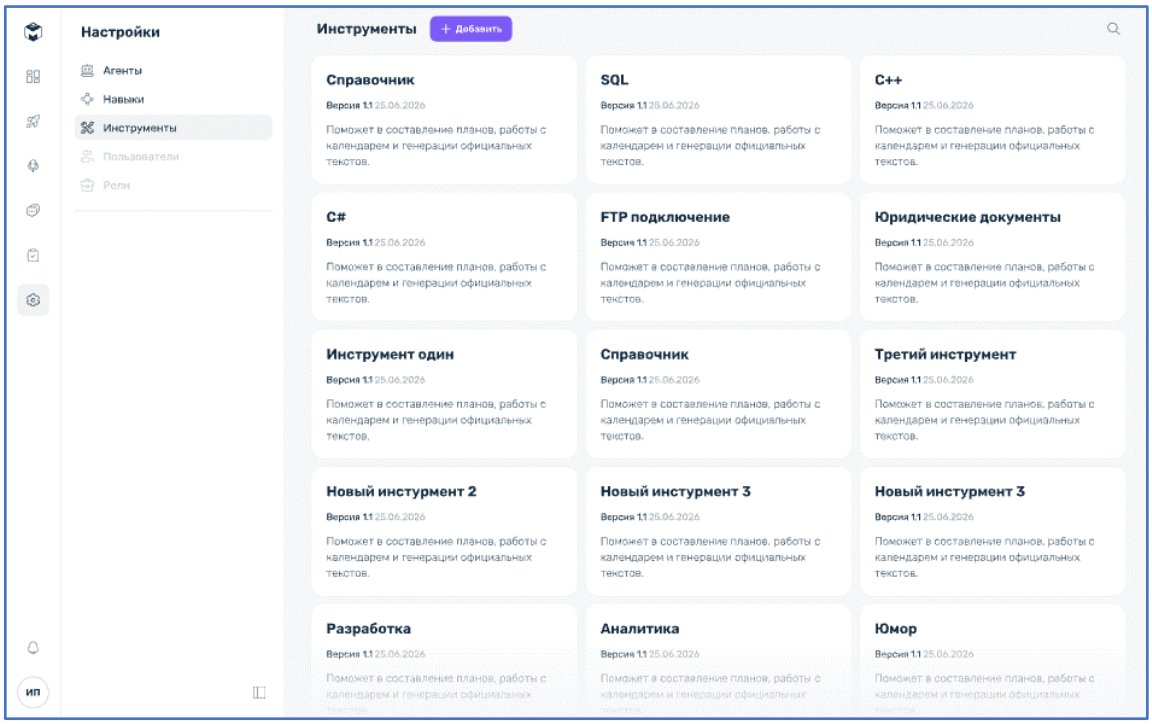


Рис. 18. Интерфейс раздела «Инструменты»

Управление инструментами включает:

- просмотр реестра инструментов;
- создание нового инструмента путем загрузки пакета формата zip;
- редактирование инструмента путем перезагрузки пакета;
- удаление инструмента;
- просмотр списка навыков, в которых используется инструмент.

7.4.1. Реестр инструментов

Страница реестра инструментов содержит:

- верхнее меню, которое содержит:
  - название раздела «Инструменты»;
  - кнопку «

» для загрузки пакета инструмента в формате zip;
  - кнопку «
», предназначенную для поиска инструментов.

- основную область, которая содержит реестр инструментов в виде мини-карточек, состоящих из:
  - названия инструмента;
  - версии инструмента с датой;
  - описания инструмента.

7.4.2. Карточка инструмента

При нажатии на мини-карточку открывается подробная карточка инструмента, содержащая поля:

- версия;
- дата;
- кнопка «Загрузить», предназначенную для перезагрузки пакета инструмента в формате zip;
- наименование;
- имя (техническое имя инструмента в системе);
- описание;
- список навыков, к которым подключен инструмент;
- кнопку «  », предназначенную для удаления инструмента;
- кнопку закрытия окна.

Интерфейс подробной карточки инструмента представлен на Рис. 19.

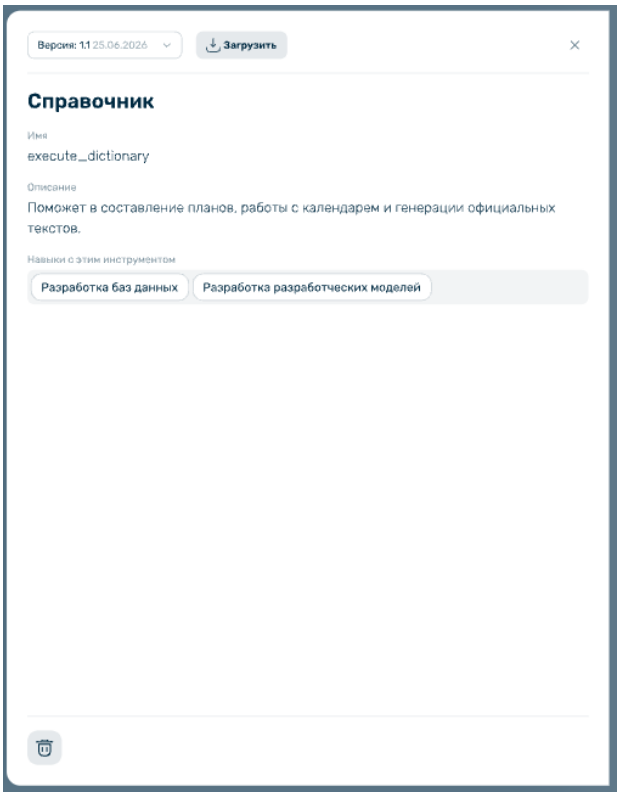


Рис. 19. Подробная карточка инструмента

Данные на форме доступны только для просмотра, все поля заполняются автоматически из загруженного пакета.

Для изменения данных предусмотрена загрузка нового инструмента из пакета по кнопке «Загрузить». В процессе обработки пакета отображается индикатор загрузки.

Предусмотрена возможность закрытия окна загрузки без сохранения внесенных изменений по кнопке « × ».

Если файл не был загружен, то отображается сообщение об ошибке с возможностью повторить загрузку путем нажатия на кнопку «Загрузить заново».

При успешной загрузке выдается сообщение и отображается кнопка «Перейти к инструменту», предназначенная для перехода в карточку инструмента.

8. УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ПОЛЬЗОВАТЕЛЕЙ

Разграничение прав пользователей защищает Модуль от несанкционированного использования путём введения ограничений на действия пользователей.

Каждый пользователь в системе оповещения имеет учётную запись (имя и пароль пользователя) одного из типов, перечисленных в подразделе 4.1.

В рамках управления учетными записями пользователей используются следующие роли:

- **Администратор платформы.** Выполняет техническое администрирование платформы: агенты, навыки (пакеты, версии), инструменты (в том числе БЗ), мониторинг и журналирование. А также функции Администратора организации и Сотрудника.
- **Администратор организации.** Администрирует пользователей своей организации, организационную структуру, роли и предоставление пользователям доступа к объектам
- **Сотрудник.** Использует разрешенные объекты: обращение к агентам, работа в доступных рабочих пространствах, участие в совместных каналах.

8.1. Раздел «Доступ и управление»

8.1.1. Подраздел «Пользователи»

Состоит из следующих блоков:

- Реестр пользователей
- Область фильтрации реестра пользователей
- Область быстрых действий с пользователем
- Переход к созданию пользователя
- Область пагинации или скрола
- Переход к блоку ведения ролей (опционально для удобства кроссперехода)

Поиск пользователей по:

- ФИО
- E-mail
- Отделу или департаменту

Фильтрация пользователей по параметрам:

- Статус
- Роль
- Организация
- Департамент
- Отдел

Создание пользователей возможно на основе следующих сценариев:

- автоматически при получении из внешнего источника (внешняя ИС, подсистема)
- вручную через Модуль ИИ-агентов Администратором платформы или организации.

При создании пользователя должен сохраняться атрибут с признаком «Источник учетной записи» (Модуль ИИ-агентов / Внешняя ИС).

8.1.2. Карточка пользователя

Состоит из следующих блоков:

- Основная информация
- Роли и доступы
- Сессии
- История изменений

8.1.2.1. Основная информация о пользователе

Блок включает следующие атрибуты:

- ФИО
- Email
- Организация
- Департамент
- Отдел
- Статус
- Источник учетной записи
- Дата создания
- Последний вход
- Редактировать (если создан вручную)
- Сменить пароль
- Заблокировать / Разблокировать

8.1.2.2. Роли и доступы

Блок включает следующие атрибуты:

- Роль пользователя определяет функциональные права пользователя в системе.
- Организационная принадлежность.
- Доступ к объектам системы.

8.1.2.3. Сессии

Блок включает следующие атрибуты:

- Устройство, с которого осуществлен доступ
- Начало сессии

- Последняя активность
- Статус сессии
- Завершить (для активной сессии)
- Завершить все активные сессии

9. ЭКСПЛУАТАЦИОННО-ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ МОДУЛЯ

Постоянная готовность к использованию Модуля достигается своевременным и качественным эксплуатационно-техническим обслуживанием.

Эксплуатационно-техническое обслуживание (ЭТО) включает в себя комплекс работ по поддержанию в исправном состоянии и восстановлению работоспособности модуля, выполняемых в период его использования по назначению.

К мероприятиям эксплуатационно-технического обслуживания относятся:

- техническое обслуживание;
- текущий ремонт;
- планирование и учёт эксплуатации и ремонта.

Эксплуатационно-техническое обслуживание осуществляется организацией, использующей модуль по назначению.

В работе Модуля возможны следующие аварийные ситуации:

- отказ модуля;
- сбой модуля.

Отказом модуля следует считать событие, состоящее в утрате работоспособности модуля и приводящее к невыполнению или неправильному выполнению контрольных примеров или функциональных задач.

Сбоем модуля следует считать событие, состоящее во временной утрате работоспособности модуля и характеризуемое возникновением ошибки при выполнении контрольных примеров или функциональных задач.

Своевременное и качественное выполнение работ по эксплуатационно-техническому обслуживанию модуля достигается:

- планированием эксплуатационно-технического обслуживания;
- систематическим контролем за техническим состоянием и готовностью модуля к использованию по предназначению;

- наличием резервных копий программного обеспечения и пользовательских данных;
- своевременным и полным обеспечением системы запасным имуществом;
- высокой профессиональной подготовкой специалистов по эксплуатационно-техническому обслуживанию и текущему ремонту;
- непрерывным и эффективным управлением силами и средствами эксплуатационно-технического обслуживания и текущего ремонта;
- изучением и обобщением опыта эксплуатационно-технического обслуживания, внедрением прогрессивных методов технического обслуживания и текущего ремонта.

Для восстановления модуля после отказа или сбоя, необходимо сначала устранить причину отказа/сбоя (заменить неисправное оборудование, устранить системные ошибки и др.), а затем предпринять следующие действия:

- установить операционную систему, а затем – соответствующий пакет обновления; проверить правильность работы домена;
- установить СУБД, а затем – соответствующий пакет обновления;
- восстановить базу данных и пользовательские данные из резервной копии; перезагрузить сервер после восстановления базы данных;
- проверить доступность модуля; чтобы убедиться в правильности работы, запустите сценарий проверки основных функций;
- активировать возможность работы пользователей в штатном режиме.

9.1. Проверка состояния компонентов Модуля

У каждого из сервисов есть эндпоинты:

~/api/health - отображает состояние сервиса, удалось ли зарегистрировать все зависимости, сбилдить контейнер,

~/api/health/ready - показывает готовность сервиса принимать запросы, пока здесь не будет статус Pass, ни один из запросов не пройдет сквозь специальный middleware компонент.

Указанные выше эндпоинты пингуются сервисом HealthStatusMonitor, который деплоится на порту 30071.

Эндпоинты HealthStatusMonitor:

/api/v1/health-monitor/services - возвращает JSON со статусами сервисов и ошибками с приоритетом сначала unhealthy, потом degraded, а после healthy.

/api/v1/health-monitor/services/html - возвращает HTML страницу со статусами сервисов и ошибками, обновляется сама в зависимости от значения PollInterval в конфигах. Удобнее открывать html страничку, там все отображается с разметкой.

Статусы:

- Pass - все ок (healthy)
- Warn - сервис работает, но есть какие-то нюансы (degraded)
- Fail - критическая ошибка (unhealthy)

9.2. Настройка диагностики

Все реализованные нами сервисы (DbService, AgentService и т.д.) подхватываются, если в DI зарегистрировали их RestPoint. Адреса RestPoint'ов подхватываются из общей конфигурации common.json (блок ServiceEndpoints), поэтому если добавили новый сервис, то нужно зарегистрировать его RestPoint и добавить в ServiceEndpoints адреса.

Реализованные не нами сервисы (SeaweedFS, OpenSearch и т.д.) нужно добавлять в конфиг самого сервиса в блок «ExternalServices». Там хранится словарь, где ключ будет служить названием сервиса при отображении, а значением будут сами настройки. Настройки следующие:

Address - адрес соответственно.

HealthPath - путь для запроса health-статуса.

ReadyPath - путь для запроса ready-статуса, если не заполнен, то ready-опрос не происходит и туда просто подставляется значение из health-опроса.

Interpreter - название интерпретатора результата, если не указывается, то по дефолту используется "status-code". Можно указать кастомный интерпретатор, поскольку контракты отличаются, их названия лежат в InterpreterName самого класса интепретатора (лежат в папке Interpreters сервиса).

OkStatusCodes - список статус кодов, считающихся успешными, используется с "status-code" интерпретаторе. Если не заполнены, то все между 200 и 299 включительно считается успехом.

Headers - словарь string-string с кастомными заголовками, используется при опросе ExternalServices, значения в этом конфиге кладутся прямо в том виде, в котором они лежат.

Пример конфигурации:

```
{
  "PollInterval": "00:00:15", // период между опросами каждого сервиса
  "ProbeTimeout": "00:00:05", // таймаут одного опроса
  "MaxParallelProbes": 8, // максимальное параллельно количество опросов
  "ExcludeRestPoints": [], // RestPoint'ы, которые надо исключить из опроса, пишется строкой (например, DbServiceRestPoint)
  "ExternalServices": {
    "OpenSearch": {
      "Address": "http://opensearch:2222",
      "HealthPath": "_cluster/health",
      "Interpreter": "opensearch",
      "Headers": {
        "SomeHeader": "SomeValue"
      }
    },
    "Docling": {
      "Address": "http://docling:3333",
      "HealthPath": "health",
      "ReadyPath": "ready",
      "Interpreter": "docling",
      "OkStatusCodes": [200]
    }
  },
  "RestServerSettings": {
    "Host": "https://0.0.0.0",
    "Port": 30071,
    "SecurityProtocol": "Tls12",
    "EnableSwagger": true
  }
}
```

9.3. Запуск диагностического модуля

На старте приложения происходит опрос адресов, указанных в ServiceEndpoints, по аналогии с другими сервисами. Пока сервисы не опрошены для резолва адреса, будет выдано предупреждение "Address is not resolved yet". Если предупреждение висит долго, значит не удается зарезолвить, возможно что-то не так в конфигах, не указан адрес или сервисы недоступны вовсе ни по одному из указанных адресов.

Пока health-опрос не удался, то ready-опрос тоже не будет производиться.

ПРИЛОЖЕНИЕ 1. ПЕРЕЧЕНЬ СТОРОННИХ КОМПОНЕНТОВ МОДУЛЯ

Основной проект C# библиотеки

Перечень сторонних компонентов основного проекта C# библиотеки, используемых Модулем, приведен в Табл. 42.

Табл. 42. Перечень сторонних компонентов основного проекта C# библиотеки

Название	Версия	Лицензия	Описание	Использо-вание
.NET	10.0	https://github.com/dotnet/dotnet/releases/tag/v10.0.0	Среда выполнения микросервисов ПО (runtime)	Используется
Autofac	9.1.0	https://github.com/autofac/Autofac/releases/tag/v9.1.0	DI-контейнер (внедрение зависимостей)	Используется
Entity Framework Core	10.0.10	https://www.nuget.org/packages/Microsoft.EntityFrameworkCore/10.0.10	ORM для работы с базами данных	Используется
Npgsql	10.0.1	PostgreSQL License https://www.nuget.org/packages/Npgsql/10.0.1	Провайдер PostgreSQL для .NET	Используется
OpenSearch (клиент)	1.8.0	https://www.nuget.org/packages/OpenSearch.Client/1.8.0	Адаптер для работы с OpenSearch (хранение контекста сессий)	Используется

Название	Версия	Лицензия	Описание	Использо-вание
Confluent Kafka	2.12.0	https://github.com/confluentinc/confluent-kafka-dotnet/releases/tag/v2.12.0	Библиотека взаимодействия с Kafka	Используется
Consul	1.8.0	MPL-2.0 https://github.com/hashicorp/consul/tree/v1.8.0	Service Discovery и централизованное хранение конфигурации	Используется
NLog	6.1.2	BSD-3-Clause https://www.nuget.org/packages/NLog/6.1.2	Логирование работы ПО	Используется
Mapster	10.0.11	https://www.nuget.org/packages/Mapster/10.0.11	Преобразование DTO и объектов (маппинг)	Используется
Swagger (Swashbuckle)	10.2.3	https://www.nuget.org/packages/Swashbuckle.AspNetCore/10.2.3	Документирование REST API	Используется
DoclingSharp	8.17.9	https://www.nuget.org/packages/DoclingSharp/8.17.9	Конвертация документов	Используется
AWS SDK S3	4.0.21.2	https://www.nuget.org/packages/AWSSDK.S3/4.0.21.2	Взаимодействие с S3-совместимым объектным хранилищем	Используется
Docker.DotNet.Enhanced	4.3.3	https://www.nuget.org/packages/Docker.DotNet.Enhanced/4.3.3	Взаимодействие с Docker из .NET-приложения	Используется
OllamaSharp	5.4.25	https://www.nuget.org/packages/OllamaSharp/5.4.25	Взаимодействие с Ollama и языковыми моделями	Используется

Название	Версия	Лицензия	Описание	Использо-вание
Kerberos.NET	4.6.146	https://www.nuget.org/packages/Kerberos.NET/4.6.146	Аутентификация пользователей	Используется
NAudio	2.2.1	https://www.nuget.org/packages/NAudio/2.2.1	Работа с аудиоданными	Используется
xUnit	2.9.2	https://www.nuget.org/packages/xunit/2.9.2	Модульное тестирование	Используется
Autofac.DependencyInjection	11.0.0	https://www.nuget.org/packages/Autofac.Extensions.DependencyInjection/11.0.0	Интеграция контейнера Autofac с механизмом внедрения зависимостей .NET	Используется
Autofac.Pooling	1.0.1	https://www.nuget.org/packages/Autofac.Pooling/1.0.1	Библиотека управления пулом объектов для Autofac	Используется
Markdig	1.1.3	BSD-2 https://www.nuget.org/packages/Markdig/1.1.3	Парсинг Markdown	Используется
Microsoft.Agents.AI	1.7.0	https://www.nuget.org/packages/Microsoft.Agents.AI/1.7.0	SDK для разработки AI-агентов	Используется
Microsoft.Agents.AI.Workflows	1.7.0	https://www.nuget.org/packages/Microsoft.Agents.AI.Workflows/1.7.0	Средства построения рабочих процессов AI-агентов	Используется
Microsoft.Agents.AI.Abstractions	1.7.0	https://www.nuget.org/packages/Microsoft.Agents.AI.Abstractions/1.7.0	Базовые абстракции и интерфейсы для AI-агентов	Используется
Microsoft.AspNetCore.Http.Abstractions	2.3.11	https://www.nuget.org/packages/Microsoft.AspNetCore.Http.Abstractions/2.3.11	Абстракции HTTP ASP.NET Core	Используется

Название	Версия	Лицензия	Описание	Использо-вание
Microsoft.Extensions.DependencyInjection	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.DependencyInjection/10.0.10	Контейнер и средства внедрения зависимостей для .NET	Используется
Microsoft.Diagnostics.HealthChecks	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.Diagnostics.HealthChecks/10.0.10	Проверки работоспособности	Используется
Microsoft.Extensions.AI.Abstractions	10.8.3	https://www.nuget.org/packages/Microsoft.Extensions.AI.Abstractions/10.8.3	Абстракции для работы с компонентами генеративного ИИ	Используется
Microsoft.Extensions.Logging	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.Logging/10.0.10	Средства журналирования для .NET	Используется
Microsoft.Extensions.Hosting	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.Hosting/10.0.10	Инфраструктура хостинга и жизненного цикла .NET-приложений	Используется
Microsoft.Extensions.Configuration	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.Configuration/10.0.10	Система конфигурации .NET-приложений	Используется
Microsoft.Configuration.Json	10.0.10	https://www.nuget.org/packages/Microsoft.Extensions.Configuration.Json/10.0.10	Провайдер конфигурации из JSON-файлов	Используется
Microsoft.Options.ConfigurationExtensions	9.0.8	https://www.nuget.org/packages/Microsoft.Extensions.Options.ConfigurationExtensions/9.0.8	Расширения для настройки параметров приложения через конфигурацию	Используется

Название	Версия	Лицензия	Описание	Использо-вание
Microsoft.Windows.Compatibility	10.0.10	https://www.nuget.org/packages/Microsoft.Windows.Compatibility/10.0.10	API совместимости Windows для .NET	Используется
Microsoft.AspNetCore.Authentication.JwtBearer	10.0.10	https://www.nuget.org/packages/Microsoft.AspNetCore.Authentication.JwtBearer/10.0.10	Аутентификация ASP.NET Core с использованием JWT Bearer	Используется
NLog.Extensions.Logging	6.1.2	BSD-2 https://www.nuget.org/packages/NLog.Extensions.Logging/6.1.2	Расширения NLog	Используется
Oracle.ManagedDataAccess.Core	2.19.91	OTN License https://www.nuget.org/packages/Oracle.ManagedDataAccess.Core/2.19.91	Провайдер данных ADO.NET для доступа .NET-приложений к базам данных Oracle	Используется
Microsoft.Data.Sqlite	10.0.10	https://www.nuget.org/packages/Microsoft.Data.Sqlite/10.0.10	Провайдер SQLite для .NET	Используется
Tmds.DBus.Protocol	0.92.0	https://www.nuget.org/packages/Tmds.DBus.Protocol/0.92.0	Протокол D-Bus для Linux	Используется
AutoFixture	4.18.1	https://www.nuget.org/packages/AutoFixture/4.18.1	Генерация тестовых данных	Используется
.NET Test SDK	18.0.1	https://www.nuget.org/packages/Microsoft.NET.Test.Sdk/18.0.1	SDK для тестирования .NET	Используется
Coverlet.Collector	6.0.4	https://www.nuget.org/packages/coverlet.collector/6.0.4	Сборщик покрытия кода	Используется
xUnit.Runner	2.9.2	https://www.nuget.org/packages/xunit.runner.visualstudio/2.9.2	Средство запуска тестов xUnit	Используется

Название	Версия	Лицензия	Описание	Использо-вание
NSubstitute	5.3.0	BSD-3 https://www.nuget.org/packages/NSubstitute/5.3.0	Библиотека моков для тестирования	Используется

Используемые в архитектуре сторонние компоненты

Перечень используемых в архитектуре Модуля сторонних компонентов приведен в Табл. 43.

Табл. 43. Используемые в архитектуре Модуля сторонние компоненты

Название	Версия	Лицензия	Описание	Использо-вание
PostgreSQL	18.3	PostgreSQL License https://www.postgresql.org/	Реляционная база данных для DbService	Используется
Microsoft.Data.Sqlite 10.0.10	10.0.10	https://www.nuget.org/packages/Microsoft.Data.Sqlite/10.0.10	ADO.NET-провайдер для работы с базой данных SQLite в .NET	Используется
OpenSearch (сервер)	3.6.0	https://github.com/opensearch-project/OpenSearch/blob/3.6.0/LICENSE.txt	Поисковый движок для ContextStorage	Используется
Consul (сервер)	1.22	BUSL-1.1 https://github.com/hashicorp/consul	Service Discovery и конфигурация	Используется
Kafka (сервер)	4.1.0	https://github.com/apache/kafka/blob/3.0.0/LICENSE	Брокер сообщений	Используется
SeaweedFS	4.40	https://github.com/seaweedfs/seaweedfs/blob/4.40/LICENSE	S3-совместимое объектное хранилище для архивов агентов, навыков и файлов	Используется

Название	Версия	Лицензия	Описание	Использо-вание
Docker	24.0.0	https://github.com/moby/moby/blob/v24.0.0/LICENSE	Контейнеризация и запуск песочниц агентов	Используется

Сторонние Frontend компоненты

Перечень сторонних Frontend компонентов, используемых Модулем, приведен в Табл. 44.

Табл. 44. Сторонние Frontend компоненты

Название	Версия	Лицензия	Описание	Использо-вание
@fontsource/source-sans-pro	5.2.5	OFL-1.1 (SIL Open Font License)	Шрифт Source Sans Pro (Frontend)	Используется
@speechpro/sx_cn-ui_kit	0.1.73-master	внутренний/private npm-пакет SpeechPro	UI Kit SpeechPro (Frontend)	Используется
React	19.2.0	https://www.npmjs.com/package/react/v/19.2.0	Библиотека UI (Frontend)	Используется
React DOM	19.2.0	https://www.npmjs.com/package/react-dom/v/19.2.0	Рендеринг React (Frontend)	Используется
React JSS	10.10.0	https://www.npmjs.com/package/react-jss/v/10.10.0	Стилизация React через JSS (Frontend)	Используется
React Router DOM	7.16.0	https://www.npmjs.com/package/react-router-dom/v/7.16.0	Маршрутизация React (Frontend)	Используется

Название	Версия	Лицензия	Описание	Использо-вание
uuid	14.0.0	https://www.npmjs.com/package/uuid/v/14.0.0	Генерация UUID (Frontend)	Используется
@lottiefiles/dotlottie-react	0.19.4	https://www.npmjs.com/package/@lottiefiles/dotlottie-react/v/0.19.4	Анимации Lottie для React (Frontend)	Используется
@types/lodash	4.17.24	https://www.npmjs.com/package/@types/lodash/v/4.17.24	Типы TypeScript для Lodash (Frontend)	Используется
@types/node	24.12.0	https://www.npmjs.com/package/@types/node/v/24.12.0	Типы TypeScript для Node.js (Frontend)	Используется
@types/react	19.2.14	https://www.npmjs.com/package/@types/react/v/19.2.14	Типы TypeScript для React (Frontend)	Используется
@types/react-dom	19.2.3	https://www.npmjs.com/package/@types/react-dom/v/19.2.3	Типы TypeScript для React DOM (Frontend)	Используется
@vitejs/plugin-react	5.2.0	https://www.npmjs.com/package/@vitejs/plugin-react/v/5.2.0	Плагин Vite для React (Frontend)	Используется
classnames	2.5.1	https://www.npmjs.com/package/classnames/v/2.5.1	Утилиты для CSS-классов (Frontend)	Используется
date-fns	4.4.0	https://www.npmjs.com/package/date-fns/v/4.4.0	Работа с датами (Frontend)	Используется
highlight.js	11.11.1	BSD-3 https://www.npmjs.com/package/highlight.js/v/11.11.1?utm_source=chatgpt.com	Подсветка синтаксиса кода (Frontend)	Используется
i18next	26.0.9	https://www.npmjs.com/package/i18next/v/26.0.9	Интернационализация (Frontend)	Используется

Название	Версия	Лицензия	Описание	Использо-вание
javascript-obfuscator	5.3.0	https://www.npmjs.com/package/javascript-obfuscator/v/5.3.0	Обфускация JavaScript (Frontend)	Используется
katex	0.16.22	https://www.npmjs.com/package/katex/v/0.16.22	Рендеринг математических формул (Frontend)	Используется
lint-staged	17.3.0	https://www.npmjs.com/package/lint-staged/v/17.3.0	Запуск линтеров на закоммиченных файлах (Frontend)	Используется
lodash	4.18.1	https://www.npmjs.com/package/lodash/v/4.18.1	Утилиты JavaScript (Frontend)	Используется
MobX	6.15.0	https://www.npmjs.com/package/mobx/v/6.15.0	Управление состоянием (Frontend)	Используется
mobx-react-lite	4.1.1	https://www.npmjs.com/package/mobx-react-lite/v/4.1.1	Интеграция MobX с React (Frontend)	Используется
Prettier	3.9.6	https://www.npmjs.com/package/prettier/v/3.9.6	Форматирование кода (Frontend)	Используется
react-i18next	17.0.6	https://www.npmjs.com/package/react-i18next/v/17.0.6	Интернационализация для React (Frontend)	Используется
react-markdown	10.1.0	https://www.npmjs.com/package/react-markdown/v/10.1.0	Рендеринг Markdown в React (Frontend)	Используется
rehype-highlight	7.0.2	https://www.npmjs.com/package/rehype-highlight/v/7.0.2	Подсветка синтаксиса в rehype (Frontend)	Используется
rehype-katex	7.0.1	https://www.npmjs.com/package/rehype-katex/v/7.0.1	Рендеринг формул KaTeX в rehype (Frontend)	Используется

Название	Версия	Лицензия	Описание	Использо-вание
remark-gfm	4.0.1	https://www.npmjs.com/package/remark-gfm/v/4.0.1	Поддержка GitHub Flavored Markdown (Frontend)	Используется
remark-math	6.0.0	https://www.npmjs.com/package/remark-math/v/6.0.0	Поддержка математики в remark (Frontend)	Используется
Sass	1.99.0	https://www.npmjs.com/package/sass/v/1.99.0	CSS-препроцессор (Frontend)	Используется
TypeScript	5.9.3	https://www.npmjs.com/package/typescript/v/5.9.3	Язык программирования (Frontend)	Используется
typescript-eslint	8.57.0	https://www.npmjs.com/package/typescript-eslint/v/8.57.0	ESLint для TypeScript (Frontend)	Используется
Vite	8.0.5	https://www.npmjs.com/package/vite/v/8.0.5	Сборщик фронтенда (Frontend)	Используется
vite-plugin-javascript-obfuscator	3.1.0	https://www.npmjs.com/package/vite-plugin-javascript-obfuscator/v/3.1.0	Плагин обфускации для Vite (Frontend)	Используется
Husky	9.1.7	https://www.npmjs.com/package/husky/v/9.1.7	Git hooks (Frontend)	Используется
ESLint	9.39.4	https://www.npmjs.com/package/eslint/v/9.39.4	Линтер JavaScript/TypeScript (Frontend)	Используется
eslint-config-prettier	10.1.8	https://www.npmjs.com/package/eslint-config-prettier/v/10.1.8	Конфигурация ESLint для Prettier (Frontend)	Используется
eslint-plugin-react-hooks	7.0.1	https://www.npmjs.com/package/eslint-plugin-react-hooks/v/7.0.1	Правила ESLint для React Hooks (Frontend)	Используется

Название	Версия	Лицензия	Описание	Использо-вание
eslint-plugin-react-refresh	0.5.2	https://www.npmjs.com/package/eslint-plugin-react-refresh/v/0.5.2	Плагин ESLint для React Refresh (Frontend)	Используется
@trivago/prettier-plugin-sort-imports	6.0.2	https://www.npmjs.com/package/@trivago/prettier-plugin-sort-imports/v/6.0.2	Плагин Prettier для сортировки импортов (Frontend)	Используется
@eslint/js	9.39.4	https://www.npmjs.com/package/@eslint/js/v/9.39.4	ESLint JS (Frontend)	Используется
globals	17.4.0	https://www.npmjs.com/package/globals/v/17.4.0	Глобальные переменные (Frontend)	Используется