**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СОГЛАСОВАНО  C:\Users\user\Pictures\adb-sign.jpgк.т.н., доцент ДПИ ФКН  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Брейман  «16» мая 2025 г. | |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Павлочев  «16» мая 2025 г. | |
| |  |  | | --- | --- | | *Подп. и дата* |  | | *Инв. № дубл.* |  | | *Взам. инв. №* |  | | *Подп. и дата* |  | | *Инв. № подл* | RU.17701729.06.12-01 34 01-1 | | **СЭД С РАСШИРЕННЫМ ВЕРСИОНИРОВАНИЕМ**  **Руководство программиста**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.12-01 34 01-1-ЛУ** | | | | | | |
|  | |  | | | | |
| Исполнитель:  студент группы БПИ 219  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Н.В. Артемов /  «16» мая 2025 г. | | | | |
|  | | | | | | |
|  | | | | |  | |

**Москва 2025**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.06.12-01 34 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | *Подп. и дата* |  | | *Инв. № дубл.* |  | | *Взам. инв. №* |  | | *Подп. и дата* |  | | *Инв. № подл* | RU.17701729.06.12-01 34 01-1 | | **СЭД С РАСШИРЕННЫМ ВЕРСИОНИРОВАНИЕМ**  **Руководство программиста**  **RU.17701729.06.12-01 34 01-1**  **Листов 13** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2025**

ОГЛАВЛЕНИЕ

[1 ВВЕДЕНИЕ 4](#_Toc198237184)

[1.1. Наименование программы 4](#_Toc198237185)

[1.2. Краткая характеристика области применения программы 4](#_Toc198237186)

[2 НАЗНАЧЕНИЕ ПРОГРАММЫ 5](#_Toc198237187)

[2.1. Функциональное назначение 5](#_Toc198237188)

[2.2. Эксплуатационное назначение 5](#_Toc198237189)

[3 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ 6](#_Toc198237190)

[3.1. Минимальный состав аппаратных средств 6](#_Toc198237191)

[3.2. Минимальный состав программных средств 6](#_Toc198237192)

[3.3. Требования к персоналу (пользователю) 6](#_Toc198237193)

[4 ХАРАКТЕРИСТИКА ПРОГРАММЫ 7](#_Toc198237194)

[4.1. Архитектура системы 7](#_Toc198237195)

[4.2. Взаимодействие компонентов 10](#_Toc198237196)

[4.3. Схемы баз данных 10](#_Toc198237197)

[5 ВЫПОЛНЕНИЕ ПРОГРАММЫ 11](#_Toc198237198)

[5.1. Запуск программы 11](#_Toc198237199)

[5.2. Работа с приложением 11](#_Toc198237200)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc198237201)

# ВВЕДЕНИЕ

## 1.1. Наименование программы

**Наименование темы разработки:** «СЭД с расширенным версионированием»

**Наименование темы разработки на английском языке:** «EDM With Advanced Versioning»

## 1.2. Краткая характеристика области применения программы

«СЭД с расширенным версионированием» - веб приложение, система электронного документооборота. Основная область применения программы – ведение документов, что включает: создание, редактирование, согласование, подписание и дальнейший ввод в действие. Включает в себя функционал гибкой настройки всех процессов ведения документов. Отличается расширенной системой версионирования шаблонов, матриц согласования и документов, а также кросс-доменной функциональностью.

# НАЗНАЧЕНИЕ ПРОГРАММЫ

## 2.1. Функциональное назначение

Программа предоставляет возможность создавать различные шаблоны и гибко настраивать процессы ведения их документов, включая их схему состояний, атрибутивный состав и процессы согласования и подписания. Программа предоставляет функционал хранения, обработки и ведения документов по настроенным в шаблонах процессам.

Функциональное назначение серверной части – обработка запросов от клиентской части и хранение всех данных по документам, и связанными с ними настройками.

## 2.2. Эксплуатационное назначение

Программа является серверной частью приложения, системы для ведения внутреннего документооборота сотрудниками компании в локальной сети. Пользователи могут настраивать процессы документооборота, и, далее, работать с документами по настроенным процессам и конфигурациям. СЭД предполагает возможность интеграции с внутренними системами компании для унификации работы с договорами, заявками, compliance и любыми другими видами документов.

Эксплуатационное назначение серверной части заключается в предоставлении технологии и инфраструктуры, которые необходимы для использования через UI приложения, а также для обеспечения надежности, безопасности и масштабируемости системы.

# УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

## 3.1. Минимальный состав аппаратных средств

Доступ к интернету.

## 3.2. Минимальный состав программных средств

Веб-браузер с поддержкой HTML5,

Docker desktop или docker engine новейшей версии.

## 3.3. Требования к персоналу (пользователю)

Для корректной работы программы достаточно одного пользователя. Программист должен иметь базовые знания о веб программировании на любом языке программирования и базовые знания об HTTP и gRPC запросах. Так же необходимы базовые знания о ПО docker для корректного запуска API. Желательно уметь работать в Postman.

# ХАРАКТЕРИСТИКА ПРОГРАММЫ

## 4.1. Архитектура системы

Система построена на микро-сервисной архитектуре и имеет разделена на три крупных слоя:

1. Клиентская часть (не реализуется в рамках данной ВКР)
2. Api Gateway
3. Api

Всего предусмотрено 2 разные клиентские части:

1. Матрицы согласования
2. Документы

Вся работа с шаблонами и с ведением документов находится на клиентской части по Документам, в свою очередь клиентская часть «Матрицы согласования» отвечает только за настройку правил определения маршрутов согласования по шаблонам.

Каждые клиент (клиентская часть) работает с серверной частью через соответствующие сервисы Api gateway:

1. edm-document-generator-gateway
2. edm-entities-approval-rules-gateway

Взаимодействие между клиентской и серверной частью построено на Rest Api через синхронные запросы по Http протоколу.

Предусмотрены следующие сервисы бэкенд серверной части:

1. edm-document-generator

Основная бизнес-логика приложения – настройка и работа с шаблонами, ведение документов

1. edm-document-searcher

Read модель по документам для быстрого и эффективного поиска. Взаимодействует с edm-document-generator асинхронно для получения синхронизации данных и синхронно с edm-document-generator-gateway

1. edm-document-classifier

Сервис ответственный за классификацию шаблонов и их документы и за все необходимые справочники. Взаимодействет синхронно с edm-document-generator-gateway и edm-entities-approval-gateway для обогащения данных необходимой информацией.

1. edm-entities-counters

Сервис для настройки счетчиков и генерации регистрационных номеров. Синхронное взаимодействие с edm-document-generator и edm-document-generator-gateway.

1. edm-entities-approval-rules

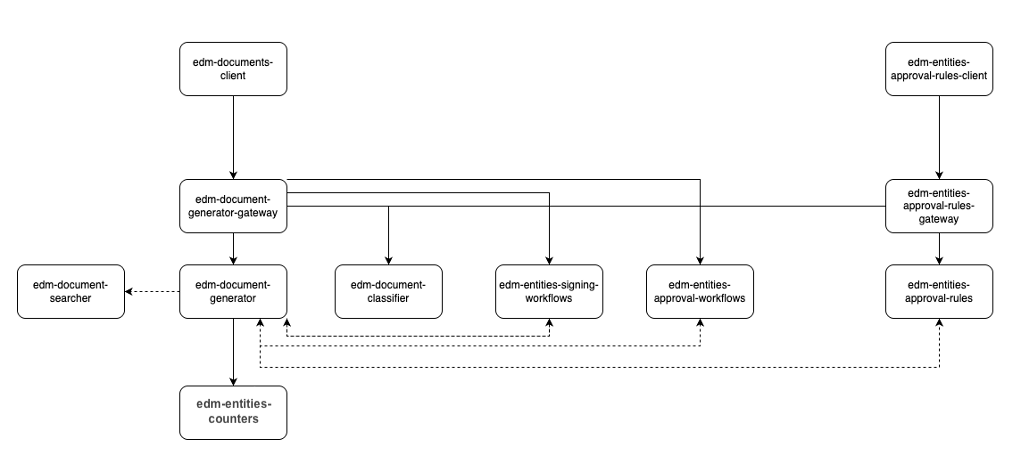
Сервис для работы с матрицами согласования. Работает асинхронно с edm-document-generator и edm-entities-approval-rules-gateway.

1. edm-entities-approval-workflows

Сервис для ведения документа по флоу согласования на основе построенного edm-entities-approval-rules маршрута согласования. Асинхронное взаимодействие с edm-document-generator и синхронное с edm-document-generator-gateway.

1. edm-entities-signing-workflows

Сервис для ведения документа по флоу подписания. Асинхронное взаимодействие с edm-document-generator и синхронное с edm-document-generator-gateway.

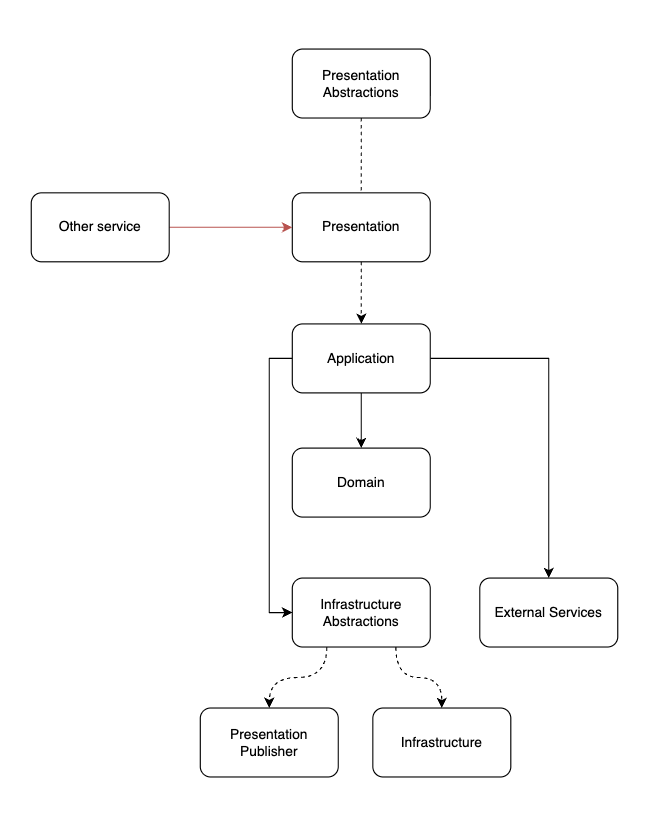


Приложение 1 – UML диаграмма общей архитектуры системы.

Два сервиса Api gateway реализуют луковую архитектуру и построенны на 3 основных слоях: Presentation, Core, ExternalServices для взаимодействия с клиентской частью, бизнес логики взаимодействия с клиенсткой частью и для взаимодействия с внешними сервисами соответственно.

Каждый из сервисов бэкенд серверной части реализует чистую архитектуру со следующими слоями:

1. Presentation Abstractions – определение интерфейса для внешних взаимодействие;
2. Presentation – реализация внешнего интерфейса;
3. Application – слой по обработке запросов через Mediatr, которые приходят из Presentation слоя;
4. Domain – основной слой, определяющий бизнес сущность сервиса с учетом Domain Driven Design, содержит в себе всю бизнес логику приложения;
5. Infrastructure – слой работы с хранилищами данных;
6. Presentation Publisher – слой для публикации асинхронных событий;
7. ExternalServices – слой для синхронного и асинхронного взаимодействия с внешними сервисами.

****

Приложение 2 – UML Диаграмма архитектуры микро-сервисов серверной части.

## 4.2. Взаимодействие компонентов

Синхронное взаимодействие между сервисами серверной части реализовано через gRPC, что обеспечивает эффективную работу за счет бинарной сериализации и низкой задержки, что критично для микро-сервисной архитектуры.

Асинхронное взаимодействие – наиболее предпочтительный тип межсервисного взаимодействия серверной части системы реализовано через Apache Kafka благодаря высокой пропускной способности и лучшей на рынке отказоустойчивости.

## 4.3. Схемы баз данных

Базы данных есть у каждого сервиса серверной части (Api). Для всех сервисов используется база данных PostgreSQL – является надежной и масштабируемой СУБД с богатым функционалом и наличием уникальных функций, таких как работа с jsonb.

Каждая база имеет 1 таблицу, которая названа по основной бизнес сущности сервиса и имеет следующие обязательные столбцы:

1. Id string,
2. Data bytea,
3. ConcurrencyToken timestamp.

База данных edm-document-searcher имеет схожую схему таблицы, однако тип поля Data вместо bytea – jsonb.

# ВЫПОЛНЕНИЕ ПРОГРАММЫ

## 5.1. Запуск программы

Для запуска программы необходимо перейти в директорию проекта и открыть ее в терминале. Далее необходимо ввести команду в терминал “docker-compose up -d”. Установка произойдет автоматически

## 5.2. Работа с приложением

При запуске серверная часть начинает принимать запросы. Далее есть 2 варианта взаимодействия с ней:

1. Отправлять запросы по gRPC и сообщения в Kafka серверной части при необходимости интеграции с внешними системами;
2. Использовать клиентскую часть (необходимо реализовать), которая будет работать с api-gateway: edm-document-generator-gateway и edm-entities-approval-rules-gateway.

Для тестирования приложения через свое Api или Postman программист может отправлять запросы к сервисам, используя документацию swagger и контракты слов Presentation.Abstractions.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.;
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Docker. [Электронный ресурс] – URL: <https://www.docker.com/> режим доступа: свободный (дата обращения: 13.03.25).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**