

**КОРПОРАТИВНЫЙ СЕРВЕР 2024**

**ОПИСАНИЕ ПРОГРАММЫ**

RU.48324255.KC2024-ОП-В1.0

Листов 41

Инв. №	Подпись и дата	Взам. инв. №	Инв. № оубл.	Подп. и дата

## АННОТАЦИЯ

В данном программном документе приведено описание Программы для ЭВМ «Корпоративный сервер 2024».

Программа предназначена для создания единого цифрового пространства, где пользователи могут совместно работать с документами, планировать задачи, управлять проектами и взаимодействовать друг с другом. **Корпоративный сервер 2024** обеспечивает централизованное хранение данных, а также предоставляет инструменты для организации расписания, ведения электронной переписки, управления контактами и создания страниц, позволяет работать с графикой, создавать интерактивные формы для сбора информации и управлять версиями файлов.

В документе приведены:

- Общие сведения о программе;
- Функциональное назначение программы;
- Описание логической структуры;
- Используемые технические средства;
- Вызов и загрузка;
- Входные данные;
- Выходные данные.

Документ разработан в соответствии с требованиями:

- ГОСТ 19.402–79 ЕСПД. Описание программы;
- ГОСТ 19.105–78 ЕСПД. Общие требования к программным документам;
- ГОСТ 19.106–78 ЕСПД. Общие требования к программным документам, выполненным печатным способом.

## СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ТЕРМИНОВ .....	5
1. ОБЩИЕ СВЕДЕНИЯ.....	6
1.1 Обозначение и наименование программы.....	6
1.2 Программное обеспечение, необходимое для функционирования системы...6	
1.3 Языки программирования, на которых написана программа.....	8
1.3.1 Язык программирования TypeScript.....	8
1.3.2 Язык программирования C#.....	8
1.3.3 Язык программирования SQL.....	9
1.3.4 Язык программирования JavaScript.....	9
1.3.5 Bash/Shell/Powershell.....	9
2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	10
2.1 Назначение программы.....	10
2.2 Описание функционирования.....	10
2.3 Ограничения на применение.....	11
3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ.....	14
3.1 Алгоритм программы.....	14
3.2 Используемые методы.....	15
3.3 Структура программы с описанием функций составных частей и связи между ними.....	18
3.3.1 Общая архитектура КС 2024.....	18
3.3.2 Подсистема «Календарь».....	19
3.3.3 Подсистема «Контакты».....	21
3.3.4 Подсистема «Почта».....	22
3.3.5 Подсистема «Проекты».....	24
3.3.6 Административный модуль.....	25

3.3.7	Общий web-контур (Frontend SPA).....	27
3.4	Связи программы с другими системами (программами).....	29
4.	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	30
5.	ВЫЗОВ И ЗАГРУЗКА.....	32
6.	ВХОДНЫЕ ДАННЫЕ.....	33
6.1	Входные данные от пользователей.....	33
6.2	Входные данные от других систем через API и интеграционные каналы.....	35
6.3	Входные данные от серверной инфраструктуры.....	36
7.	ВЫХОДНЫЕ ДАННЫЕ.....	38
7.1	Модуль «Календарь».....	38
7.2	Модуль «Контакты».....	38
7.3	Модуль «Почта».....	38
7.4	Модуль «Проекты».....	39
7.5	Административный модуль.....	39
7.6	Уведомления и выходные сущности.....	39
7.7	Диагностические и служебные выходные данные.....	40
7.8	Примеры выходных данных.....	40
	Лист регистрации изменений.....	41

## ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ТЕРМИНОВ

Обозначение	Определение
Гб	Гигабайт, единица измерения количества информации
ГГц	Гигагерц, единица измерения частоты периодических процессов в Международной системе единиц (СИ)
LDAP	Протокол доступа к службам каталогов
MQ	Система управления очередями сообщений (Message Queue)
КС 2024	Корпоративный сервер 2024
ОС	Операционная система
ПК	Персональный компьютер
Redis	Система временного кеша и хранения сессионных данных
IP-фильтрация	Ограничение доступа к системе по IP-адресам

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1 Обозначение и наименование программы

Полное название: Корпоративный сервер 2024.

Сокращенное наименование: КС 2024.

Номер документа: RU.48324255.КС2024-ОП-В1.0.

## 1.2 Программное обеспечение, необходимое для функционирования системы

Для корректной работы **КС 2024** необходимо соблюдение следующих требований к программному и аппаратному обеспечению:

### Требования к клиентской части (веб-браузеры)

Система представлена в виде веб-приложения, совместимого с современными актуальными версиями большинства браузеров:

- Google Chrome;
- Яндекс Браузер;
- Mozilla Firefox;
- Safari.


### Минимальные требования к серверу:

- Поддерживается только 64-битная архитектура;
- Процессор от 2 ядер;
- Оперативная память от 8 Гб;
- Свободное место на жестком диске от 50 Гб.

### Поддерживаемые операционные системы:

- Astra Linux Special Edition:
  - Astra Special Edition 1.8.1 «Максимальная защищенность Смоленск» (только онлайн-установка);
  - Astra Special Edition 1.7.6 «Максимальная защищенность Смоленск» (только онлайн-установка);

- Astra Special Edition 1.7.4 «Базовая защищенность Орел» (только офлайн-установка).
- РЕД ОС:
  - Версия 8.0;
  - Версия 7.3.4.
- Альт Линукс:
  - Дистрибутив «Альт Сервер» версии 10.2.
- Дебиан:
  - Дебиан 12.


 **Примечание**

Онлайн-установка для версии 2025.4.6.16449  
Офлайн-установка только для версии 2025.3.18.15967

### **Установка с помощью инсталлятора в Docker-compose**

Установка с помощью Docker возможна на дистрибутивы:

- ALT Linux версии Альт сервер 10.1;
- Astra Special Edition 1.8.1 Редакция Максимальная защищенность Смоленск;
- РЕД ОС 8.0 стандартная редакция;
- Debian 12.

 **Примечание**

ОС должна поддерживать Python версии 3.9 или выше.

### **Требования к клиентским рабочим станциям:**

Клиентские рабочие станции должны соответствовать следующим минимальным техническим характеристикам:

- Двухъядерный процессор, с тактовой частотой от 2 ГГц и выше;
- Не менее 2 ГБ оперативной памяти;
- Не менее 2 ГБ свободной памяти на жёстком диске.

### 1.3 Языки программирования, на которых написана программа

**Корпоративный сервер 2024** написан на основе следующих языков программирования:

- bash/sh;
- C#;
- JavaScript;
- Bash;
- Java;
- SQL;
- TypeScript.

#### 1.3.1 Язык программирования TypeScript

**TypeScript** — объектно-ориентированный язык программирования, разработанный компанией как надстройка над JavaScript. Поддерживает статическую типизацию, классы, интерфейсы и другие продвинутое конструкции, повышающие надежность и читаемость кода.

**TypeScript** используется для реализации клиентской части приложения. На нём написаны ключевые компоненты пользовательского интерфейса, логика состояний, компоненты форм, маршруты навигации и сервисы интеграции с сервером.

#### 1.3.2 Язык программирования C#

**C#** — объектно-ориентированный язык программирования, созданный специально для платформы **.NET**. Характеризуется высокой производительностью, богатым набором встроенных библиотек и развитой системой типов.

Серверная часть приложения реализована преимущественно на языке **C#** с применением платформы **.NET 6**. Этот язык выбран для реализации бизнес-логики, контроллеров **REST API**, фоновых задач и интеграции с внешними сервисами. Использование **C#** позволило добиться высокой производительности и надежности серверной составляющей.

### 1.3.3 Язык программирования SQL

**SQL** (Structured Query Language) — специализированный декларативный язык программирования, предназначенный для работы с реляционными базами данных. Используется для создания, чтения, обновления и удаления данных, а также для управления схемой базы данных.

**SQL** применяется для работы с базой данных. Скрипты, содержащиеся в проекте, предназначены для создания и модификации структуры таблиц, индексов и представлений в СУБД **PostgreSQL**. Эти скрипты определяют структуру хранилищ данных, используемых приложением, и обеспечивают целостность и согласованность хранимой информации.

### 1.3.4 Язык программирования JavaScript

**JavaScript** — универсальный динамический язык программирования, широко применяемый для веб-разработки и автоматизации задач. Является основой популярных сред исполнения вроде Node.js и браузеров.

**JavaScript** используется в инфраструктурных целях: для настройки сборочного процесса, автоматизации CI/CD, компиляции и упаковки приложения. Также JS применяется для написания служебных скриптов и конфигурационных файлов, поддерживающих разработку и эксплуатацию приложения.

### 1.3.5 Bash/Shell/Powershell

**Bash/Shell/Powershell** — языки сценариев командной оболочки операционной системы, предназначенные для автоматизации рутинных задач и управления процессами. Применяются для выполнения последовательностей команд, настройки окружения и управления службами. Bash/Shell/Powershell применяются для автоматизации процедур установки, обновления и развёртывания приложения. Они используются в установочных и конфигурационных скриптах, предназначенных для подготовки рабочей среды и запуска служб приложения.

## 2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

### 2.1 Назначение программы

**КС 2024** предназначен для организации единой цифровой среды предприятия, позволяющей сотрудникам хранить, обрабатывать и совместно работать с информацией, документами и задачами. Программа объединяет функции коллективной работы, управления проектами, календарного планирования, почтовой службы и централизованной системы хранения данных.

### 2.2 Описание функционирования

**КС 2024** представляет собой платформу, объединяющую клиентские и серверные компоненты для обеспечения взаимодействия пользователей организации.

Основные функциональные возможности включают:

- Система совместного хранения и работы с документами:
  - Централизованный облачный диск для хранения файлов любых форматов;
  - Управление версиями файлов, восстановление предыдущих версий, создание копий;
  - Возможность предоставления доступа к документам внешним пользователям с настраиваемыми ограничениями (копирование, скачивание, печать).
- Электронная почта и календарь:
  - Интегрированная почтовая служба с поддержкой стандартных протоколов IMAP/SMTP;
  - Планировщик мероприятий с возможностью отправки приглашений, отслеживания статусов участия и синхронизации с внешними календарями CalDAV/iCalendar.
- Управление задачами и проектами:
  - Инструменты постановки и распределения задач среди сотрудников;

- Контроль сроков выполнения, статуса задач и назначений ответственным лицам;
- Гибкая система фильтров и отчетов по проектам и задачам.
- Централизованная система контактов:
  - Общий реестр контактов сотрудников и партнеров компании;
  - Синхронизация контактов с почтой, календарем и рабочими задачами.
- Модуль «Графика»:
  - Предназначен для работы с векторной и растровой графикой, аналог программ типа Microsoft Visio или CorelDRAW, интегрированных непосредственно в офисный пакет;
  - Позволяет создавать, редактировать и форматировать графические объекты прямо в документах либо как самостоятельные файлы.
- Модуль «Формы»:
  - Представляет собой инструмент для создания, настройки и распространения интерактивных форм сбора данных;
  - Цель модуля — предоставить пользователям удобный способ создания анкет, опросов и других типов форм с возможностью гибкой настройки и анализа полученных ответов.
- Административные функции:
  - Мониторинг активности пользователей, аудит действий и контроль доступа;
  - Настройки безопасности и резервного копирования данных.

### 2.3 Ограничения на применение

Ограничения на применение **Корпоративный сервер 2024** можно условно разделить на три группы:

- Технические и системные;
- Функциональные;
- Лицензионные и административные.

#### 1. Технические и системные ограничения:

- Требования к оборудованию: минимальные рекомендуемые системные требования составляют 2 ядра CPU и 8 ГБ оперативной памяти, что подходит для одновременной работы до 300 пользователей. Для увеличения числа пользователей потребуется соответствующее увеличение вычислительных ресурсов;
- Операционная система: поддерживается исключительно 64-битная архитектура;
- Ограничения на размеры файлов: по умолчанию действует ограничение на загрузку и копирование файлов через протокол WebDAV размером более 100 МБ. Данные лимиты можно изменить вручную через конфигурационные файлы сервера.

## 2. Функциональные ограничения:

- Цифровая подпись (ЭП): в настоящий момент цифровая подпись документов доступна только в десктопных приложениях для локально открытых файлов. Подписывать документы, хранящиеся на сервере, через браузерную версию невозможно;
- Модули расширения: базовая поставка «Корпоративного сервера 2024» не включает модуль «Р7-Графика», который необходим для работы с векторной и растровой графикой. Этот модуль доступен отдельно и требует отдельной установки;
- Макросы и скрипты: макросы и скрипты автоматизации работают только в пределах отдельного документа. Невозможно создать единый глобальный макрос, работающий одновременно со всеми документами на сервере.

## 3. Лицензионные и административные ограничения:

- Квоты и ограничения дискового пространства: администратор системы вправе ограничить максимальный объем дискового пространства индивидуально каждому пользователю, группе пользователей или целым общим папкам («комнаты»);

- Права администраторов: существует разделение уровней администраторов. Обычные администраторы обладают ограниченными полномочиями и не могут изменять или удалять учетную запись суперадминистратора системы.

### 3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

#### 3.1 Алгоритм программы

**Корпоративный сервер 2024** построен на микросервисной архитектуре и реализует следующий набор алгоритмов:

- ASP.NET Core — основа серверной части, обеспечивающая высокопроизводительную реализацию REST API и обработку HTTP-запросов;
- RabbitMQ — используется для асинхронной обработки задач, постановки заданий в очереди и межсервисного взаимодействия (pub/sub);
- JWT (JSON Web Token) — технология аутентификации и авторизации пользователей, передача прав доступа и идентификационной информации между клиентами и сервером;
- LDAP/Kerberos/OAuth2/SAML2 — внешние механизмы интеграции с системами аутентификации и авторизации;
- DLP/ICAP — встроенная защита данных, мониторинг и контроль потоков информации;
- Syslog/SIEM — механизмы интеграции с центрами мониторинга и безопасности для регистрации событий и аудита;
- Processing Host — фоновый сервис для выполнения длительных операций (миграция данных, синхронизация, резервное копирование, экспорт данных).
- Централизованное управление правами доступа — проверка прав пользователей на основе ролей и иерархии административной модели.

Основные этапы обработки запросов и выполнения операций:

1. Входящие HTTP-запросы поступают на балансировщик нагрузки (nginx) и направляются на соответствующие микросервисы.
2. Микросервисы выполняют проверку подлинности пользователя и его прав доступа.

3. Затем производится обработка бизнес-логики (создание, изменение, удаление объектов, синхронизация данных, выполнение фоновых задач).
4. Ответ возвращается клиенту в формате JSON или направляется в очередь сообщений для последующей асинхронной обработки.
5. Фоновые задания выполняются отдельными сервисами (Processing Host), результаты возвращаются пользователю или фиксируются в журнале.

### **3.2 Используемые методы**

В КС 2024 используются следующие методы:

#### **Методы клиент-серверного взаимодействия:**

- REST API по HTTP/HTTPS — основное средство взаимодействия клиентской и серверной частей, использующее стандартные методы (GET, POST, PUT, DELETE) и передачу данных в формате JSON;
- Reverse Proxy (nginx) — распределение входящих запросов между микросервисами, обеспечение безопасности и отказоустойчивости;
- Cookie-Based Authentication — аутентификация и авторизация пользователей через cookies, передаваемые в заголовках HTTP-запросов;
- SSO/OAuth2/SAML2/Ldap/Kerberos — интеграция с внешними системами аутентификации и авторизации;
- Центр обработки ошибок — централизованная обработка HTTP-статусов (401, 403, 404, 500 и др.) и информирование пользователя.

#### **Методы централизованного хранения данных:**

- Централизованная реляционная база данных (PostgreSQL) — хранение данных пользователей, ролей, подразделений, настроек, квот, аудиторных записей и других метаданных;
- Redis — временное кеширование сессионных данных, блокировки, списки присутствия пользователей;

- **Файловое хранилище** — локальное или распределённое (S3-совместимое) хранилище для файлов, прикрепляемых к объектам (документы, сертификаты, логотипы и прочее);
- **Воскрешение** — регулярное резервное копирование базы данных и файловой инфраструктуры (осуществляется вне клиентской части, средствами DevOps-инфраструктуры).

#### **Методы разграничения прав доступа:**

- **Ролевая модель (RBAC)** — разграничение прав доступа на основе ролей пользователей (Администратор, Суперадминистратор, Менеджер, Пользователь и др.);
- **Иерархическая модель доступа** — наследование прав доступа от родительских объектов (организация → подразделение → группа → пользователь);
- **IP-фильтрация** — белые и черные списки IP-адресов для ограничения доступа к критичным действиям и данным;
- **Разграничение прав на уровне объектов** — индивидуальные разрешения на доступ к конкретным элементам (документ, задача, проект, контакт и т.д.);
- **Аудит и журнал действий** — фиксация всех значимых операций пользователей для последующего анализа и расследования.

#### **Методы интеграции приложений:**

- **Внешние интеграции** — подключение внешних систем через REST API, LDAP, OAuth2, SAML2, ICAP, SMTP, IMAP, CalDAV, iMIP;
- **RabbitMQ** — внутренняя шина сообщений для асинхронного взаимодействия между модулями и сервисами;
- **Синхронизация данных** — двусторонняя синхронизация с внешними системами (Active Directory, Exchange, Google Workspace и др.).

#### **Методы уведомлений и оповещений:**

- UI-уведомления — всплывающие оповещения в интерфейсе пользователя о статусе выполненных операций, изменениях данных и ошибках;
- Email-уведомления — рассылка уведомлений пользователям по электронной почте о важных событиях и изменениях;
- Push-уведомления — доставка push-сообщений пользователям через мобильные устройства и десктопные клиенты;
- Журналирование событий — ведение журнала событий и уведомлений для дальнейшего анализа и аудита.

#### **Методы централизованного управления пользователями:**

- Единая система управления пользователями — централизованное управление профилями, ролями, подразделениями, организациями и правами доступа;
- Массовое управление пользователями — массовое создание, изменение, удаление, перенос пользователей между подразделениями и организациями;
- Управление статусом пользователей — активация, деактивация, блокировка пользователей;
- Управление правами и ролями — назначение и изменение ролей, прав доступа, привилегий и разрешений.

#### **Методы резервного копирования и восстановления:**

- Регулярное резервное копирование — автоматическое создание резервных копий базы данных и файловой инфраструктуры (осуществляется средствами DevOps-инфраструктуры);
- Экспорт и импорт данных — возможность выгрузки и загрузки данных пользователей, настроек, профилей и других объектов;
- Восстановление данных — восстановление утраченных данных из резервных копий (осуществляется силами технической поддержки и DevOps).

#### **Методы интеграции с системами аутентификации:**

- Собственная аутентификация — встроенная система аутентификации пользователей с использованием паролей, двухфакторной аутентификации и одноразовых паролей;
- SSO/OAuth2/SAML2/Ldap/Kerberos — интеграция с внешними системами аутентификации и авторизации;
- Многофакторная аутентификация (MFA) — дополнительная защита учётных записей пользователей с помощью SMS, мобильного приложения или аппаратных токенов.

### **3.3 Структура программы с описанием функций составных частей и связи между ними**

**Корпоративный сервер 2024** представляет собой решение, объединяющее несколько подсистем. Его архитектура построена по принципам микросервисной модели с чётким разделением обязанностей между отдельными компонентами и слоями. Ниже представлено подробное описание структуры каждой подсистемы, входящих в состав **КС 2024**, с указанием их функций и взаимосвязей.

#### **3.3.1 Общая архитектура КС 2024**

**КС 2024** организован по слоистой архитектуре, включающей следующие уровни (Рисунок 1):

- Внешний API-слой — публичный интерфейс подсистем, принимающий HTTP-запросы клиентов и осуществляющий взаимодействие с внутренними сервисами;
- Сервисный доменный слой — центральный уровень оркестрации, содержащий бизнес-логику и координацию работы подсистем;
- Слой хранения данных — слой, ответственный за постоянное хранение информации в реляционной базе данных PostgreSQL;
- Фоновый processing-контур — асинхронный слой, выполняющий длительные и фоновые операции независимо от пользовательских запросов.

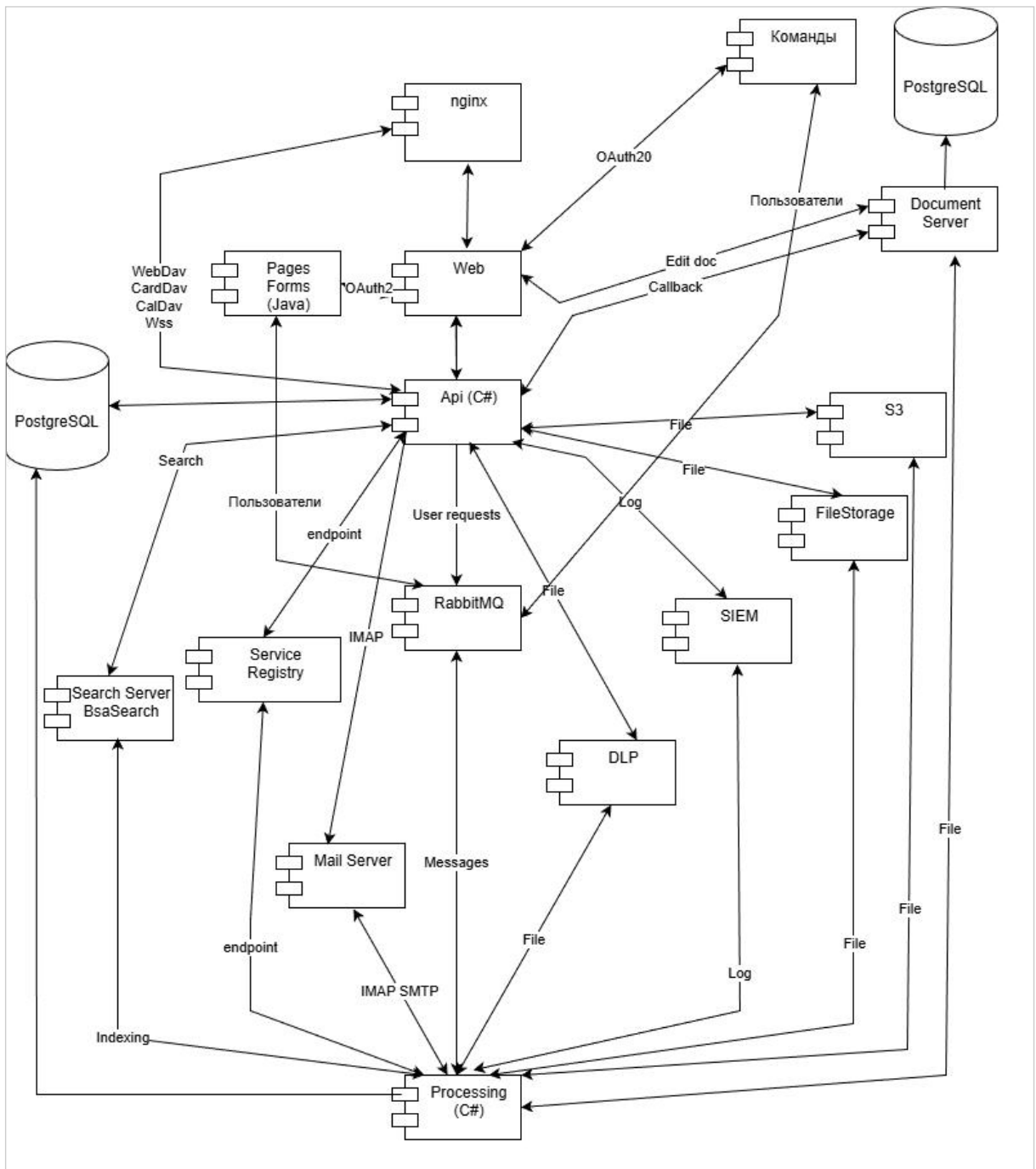


Рисунок 1 – Архитектура КС 2024

### 3.3.2 Подсистема «Календарь»

#### Логическая структура подсистемы

Подсистема «Календарь» организована по трехуровневой архитектуре:

- **Внешний API-слой** — принимает HTTP-запросы от клиентов и транслирует их во внутренние сервисы;

- **Сервисный доменный слой** — реализует бизнес-логику календаря, управляя событиями, календарями, участниками и интеграциями;
- **Слой хранения данных** — осуществляет постоянное хранение календарных данных в PostgreSQL;
- **Фоновый processing-контур** — выполняет асинхронные операции, такие как синхронизация Exchange и рассылка уведомлений.

Таблица 1 – Компоненты подсистемы «Календарь»

Компонент	Назначение	Основные связи
<b>Внешний API-слой</b>		
CalendarsController	Управление календарями: активация, очистка, сохранение, выборка подключённых календарей и видимость	CalendarService
EventsController	Управление событиями: создание, изменение, удаление, выборка, материалы, смена владельца	EventService
EventParticipantsController	Обработка ответов участников по событию	EventService
EventInvitationController	Работа с приглашениями из почты	EventInvitationService
ExchangeCalendarController	Синхронизация Exchange, подготовка списка Exchange-календарей	ExchangeCalendarSyncService
<b>Сервисный доменный слой</b>		
CalendarService	Координация работы с календарями: получение, создание, удаление, очистка, работа с временными зонами, шаринг	CalendarRepository
CalendarListVisibilityService	Настройки видимости календарей	CalendarListVisibilityRepository
EventService	Центральный сервис подсистемы: управление событиями, расчёт занятости, публикация уведомлений	EventRepository, EventRecurringRepository, etc.
EventInvitationService	Работа с email-интеграцией: разбор text/calendar, связывание сообщений с событиями	EventEmailInviteRepository
CalDavMapperService	Трансформация между внешними CalDAV/iCalendar и внутренними моделями	EventService
ExchangeCalendarSyncService	Синхронизация Exchange: обновление, mapping участников и вложений	EventService
<b>Слой хранения данных</b>		
CalendarRepository	Операции с календарями и настройками	PostgreSQL
CalendarListVisibility	Настройки видимости календарей	PostgreSQL

Компонент	Назначение	Основные связи
Repository		
EventRepository	Основное хранилище событий: выборка, фильтрация, маппинг моделей	PostgreSQL
EventRecurringRepository	Правила повторяемости событий	PostgreSQL
EventExceptionRepository	Исключения для повторяемых серий	PostgreSQL
EventAttendingRepository	Хранилище ответов участников	PostgreSQL
EventEmailInviteRepository	Связь между email-сообщениями и событиями	PostgreSQL
<b>Фоновый контур</b>		
ExchangeCalendarSyncMessage	Синхронизация Exchange	EventService
Отправка уведомлений	Рассылка email-уведомлений по изменениям событий	EventService
Batch-команды	Массовая отправка уведомлений	EventService

### 3.3.3 Подсистема «Контакты»

#### Логическая структура подсистемы

Подсистема «Контакты» организована аналогичным (п. 3.3.2) образом:

- **Внешний API-слой** — принимает HTTP-запросы от клиентов;
- **Сервисный доменный слой** — реализует бизнес-логику управления контактами, адресными книгами и группами;
- **Слой хранения данных** — осуществляет постоянное хранение контактных данных в PostgreSQL;
- **Фоновый processing-контур** — выполняет асинхронные операции, такие как индексация новых контактов.

Таблица 2 – Компоненты подсистемы «Контакты»

Компонент	Назначение	Основные связи
<b>Внешний API-слой</b>		

Компонент	Назначение	Основные связи
AddressBookController	Сохранение, получение, удаление списка связей контактов в адресной книге	AddressBookService
GroupsController	Управление группами контактов	GroupService
ContactsController	Поиск и получение деталей контактов	ContactService
<b>Сервисный доменный слой</b>		
AddressBookService	Управление адресными книгами: сохранение, удаление связей, получение информации по контактам	UserAddressBookRepository
GroupService	Управление группами контактов: создание, удаление, проверка прав, получение состава групп	GroupRepository
ContactService	Центральный сервис подсистемы: создание, получение, удаление контактов, проверка прав, взаимодействие с сервисами	ContactRepository
<b>Слой хранения данных</b>		
UserAddressBookRepository	Адресные книги, контакты, связи	PostgreSQL
ContactRepository	Контакты, пользователи	PostgreSQL
GroupRepository	Группы, отношения групп и контактов	PostgreSQL
<b>Фоновый контур</b>		
IndexingContactMessageHandler	Индексация новых контактов	ContactService

### 3.3.4 Подсистема «Почта»

#### Логическая структура подсистемы

Подсистема «Почта» организована по классической трёхуровневой архитектуре:

- **Внешний API-слой** — принимает HTTP-запросы от клиентов.
- **Сервисный доменный слой** — реализует бизнес-логику работы с письмами, папками, метками, отправкой и получением.
- **Слой хранения данных** — осуществляет постоянное хранение почтовых данных в PostgreSQL.
- **Фоновый processing-контур** — выполняет асинхронные операции, такие как синхронизация Exchange и рассылки уведомлений.

Таблица 3 – Компоненты подсистемы «Почта»

Компонент	Назначение	Основные связи
<b>Внешний API-слой</b>		
EmailAddressBookController	Сохранение, получение, удаление списка связей контактов в адресной книге	AddressBookService
EmailAttachmentController	Получение прав на вложение	LinkService
EmailCanonicalNameController	Работа с каноническими именами	EmailCanonicalNameService
EmailController	Основной контроллер по работе с почтой: взаимодействие с различными сервисами	EmailService
EmailDomainController	Работа с доменами	EmailDomainsService
EmailExportController	Постановка задач экспорта PST в очередь	EmailExportService
EmailLabelsController	Работа с метками писем	EmailLabelService
EmailServerController	Работа с настройками почтовых серверов	EmailServerService
<b>Сервисный доменный слой</b>		
AddressBookService	Управление адресными книгами: сохранение, удаление связей, получение информации по контактам	UserAddressBookRepository
LinkService	Управление ссылками: сохранение, удаление, проверка прав, получение объектов по ссылкам	CreateLinkMessageHandler
EmailService	Центральный сервис подсистемы: создание, получение, удаление писем, проверка прав, взаимодействие с сервисами	EmailMessageRepository
EmailFetchService	Получение новых писем с почтового сервера, обновление сообщений по imap папкам	EmailMessageRepository
EmailMessageOutboxService	Отправка массива сообщений по SMTP, отмена отправки, планирование отправки	EmailMessageSendItemsRepository
UserEmailService	Управление почтовыми аккаунтами: получение, удаление связей пользователей с почтовыми адресами	UserEmailRepository
<b>Слой хранения данных</b>		
EmailMessageRepository	Письма, контакты, вложения, приглашения	PostgreSQL
EmailFolderRepository	Папки, письма, вложения	PostgreSQL
UserEmailRepository	Почтовые аккаунты пользователей	PostgreSQL
EmailDomainsRepository	Почтовые домены	PostgreSQL
EmailMessageSendItemsRepository	Отправляемые письма	PostgreSQL

Компонент	Назначение	Основные связи
EmailMessageAutoActionsRepository	Автоответы и автопересылка	PostgreSQL
EmailLabelRepository	Метки писем	PostgreSQL
<b>Фоновый контур</b>		
RefreshEmailFolderItemsMessage	Синхронизация Exchange	EmailService
RefreshEmailFoldersMessage	Синхронизация папок	EmailService
SendEmailImapMessage	Отправка писем через IMAP	EmailService
DeleteUserEmailMessage	Удаление писем	EmailService
EmailBellNotificationMessage	Уведомления по звонку	Notification services
SendEmailInvitationsMessage	Отправка приглашений	EmailService

### 3.3.5 Подсистема «Проекты»

#### Логическая структура подсистемы

Подсистема «Проекты» организована по классическому трёхуровневому дизайну:

- **Внешний** API-слой — принимает HTTP-запросы от клиентов;
- **Сервисный** доменный слой — реализует бизнес-логику управления проектами, задачами, атрибуцией и связями;
- Слой хранения данных — осуществляет постоянное хранение проектной информации в PostgreSQL;
- **Фоновый** processing-контур — выполняет асинхронные операции, такие как экспорт и импорт проектов.

Таблица 4 – Компоненты подсистемы «Проекты»

Компонент	Назначение	Основные связи
<b>Внешний API-слой</b>		
ProjectController	Управление проектами, задачами и вложениями	ProjectService

Компонент	Назначение	Основные связи
ProjectTaskController	Управление задачами проекта	ProjectService
ProjectAttributesController	Управление пользовательскими атрибутами	ProjectService
ProjectTaskLinksController	Управление связями задач	ProjectService
<b>Сервисный доменный слой</b>		
ProjectService	Центральный сервис подсистемы: управление проектами, задачами, атрибутами, связями, вложениями, формирование отчётов	ProjectRepository, ProjectTaskRepository, etc.
LinkService	Генерация пригласительных ссылок, открытие документов из проектов	ProjectService
<b>Слой хранения данных</b>		
ProjectRepository	Проекты, статусы, типы задач	PostgreSQL
ProjectTaskAttributeRepository	Атрибуты задач, пользовательские атрибуты	PostgreSQL
ProjectAttachmentRepository	Вложения проектов	PostgreSQL
ProjectLabelRepository	Метки проектов	PostgreSQL
ProjectSettingsRepository	Настройки проектов	PostgreSQL
ProjectTaskActivityRepository	Активности задач	PostgreSQL
ProjectTaskLabelLinksRepository	Связи задач и меток	PostgreSQL
ProjectTaskRepository	Задачи, типы задач, пользователи	PostgreSQL
<b>Фоновый контур</b>		
ProjectSprintChangeBellNotificationHandler	Уведомления об изменении спринта	ProjectService
ProjectTaskChangedBellNotificationHandler	Уведомления об изменении задачи	ProjectService
ProjectTaskChangedMessageHandler	Уведомления об изменении задачи	ProjectService
ExportProjectTaskMessageHandler	Экспорт проектов в формате JSON/PDF	ProjectService
ImportProjectByJsonMessageHandler	Импорт проектов из JSON	ProjectService

### 3.3.6 Административный модуль

#### Логическая структура модуля

Административный модуль КС 2024 организован по двухуровневой архитектуре:

- **Пользовательский web-интерфейс** — отдельная SPA-оболочка для административной панели.
- **API Host** — общий серверный хост, обрабатывающий административные запросы и выполняющий основную бизнес-логику.
- **Централизованная база данных** — хранение пользователей, групп, подразделений, ролей, настроек и журналов.
- **RabbitMQ и Processing Host** — выполнение фоновых административных операций.
- **SSO Host** — использует настройки федеративной аутентификации, заданные через админку.
- **Внешние интеграции** — обеспечивают работу LDAP, почтовых сервисов, Telegram, DLP/ICAP, Syslog/SIEM и других подключаемых компонентов.

Таблица 5 – Компоненты административного модуля

Компонент	Назначение	Основные связи
<b>Пользовательский интерфейс</b>		
Admin SPA	Отдельная оболочка для административной панели	Browser, API Host
<b>API Host</b>		
Administrative controllers	Пользователи, роли, организации, настройки, аудит, безопасность	Services, Repositories
Authentication controller	Вход, выход, аутентификация, SSO, Kerberos	External systems, SSO Host
Statistics/Audit controller	Статистика, отчёты, мониторинг активности	Database, external integrations
Integration controller	Управление LDAP, SSO, Storage, DLP, ICAP, mail integration	External systems
License controller	Активация, лицензии, подписки	Database
<b>Централизованная база данных</b>		
Users	Пользователи, роли, подразделения, лицензии	PostgreSQL
Organizations	Организации, подразделения	PostgreSQL
Roles	Роли, права, привилегии	PostgreSQL
Licenses	Лицензии, подписки, активации	PostgreSQL
Logs	Журнал действий, аудита	PostgreSQL
Configurations	Настройки системы, интеграции	PostgreSQL
<b>RabbitMQ и Processing Host</b>		

Компонент	Назначение	Основные связи
Background tasks	Фоновые задания: LDAP sync, license checks, export reports, monitoring	Database, external integrations
<b>SSO Host</b>		
Federated auth settings	Конфигурация SSO, Kerberos, OIDC	Database, external identity providers
<b>Внешние интеграции</b>		
LDAP	Управление пользователями, организациями, ролями через LDAP	External directory servers
SSO/OAuth	Однократная аутентификация, интеграция с Identity Providers	External identity providers
Mail Servers	Управление почтовыми серверами, интеграция с почтовыми системами	External mail platforms
DLP/ICAP	Управление защитой данных, контроль утечек	External security solutions
Syslog/SIEM	Мониторинг событий, журналирование, оповещения	External logging solutions

### 3.3.7 Общий web-контур (Frontend SPA)

#### Логическая структура фронтенда:

Общая фронтальная часть **КС 2024** организована по современной архитектуре Single Page Application (SPA):

- **Пользовательский браузер** — среда выполнения клиентского приложения;
- **Reverse Proxy / Web Server** — раздаёт статическую сборку SPA и проксирует запросы /api во внутренние серверные сервисы;
- **Frontend SPA** — основная исполняемая оболочка, реализующая логику пользовательского интерфейса;
- **Маршрутизация и страницы** — управление навигацией между экранами;
- **UI-компоненты и формы** — представление интерфейсов и форм ввода;
- **Redux Store** — централизованное хранение состояния приложения;
- **API-адаптеры** — выполнение вызовов серверных методов;
- **Обработчик уведомлений** — приём внешних событий и инициация повторной загрузки данных;
- **LocalStorage** — хранение вспомогательных клиентских данных.

Таблица 6 – Компоненты фронтенда

Компонент	Назначение	Основные связи
Пользователь	Работает с веб-интерфейсом КС-24: выполняет вход, поиск, настройку, администрирование, экспорт и запуск операций	Браузер
Браузер	Среда выполнения клиентской части приложения	Веб-сервер, SPA, Web API
Веб-сервер / Reverse Proxy	Раздаёт статическую сборку SPA, обеспечивает HTTPS и проксирование API	Браузер, backend API, сертификаты
Frontend SPA	Основная оболочка приложения: логика интерфейса, маршруты, формы, запросы к API и обработка ответов	Маршрутизация, UI, Store, API-слой
Маршрутизация и страницы	Переключение между разделами: пользователи, роли, организации, профиль, настройки, статистика, активации, подписки	UI, Store, API
UI-компоненты и формы	Отображение интерфейсов: списки, карточки, формы ввода, уведомления, локализованный интерфейс	Пользователь, API, Store
Redux Store	Централизованное хранение состояния: настройки, сессии, фильтры, статус загрузки, ошибки	UI, API, localStorage
API-адаптеры	Выполнение REST-вызовов, передача токенов, обработка ошибок, загрузка файлов и сертификатов	Backend API
TipTap Редактор	Богатый текстовый редактор для составления писем: форматирование, ссылки, выравнивание	Форма составления письма
DOMPurify	Санитизация HTML-содержимого писем и calendar-приглашений перед рендерингом	Просмотр письма, calendar-интеграция
Опрос сервера (Polling)	Периодическое обновление счётчиков и списка писем каждые 30 секунд	API, Store
Backend API КС-24	Серверная бизнес-логика и централизованные данные	Auth, Folders, Messages, Settings, Integrations
Аутентификация	Вход в систему, выдача токенов, глобальные настройки	Клиент, внешние IdP
Адресная книга	Автодополнение получателей, управление контактами, избранные	Форма составления, API
Календарь	Обработка приглашений на встречи из писем, статусы (принято/отклонено)	Просмотр письма, API
Файловое хранилище (Disk)	Загрузка и сохранение вложений, генерация ссылок доступа	Форма составления, просмотр письма, API
Внешние системы	Внешние почтовые серверы, хранилища, SSO	Backend API
Обработка ошибок	React ErrorBoundary — перехват критических ошибок рендеринга,	Клиент

Компонент	Назначение	Основные связи
	вывод fallback-экрана с кнопкой перезагрузки, логирование в консоль	
i18next локализация	Перевод интерфейса (RU, EN)	UI-компоненты
LocalStorage	Настройки, feature-флаги, токен разработки	Redux Store, SPA
Router (react-router-dom)	Клиентская маршрутизация без перезагрузки страницы	SPA

### 3.4 Связи программы с другими системами (программами)

**Корпоративный сервер 2024** поддерживает интеграцию с системами:

1. Связка **КС 2024 – Сервер документов**: Сервер документов устанавливается отдельно и интегрируется с Корпоративным сервером 2024, обеспечивая онлайн-просмотр и редактирование текстов, таблиц и презентаций прямо из интерфейса **КС 2024**.
2. Подключённые внешние файловые сервисы: при такой связке можно из **КС 2024** просматривать и редактировать файлы, размещённые в подключённых внешних хранилищах:
  - Yandex.Disk;
  - Mail.ru Диск;
  - OwnCloud;
  - Nextcloud.

## 4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для обеспечения работоспособности **КС 2024** на сервере должны быть развёрнуты следующие программно-технические средства:

### Аппаратные требования:

- Процессор: не менее 2 ядер (рекомендуется 4 и более для работы под нагрузкой);
- Оперативная память: от 16 ГБ (рекомендуется 32 ГБ и выше для крупных инсталляций);
- Дисковая подсистема: SSD или SAS, объём зависит от количества пользователей и объёма хранимых данных;
- Сетевой интерфейс: гигабитный Ethernet.

### Программные требования:

- Операционная система:
  - Linux (Astra Linux Special Edition 1.7 и выше, РЕД ОС 7.3 и выше, другие совместимые дистрибутивы).
- Веб-сервер:
  - nginx или аналогичный HTTP(S)-сервер для публикации статических файлов и проксирования API-запросов.
- Система управления базами данных:
  - PostgreSQL (основная СУБД для хранения данных).
- Платформа выполнения серверных компонентов:
  - .NET (версия согласно требованиям подсистем).
- Средства управления сервисами:
  - systemd и/или supervisor.
- Асинхронная обработка задач:
  - RabbitMQ.
- Сетевое взаимодействие:
  - HTTP/HTTPS, IMAP, CalDAV, CardDAV, LDAP.

### Дополнительные компоненты:

- TLS/SSL-сертификат для обеспечения безопасной работы по HTTPS;
- Backend API-сервер **КС 2024** (обработка REST API-запросов);
- R7-Офис (Admin-модуль) — для централизованного управления пользователями, аутентификации, SSO.

**Клиентские требования:**

- Браузер: (Chrome, Firefox, Edge, Safari последних версий) с поддержкой ES2020+;
- JavaScript: включён;
- Cookies: разрешены (используются для аутентификации).

## 5. ВЫЗОВ И ЗАГРУЗКА

Для входа в систему **КС 2024** необходимо:

1. В адресной строке веб-браузера ввести адрес, присвоенный при установке и развертывании Системы.
2. В появившемся окне авторизации ввести логин и пароль пользователя (Рисунок 2).

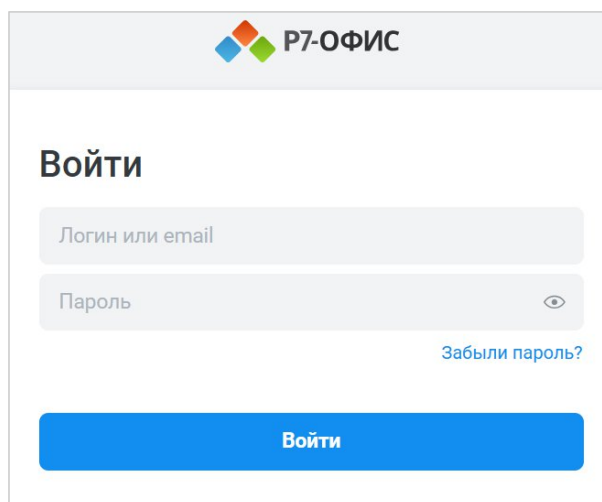


Рисунок 2 – Авторизация

В случае успешной авторизации, пользователь попадает на главную страницу с выбором модулей (Рисунок 3).

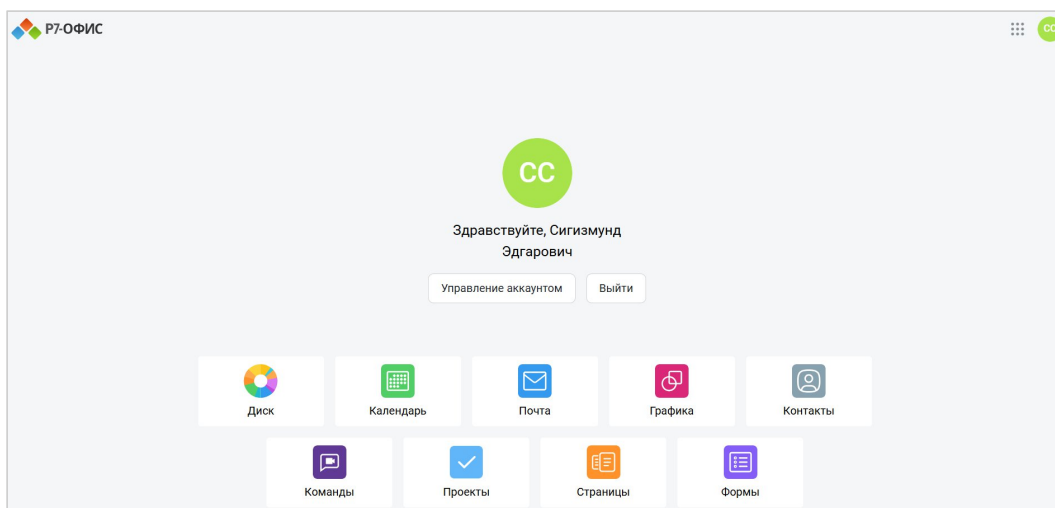


Рисунок 3 – Главная страница

## 6. ВХОДНЫЕ ДАННЫЕ

**Корпоративный сервер 2024** получает данные из различных источников, которые можно разделить на три основные группы:

- Входные данные от пользователей (через клиентский интерфейс);
- Входные данные от внешних систем (через API и интеграционные каналы);
- Входные данные от серверной инфраструктуры.

### 6.1 Входные данные от пользователей

#### Общие данные:

- Учётные данные для входа: логин, пароль, коды приглашения, коды подтверждения email, коды восстановления пароля;
- Поисквые строки и параметры фильтрации по подразделениям, ролям, другим критериям;
- Команды навигации и выбора разделов адресной книги, модулей системы.

#### Модуль «Почта»:

- Данные составляемого письма: получатели (To, Сс, Всс), тема, тело письма, вложения;
- Параметры расширенного поиска писем: поисковый запрос, тема, отправитель, получатель, диапазон дат, наличие вложений, статус прочитанности, важность, метки;
- Управление папками: создание, переименование, удаление, изменение структуры;
- Управление метками: добавление, удаление, изменение цвета и названия; параметры настроек: подписи, автоответчик (текст, даты активности), правила фильтрации (условия и действия), параметры пересылки;
- Параметры общего доступа: email пользователей, права доступа к письмам и папкам;

- Параметры отложенных писем: идентификатор письма, время восстановления;

параметры отложенной отправки: идентификатор письма, дата и время отправки;

- Файлы вложений для загрузки.

#### **Модуль «Календарь»:**

- Данные календаря: название, цвет, тип, статус, временная зона, период отображения;
- Данные события: наименование, описание, место, адрес, время начала, длительность, признак «весь день», приватность;
- Данные повторяемости: тип повтора, интервал, дни недели, день месяца, неделя месяца, дата окончания, число повторов;
- Данные участников: состав участников, статус участия, комментарии, ответы «иду/не иду/возможно»;
- Команды управления событиями: создать, изменить, удалить, отменить, восстановить, переместить событие, изменить владельца, добавить материалы;
- Фильтры выборки: интервал дат, набор календарей, текстовый поиск, пагинация, порядок сортировки.

#### **Модуль «Контакты»:**

- Данные карточки контакта: фамилия, имя, отчество, email, мобильный телефон, внутренний телефон, дата рождения, организация, подразделение, должность, адрес, примечание/описание;
- Команды добавления/удаления из избранного;
- Команды создания, изменения и удаления групп контактов;
- Команды добавления/удаления контактов в группы;
- Команды назначения прав доступа пользователям к группам.

#### **Модуль «Проекты»:**

- Данные проектов: название, описание, список статусов и типов задач;

- Данные спринтов: название, описание, статусы;
- данные по задачам: название, описание, вложения, комментарии, пользовательские типы.

#### **Управление:**

- JSON-команды и параметры фильтрации для создания и изменения пользователей, групп и подразделений, ролей, квот, политик и настроек;
- Данные для управления безопасностью и интеграциями: IP-списки, параметры LDAP, SMTP, SSO, Telegram, Syslog/SIEM и storage-конфигураций; файлы и multipart-запросы для отдельных административных сценариев (например, загрузка SSO-сертификата);
- Административные команды на запуск фоновых операций: LDAP-sync, экспорт из legacy CS, переиндексация, миграция хранилища и другие сервисные действия.

## **6.2 Входные данные от других систем через API и интеграционные каналы**

#### **Общие:**

- Глобальные настройки системы;
- Данные текущего пользователя;
- Иерархия папок и их счётчики;
- Списки писем с пагинацией и метаданными;
- Полное содержимое письма (заголовки, тело в HTML/plain-text, вложения);
- Данные контактов адресной книги (имя, email, телефон);
- Данные модульных доступов (ModuleSettings): URL, тип доступа; данные прав общего доступа (v1/Share/Access);
- Часовые пояса для calendar-интеграции;
- URL готовых отчётов и результатов серверных фоновых задач.

#### **Почта:**

- Данные Exchange-почты при синхронизации;
- Сведения о пользователях и настройках интеграции из общей платформы.

**Календарь:**

- CalDAV/iCalendar объекты событий; email-приглашения и отмены в формате text/calendar;
- Ответы участников на приглашения; данные Exchange-календарей при синхронизации.

**Контакты:**

- Список адресных книг;
- Список контактов и результаты поиска;
- Детальные карточки контактов;
- Сведения о группах контактов;
- Сведения о правах доступа и совместном использовании групп.

**Управление:**

- Справочники ролей, подразделений, адресов; данные пользователей, ролей, лицензий, квот; статистика и аудит;
- Конфигурации хранилищ и статусы миграций.

**6.3 Входные данные от серверной инфраструктуры****Общие:**

- Конфигурация окружения (appsettings, параметры доменов);
- Настройки БД;
- Настройки RabbitMQ;
- Системное время и временные зоны;
- Сетевые параметры;
- Параметры окружения (VITE\_);
- Доменная конфигурация и правила проху; заголовки reverse проху (в том числе служебный X-Module);
- Параметры запуска (appType);

- Данные TLS/HTTPS-сессии.

**Аутентификация:**

- Cookie и токены авторизации;
- Device ID пользователя.

**Системные сообщения:**

- Сообщения из очередей (RabbitMQ/ActiveMQ) — результаты конвертации, публикации изменений.

**Управление:**

- Конфигурационные файлы (config/.json\*);
- Лицензионные файлы (license.lic).

## 7. ВЫХОДНЫЕ ДАННЫЕ

**Корпоративный сервер 2024** формирует типы выходных данных, которые можно классифицировать по модулям и назначению.

### 7.1 Модуль «Календарь»

- API-ответы со списками календарей, событий, карточками событий (детали, участники, повторяемость, материалы, статусы);
- Результаты проверки занятости пользователей;
- Данные по подключённым календарям и их видимости;
- Данные о приглашениях и связанных с письмами событиях;
- Обновлённые статусы участия;
- Внутренние и внешние уведомления по событиям;
- Email-реакции на приглашения;
- Задания на синхронизацию с Exchange;
- Записи в БД об изменениях событий, повторях, исключениях, участниках;
- Служебные логи и записи о пользовательских действиях.

### 7.2 Модуль «Контакты»

- API-ответы со списками адресных книг, групп контактов, списком контактов;
- Результаты поиска контактов по подстроке;
- Детальные карточки контактов, сведения о правах доступа;
- Уведомления об успехе, ошибках, предупреждениях;
- Визуальные индикаторы загрузки и пустых состояний;
- HTTP-запросы на создание, изменение, удаление контактов и групп;
- Переходы в связанные модули (Mail, Calendar, Admin).

### 7.3 Модуль «Почта»

- API-ответы со списками почтовых ящиков, папок, писем;

- Результаты фильтрации писем по параметрам;
- Обновлённые статусы писем;
- Внутренние и внешние уведомления;
- Email-реакции на отправку писем;
- Задания на синхронизацию с Exchange;
- Записи в БД об изменениях писем, удалении, сохранении.

#### **7.4 Модуль «Проекты»**

- API-ответы со списками проектов, спринтов, задач, статусов, типов;
- Результаты фильтрации задач, проектов, спринтов.

#### **7.5 Административный модуль**

- REST/API ответы со списками и карточками пользователей, групп, ролей, организаций, квот, настроек, статистики, журналов действий.
- Изменения в БД: создание/обновление пользователей, ролей, прав, настроек, квот, записей аудита.
- Асинхронные команды и события в RabbitMQ для дальнейшей обработки.
- Экспортные данные и журналы: формирование файлов выгрузки, экспорт журналов действий.

#### **7.6 Уведомления и выходные сущности**

- Идентификаторы объектов (календаря, события, контакта, письма, задачи);
- Временные параметры (дата, время, длительность);
- Организатора и участников;
- Статус события, участия, письма или задачи;
- Данные о повторяемости (если применимо);
- Для интеграционных сценариев — UID, RECURRENCE-ID, SEQUENCE и другие поля календарного обмена.

### 7.7 Диагностические и служебные выходные данные

- Структурированные логи пользовательских действий и системных событий (DEBUG/INFO/WARN/ERROR);
- Телеметрия и отчёты об ошибках (например, Sentry);
- Данные о ходе асинхронных операций (статус, процент выполнения, признаки ошибок).

### 7.8 Примеры выходных данных

В таблице (Таблица 7) приведены основные типы выходных данных, которые формирует «Корпоративный сервер 2024» в процессе работы. Эти данные используются для взаимодействия между модулями, интеграции с внешними системами, а также для администрирования и мониторинга.

**Таблица 7 – Пример выходных данных**

Тип выходных данных	Описание / Примеры
API / REST ответы	JSON/XML с данными календарей, событий, контактов, писем, задач
Карточки объектов	Детализированные представления событий, контактов, писем
Уведомления	Внутренние (UI), внешние (email), RabbitMQ-события
Записи в БД	Изменения сущностей, журналы аудита
Логи	Структурированные файлы / консоль
Экспорт	CSV-отчёты, выгрузки журналов

