

Реферат

Данная работа посвящена разработке системы электронного документооборота с расширенным версионированием и гибкой настройкой процессов документооборота.

Современные организации активно используют системы управления документами, однако стандартные решения часто не обеспечивают необходимой гибкости для использования в различных сценариях, и детализированного контроля изменений, что может привести к потере важной информации и сложностям в согласовании документов. Разрабатываемая система ориентирована на повышение прозрачности и эффективности внутреннего документооборота, обеспечение надежного хранения и обработки всех документов.

Работа содержит: 31 страница, 3 главы, 13 иллюстраций и 5 источников.

Ключевые слова: система электронного документооборота, версионирование документов, домен, микро-сервисная архитектура, API.

Abstract

This work is dedicated to the development of an Electronic Document Management System with advanced versioning and flexible customization of document workflow processes.

Modern organizations actively use document management systems; however, standard solutions often lack the necessary flexibility for various scenarios and detailed control over changes, which can lead to the loss of important information and difficulties in approval processes. The developed system is aimed at increasing transparency and efficiency in internal document management, ensuring reliable storage and processing of all documents.

The work includes: 31 pages, 3 chapters, 13 illustrations, and 5 sources.

Keywords: electronic document management system, document versioning, domain, microservice architecture, API.

Основные определения, термины и сокращения

1. API (Application Programmin Interface) – интерфейс программы, набор контрактов и операций для взаимодействия с ним, описывающий взаимодействие.
2. СЭД (EDM, EDMS) – система электронного документооборота.

Москва 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
ГЛАВА 1. ПРЕДМЕТНАЯ ОБЛАСТЬ И СУЩЕСТВУЮЩИЕ РЕШЕНИЯ	9
1.1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.2. ОПИСАНИЕ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	9
1.2.1. <i>Контур.Диалок</i>	9
1.2.2. <i>СЭД Тезис</i>	10
1.2.3. <i>1С Документооборот</i>	10
1.2.4. <i>Астрал ЭДО</i>	10
1.2.3. ОПИСАНИЕ РАЗРАБАТЫВАЕМОГО РЕШЕНИЯ	10
1.2.4. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	11
1.2.4.1. <i>Гибко настраиваемый флоу ведения документа</i>	11
1.2.4.2. <i>Расширенное версионирование шаблонов, матриц согласования и документов</i>	11
1.2.4.3. <i>Сложность внедрения и поддержки</i>	12
1.2.4.4. <i>Динамическое построение маршрутов согласования</i>	12
1.2.4.5. <i>Простая интеграция с внешними системами</i>	13
1.2.4.6. <i>Высокая масштабируемость</i>	13
1.2.4.7. <i>Кросс-доменная функциональность</i>	13
Выводы	14
ГЛАВА 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	15
2.1. ОСНОВНЫЕ ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИИ	15
2.1.1. <i>Создание и конфигурация шаблона документов</i>	15
2.1.2. <i>Создание и конфигурация матрицы согласования для шаблона</i>	15
2.1.3. <i>Ведение документа по настроенному флоу</i>	15
2.2. АРХИТЕКТУРА СИСТЕМЫ	16
2.3. МЕТОДЫ И СРЕДСТВА РЕАЛИЗАЦИИ	19
2.3.1. <i>Схемы баз данных</i>	19
2.3.2. <i>Серверная часть и взаимодействие</i>	19
Выводы	19
ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	20
3.1. СИСТЕМА СЧЕТЧИКОВ	20
3.2. МЕНЕДЖЕР СПРАВОЧНИКОВ И КЛАССИФИКАТОР ДОКУМЕНТОВ	20
3.3. СИСТЕМА ПОИСКА ДОКУМЕНТОВ И СТЕППЕРА	22
3.4. СИСТЕМА ШАБЛОНОВ И ДОКУМЕНТОВ	23
3.4. СИСТЕМА СОГЛАСОВАНИЯ	25

3.5. СИСТЕМА ПОДПИСАНИЯ 26

3.6. СИСТЕМА МАТРИЦЫ СОГЛАСОВАНИЙ 27

Выводы по главе 29

ЗАКЛЮЧЕНИЕ 30

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31

Введение

В условиях постоянных изменений требований и внутренних процессов современные организации нуждаются в гибких и адаптивных инструментах для эффективного управления документами, однако существующие решения часто ограничены в возможностях и гибкости настройки, что затрудняет адаптацию процессов под специфические нужды компании.

В то же время компании стремятся к унификации рабочих процессов, чтобы снизить финансовые избытки и сложность эксплуатации множества разрозненных систем. Такое стремление объединить различные функциональные направления – таких, как обработку заявок технической поддержки и визирование контрагентов – стимулирует разработку кросс-доменного приложения, способного эффективно интегрировать и адаптировать бизнес-процессы в условиях постоянных перемен.

В процессе постоянного редактирования конфигураций и изменения бизнес-процессов возникают существенные трудности с поддержанием целостности данных и стабильной работоспособностью системы. При этом параметры, актуальные для работы с конкретным объектом в определённый момент времени, со временем теряют свою релевантность, что затрудняет работу со старыми данными, их анализ и аудит.

С учетом выявленных проблем, разработка кросс-доменного приложения, объединяющего в себе гибкую настройку бизнес-процессов, унификацию работы пользователей и системное управление изменениями с использованием расширенного версионирования, является крайне актуальной задачей для современных организаций. Такое приложение позволит обеспечить прозрачность и структурированность процессов, повысить эффективность работы сотрудников и создать устойчивую основу для дальнейшего развития бизнеса в условиях динамичной среды.

Цель работы: разработать кросс-доменную систему электронного документооборота с расширенным версионированием, ориентированную на гибкую настройку процессов, позволяющую адаптировать флюу под специфические требования и обеспечить многофункциональность.

Задачи работы:

- Анализ существующих решений;
- Формирование требований;
- Проектирование архитектуры;
- Реализация серверной части приложения;
- Реализация кросс-доменной функциональности;

СЭД с расширенным версионированием. Артемов Н. В.

- Тестирование веб-приложения;
- Разработка документации;

Глава 1. Предметная область и существующие решения

1.1. Описание предметной области

Разрабатываемая система предназначена для организации электронного документооборота и охватывает широкий спектр процессов, включая: создание, обработку, хранение и обмен документов в цифровом формате.

В современных организациях документооборот выходит за рамки договорной работы, он включает в себя: обработку заявок, служебных записок, отчетов и других сущностей, которые можно представить документами. Помимо работы с договорной документацией, система должна эффективно поддерживать процессы обработки заявок технической поддержки, внутренней корреспонденции и любых иных административных операций, которые характерны для разных подразделений компании. Это требует разработки гибкого и масштабируемого решения, способного адаптироваться под специфику каждого направления или департамента.

Особое внимание уделяется расширенному управлению версиями документов и конфигураций бизнес-процессов. В условиях постоянного обновления документов и шаблонов критически важно вести полную историю изменений. Такая функциональность обеспечивает возможность проведения аудита, повышает прозрачность работы системы и позволяет пользователям не только работать с актуальной информацией, но и при необходимости возвращаться к предыдущим версиям для анализа или восстановления данных.

В итоге, предметная область проекта охватывает комплекс задач по автоматизации, унификации и адаптации рабочих процессов документооборота, что способствует более эффективному и прозрачному управлению и учету в современных организациях.

1.2. Описание существующих решений

1.2.1. Контур.Диадок

Контур.Диадок – это СЭД, ориентированная преимущественно на юридическую документацию и взаимодействие между разными организациями. Основные возможности системы включают автоматизацию документооборота, обеспечение обмена документами с применением электронной подписи, а также интеграцию с налоговыми и бухгалтерскими сервисами. Система отличается простотой внедрения, высокой степенью защиты передаваемых данных и удобным интерфейсом для контроля процессов подписания и согласования документов. Ее преимущество – возможность обмена юридически значимыми документами в режиме онлайн, что существенно ускоряет документооборот в бизнесе.

1.2.2. СЭД Тезис

СЭД ТЕЗИС позиционируется как комплексное решение для автоматизации внутреннего документооборота организаций. Продукт ориентирован на обеспечение управления документами в рамках крупных корпоративных структур, где требуется интеграция множества бизнес-процессов. Среди ключевых возможностей – настройка маршрутов документов, разграничение прав доступа, возможность работы с различными типами документов (заявки, служебные записки, акты) и поддержка процессов согласования. СЭД ТЕЗИС подчеркивает гибкость настройки и масштабируемость системы, что делает её привлекательной для компаний, где документооборот тесно связан с бизнес-процессами разных подразделений.

1.2.3. 1С Документооборот

Решение от 1С – «1С Документооборот» – является одним из наиболее распространённых на российском рынке систем электронного документооборота. Оно интегрируется с другими продуктами 1С, что позволяет охватить широкий спектр задач: от ведения внутреннего документооборота до обмена документами с контрагентами. Продукт содержит в себе инструменты для создания, регистрации, хранения, поиска и согласования документов, а также весьма гибкую настройку бизнес-процессов. Отличительной чертой можно назвать возможность тесной интеграции с другими продуктами компании по бухгалтерскому и управленческому учёту.

1.2.4. Астрал ЭДО

Астрал ЭДО – это СЭД, направленная на интеграцию с юридическими сервисами и информационными системами организаций. Среди важных особенностей – поддержка электронной подписи, возможность обмена документами между разными организациями, а также возможность адаптации бизнес-процессов к специфике различных подразделений компаний. Система обладает широким функционалом по регистрации хранению и контролю документооборота и обеспечивает высокий уровень безопасности. Продукт отличается быстрым внедрением и стабильной работы системы без значительных затрат на инфраструктуру.

1.2.3. Описание разрабатываемого решения

Разрабатываемое приложение представляет собой кросс-доменную систему электронного документооборота, главная черта которой это очень гибкая настройка процессов документооборота и согласования документов в частности. Приложение позволяет настраивать процессы документов: создавать шаблоны документов, задавать их атрибутивный состав, валидации, уведомления и прочие настройки, и настраивать их состояния и жизненный цикл, включая этапы проверки, согласования и подписания. Особенностью системы является возможность динамического формирования маршрутов согласования по их матрицам согласования на основе настроек шаблона и текущих значений атрибутов.

Система хранит полную историю изменений шаблонов и любых других конфигураций, что обеспечивает прозрачность процессов и упрощает не только работу с приложением, но и качественный аудит. Благодаря обобщенной модульной архитектуре приложение адаптируется под специфические потребности различных подразделений, объединяя работу с договорами, заявками, служебными записками и другими документами в единой платформе. В итоге, данное решение обеспечивает эффективное и прозрачное управление документами, снижает затраты на эксплуатацию разрозненных систем и помогает простить и унифицировать процессы внутри компании.

1.2.4. Анализ существующих решений

1.2.4.1. Гибко настраиваемый флоу ведения документа

Подкритерии: настройка шаблонов, конфигурация жизненного цикла, возможность задавать правила переходов

1. **Контур.Диалок:** Частично
– Система предоставляет стандартные шаблоны и базовые механизмы маршрутизации, но возможности глубокого кастомного конфигурирования ограничены.
2. **СЭД ТЕЗИС:** Да
– Предназначена для крупных организаций, поддерживает достаточно гибкую настройку рабочих процессов, позволяющую адаптироваться под индивидуальные требования.
3. **1С Документооборот:** Частично
– Имеется возможность настройки шаблонов и маршрутов, однако глубина кастомизации часто зависит от опыта специалистов и особенностей интеграции в экосистему 1С.
4. **Астрал ЭДО:** Частично
– Решение поддерживает стандартные бизнес-процессы, но гибкая настройка флоу ограничена набором предустановленных опций.
5. **Разрабатываемое решение:** Да
– Система спроектирована с упором на максимальную гибкость: пользователи самостоятельно создают шаблоны, определяют атрибутивный состав и настраивают весь жизненный цикл документа с возможностью динамического изменения маршрутов.

1.2.4.2. Расширенное версионирование шаблонов, матриц согласования и документов

1. **Контур.Диалок:** Нет
– Основной акцент сделан на обмене юридически значимыми документами, без глубокого механизма управления версиями.
2. **СЭД ТЕЗИС:** Частично
– Поддерживается базовое версионирование документов, но расширенный учет изменений шаблонов и маршрутов согласования реализован не полностью.
3. **1С Документооборот:** Частично
– Возможна регистрация версий документов, однако управление изменениями в шаблонах и матрицах согласования ограничено стандартными механизмами.
4. **Астрал ЭДО:** Нет
– Функционал ориентирован на стабильный обмен документами, без акцента на расширенное управление версиями.

5. **Разрабатываемое решение:** Да

– Одно из ключевых преимуществ – ведение полной истории изменений для шаблонов, матриц согласования и самих документов, что обеспечивает прозрачность и упрощает аудит.

1.2.4.3. Сложность внедрения и поддержки

Подкритерии: требуемые технические ресурсы, обучение пользователей, интеграция в существующую инфраструктуру

1. **Контур.Диадок:** Да

– Облачное решение, которое внедряется быстро и требует минимальных затрат на поддержку.

2. **СЭД ТЕЗИС:** Частично

– Глубокая настройка и богатый функционал требуют значительных усилий при внедрении и обучении персонала.

3. **1С Документооборот:** Частично

– Интеграция с экосистемой 1С упрощает внедрение в среде 1С, однако в иных инфраструктурах может возникнуть сложность.

4. **Астрал ЭДО:** Частично

– Внедрение возможно, но потребуются адаптация для работы с конкретными бизнес-процессами организации.

5. **Разрабатываемое решение:** Да

– Система спроектирована с учётом современных технологий и модульной архитектуры, что обеспечивает простоту внедрения, масштабирование и поддержку.

1.2.4.4. Динамическое построение маршрутов согласования

Подкритерии: возможность адаптации маршрутов в зависимости от значений атрибутов и внешних условий

1. **Контур.Диадок:** Частично

– Поддерживает стандартные маршруты, но динамическая настройка, основанная на изменяющихся условиях, ограничена.

2. **СЭД ТЕЗИС:** Частично

– Возможности построения маршрутов присутствуют, однако адаптация маршрутов под индивидуальные сценарии реализована не во всей полноте.

3. **1С Документооборот:** Частично

– Имеется механизм маршрутизации, но динамическая адаптация маршрутов, зависящая от параметров документа, реализована частично.

4. **Астрал ЭДО:** Нет

– Решение не поддерживает динамическое построение маршрутов согласования.

5. **Разрабатываемое решение:** Да

– Фундаментальная особенность системы – динамическое формирование маршрутов согласования на основе заранее настроенных матриц и конфигураций шаблона, а так же значений атрибутивного состава данного договора и иного текущего контекста. Позволяет автоматически определять этапы согласования и их последовательность.

1.2.4.5. Простая интеграция с внешними системами

Подкритерии: наличие API, стандарты обмена данными, возможность интеграции с ERP/CRM и другими системами

1. **Контур.Диалок:** Да
– Предоставляет API и возможность пользоваться «облаком», что упрощает интеграцию
2. **СЭД ТЕЗИС:** Частично
– Имеет стандартные механизмы интеграции, однако требует дополнительной настройки
3. **1С Документооборот:** Частично
– Хорошо интегрируется в среде 1С, но взаимодействие с другими платформами требует доработок, и чаще всего эти доработки значительны
4. **Астрал ЭДО:** Частично
– Имеется API, но возможности интеграции ограничены определенными сценариями
5. **Разрабатываемое решение:** Да
– Система реализована с кросс-доменным функционалом и ориентирована на интеграцию с внутренними системами компании, поэтому обеспечена быстрая и простая интеграция с любыми подходящими системами.

1.2.4.6. Высокая масштабируемость

Подкритерии: возможность обработки большого объема документов, параллельная работа множества пользователей, модульность архитектуры

1. **Контур.Диалок:** Да
– Облачное решение, которое легко масштабируется под растущие потребности.
2. **СЭД ТЕЗИС:** Частично
– Предназначена для крупных организаций с большой нагрузкой, но сложность настройки ограничивает гибкость масштабирования.
3. **1С Документооборот:** Частично
– Масштабирование зависит от инфраструктуры 1С и может оказаться ограниченным в зависимости от количества используемых продуктов компании и возможных интеграциях.
4. **Астрал ЭДО:** Частично
– Решение масштабируется в своих рамках, однако при высоких объемах данных возможны ограничения.
5. **Разрабатываемое решение:** Да
– Используется модульная архитектура и современные стандарты разработки, система легко масштабируется под любую нагрузку и поддерживает большие объемы данных и большое количество одновременных пользователей.

1.2.4.7. Кросс-доменная функциональность

1. **Контур.Диалок:** Нет
– Система ориентирована преимущественно на юридически значимые документы и обмен между организациями, что ограничивает возможность работы с другими бизнес-доменами.
2. **СЭД ТЕЗИС:** Частично
– Решение в основном направлено на управление документами, однако возможность адаптации под другие бизнес-процессы реализована не в полной мере.
3. **1С Документооборот:** Частично
– Благодаря интеграции с ERP-системой 1С можно охватить и другие бизнес-процессы, но функционал в этой области зависит от конкретной конфигурации.

4. Астрал ЭДО: Нет
 - Ориентирован на документооборот юридической и бухгалтерской направленности, без явной поддержки мультидоменных сценариев.
5. Разрабатываемое решение: Да
 - Проект спроектирован как кросс-доменное решение, способное вести не только традиционные документы, но и заявки, служебные записки и другие информационные потоки, характерные для различных подразделений организации.

Выводы

Разрабатываемое решение демонстрирует явные преимущества по ключевым критериям по сравнению с существующими системами. Оно обеспечивает максимальную гибкость в настройке рабочих процессов, расширенное управление версиями, динамическое построение маршрутов согласования и простую интеграцию с внешними системами. Кроме того, современная архитектура проекта позволяет легко масштабировать систему под растущие потребности бизнеса, что является критически важным в условиях динамичной среды. В совокупности данные особенности делают разрабатываемое решение особенно привлекательным для организаций, стремящихся к оптимизации документооборота и повышению эффективности бизнес-процессов.

Глава 2. Проектирование системы

2.1. Основные пользовательский сценарии

2.1.1. Создание и конфигурация шаблона документов

Пользователь, обладающий правами администратора заходит в раздел управления шаблонами. Здесь инициируется процесс создания нового шаблона, где сначала определяется базовая структура будущего документа. Пользователь выбирает необходимые поля (текстовые, числовые, даты, выпадающие списки и т.д.), задаёт обязательность заполнения, а также устанавливает возможные значения по умолчанию и валидаторы для корректного ввода данных. После формирования атрибутивного состава шаблона начинается настройка схемы состояний: пользователь определяет жизненный цикл документа, настраивает переходы между состояниями, указывает условия валидации и настраивает уведомления для ответственных сотрудников. В рамках этого этапа также можно задать, в каких состояниях будет инициироваться процесс согласования или подписания. Так же настраивается нумератор для определения формата генерации регистрационного номера документа, поддерживается выбор и положение констант, дат и счетчиков, которые можно отдельно заводить и настраивать. После проверки и сохранения шаблона он становится доступным для дальнейшего использования при создании новых документов.

2.1.2. Создание и конфигурация матрицы согласования для шаблона

После настройки шаблона пользователь переходит в модуль матриц согласования. Здесь для выбранного шаблона инициируется создание новой матрицы, которая определит маршрут согласования документа на основе значений его атрибутов и внешних факторов. Пользователь задаёт правила: например, если сумма документа превышает определённый порог или выбран конкретный вид услуги, то документ направляется на согласование ответственному руководителю, далее – в финансовый отдел. Интерфейс позволяет легко добавлять, редактировать и комбинировать условия, устанавливая последовательность этапов согласования и подписания. Сохранённая матрица автоматически привязывается к шаблону, обеспечивая динамический выбор маршрута в зависимости от введённых данных при создании документа.

2.1.3. Ведение документа по настроенному флоу

После завершения настройки шаблона и матрицы согласования пользователь переходит к созданию нового документа. Система подгружает выбранный шаблон с уже настроенной структурой, атрибутами и жизненным циклом. Пользователь заполняет необходимые поля, и документ инициируется в начальном состоянии рабочего флоу. По мере обработки документа система автоматически проверяет условия валидации, отправляет уведомления и, при достижении состояний, требующих согласования, применяет соответствующую матрицу для определения маршрута. Одновременно с этим документ проходит этапы согласования и подписания, а все

действия фиксируются в истории изменений. Пользователи, участвующие в согласовании, получают уведомления и могут принимать решения, влияющие на переход документа в следующие состояния, пока он не достигнет финального (утверждённого или подписанного) статуса.

2.2. 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

2. 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

3. edm-document-classifier

Сервис ответственный за классификацию шаблонов и их документы и за все необходимые справочники. Взаимодействует синхронно с edm-document-generator-

gateway и edm-entities-approval-gateway для обогащения данных необходимой информацией.

4. edm-entities-counters

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

5. edm-entities-approval-rules

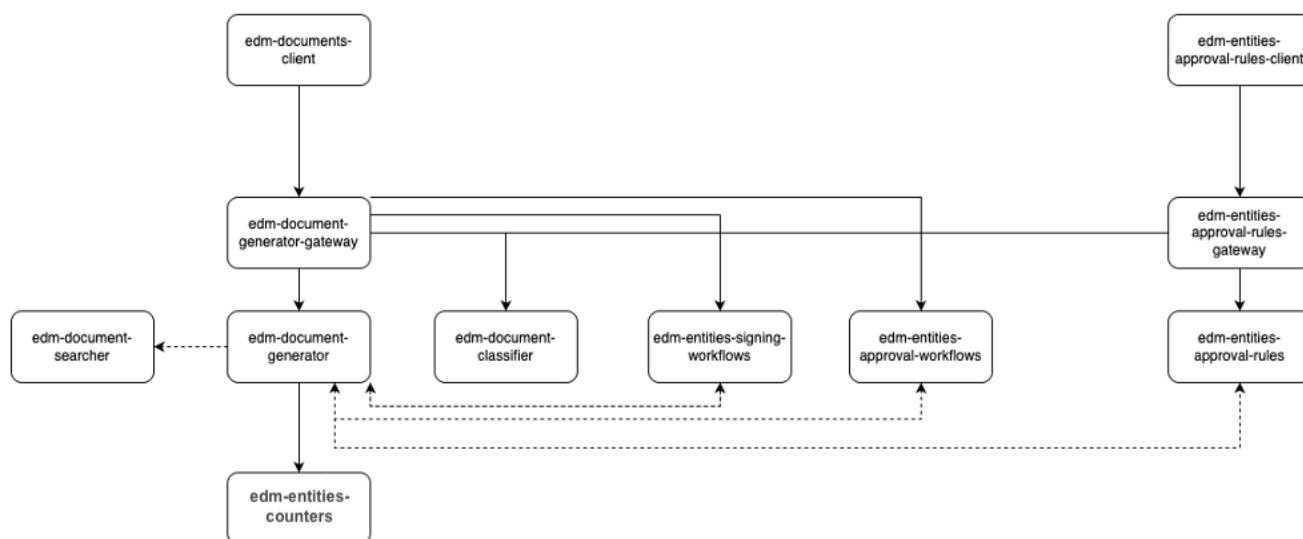
Сервис для работы с матрицами согласования. Работает асинхронно с edm-document-generator и edm-entities-approval-rules-gateway.

6. 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.

7. edm-entities-signing-workflows

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

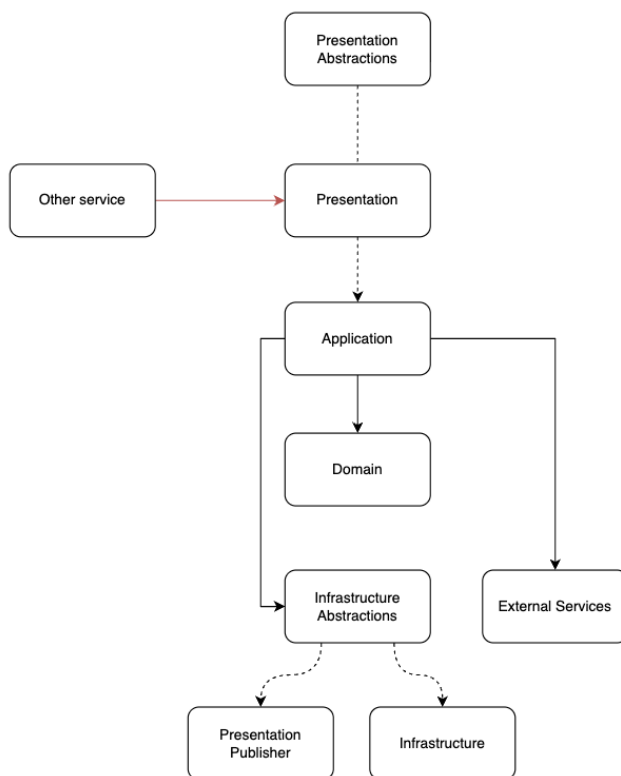


Приложение 1 – UML диаграмма общей архитектуры системы.

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

2.3. Методы и средства реализации

2.3.1. 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.

2.3.2. Серверная часть и взаимодействие

Все сервисы серверной части реализованы на платформе .Net на языке программирования C#. Предоставляет высокую производительность и удобство разработки.

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.

Синхронное взаимодействие между клиентской и серверной частью будет осуществляться через Rest Api, через https протокол.

Выводы

В данной главе предоставлено описание функциональных характеристик разработанной системы с помощью пользовательских сценариев, а также архитектуры разработанной системы. Затем рассмотрены использованные при разработке технологии и описан принцип их работы.

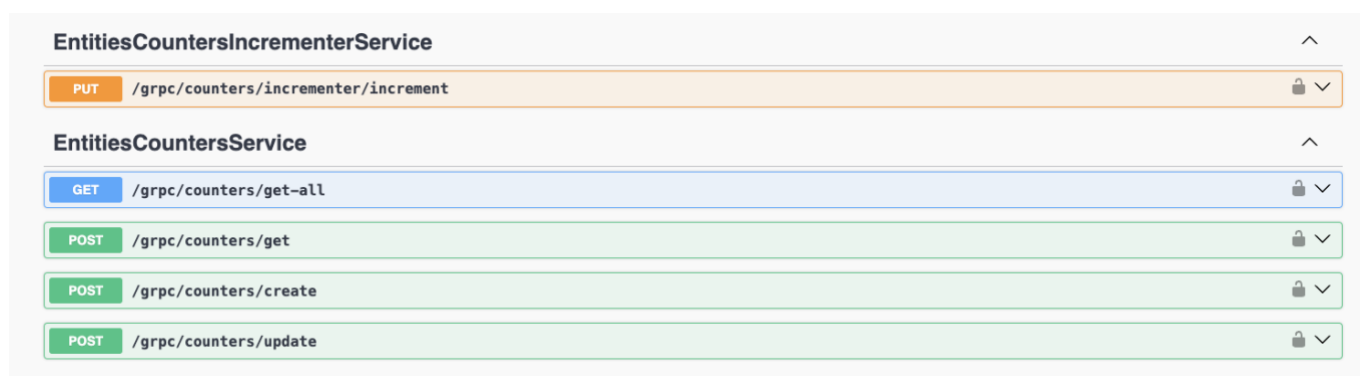
Глава 3. Программная реализация

3.1. Система счетчиков

Система счетчиков реализована для возможности автоматической инкрементальной генерации регистрационных номеров.

Формат регистрационного номера определяется в сервисе по работе с документами, а эта система отвечает за заведение и редактирование разных счетчиков в разных доменах.

При обращении к сервису `edm-entities-counters` по ручке получения нового номера – инкрементируется текущее значение в базе данных и это значение возвращается



Приложение 3. Иллюстрация документации Swagger. Интерфейс системы счетчиков

3.2. Менеджер справочников и классификатор документов

Система справочников реализована в формате хранения системами уникальных идентификаторов значений справочников. При необходимости получения информации о справочнике для пользователя на UI или для реализации бизнес-логики – вся информация о справочнике и о его значениях предоставляется провайдерами данных в сервисе `edm-document-classifier`. Справочники могут быть как динамическими с обращениями к внешним сервисам, так и фиксированными списками значений с описаниями. Примером динамического справочника может служить справочник ссылки на документ, таким образом идентификатор значения справочника – это идентификатор документа, а получение данных о значении происходит через обращения сервиса к `edm-document-generator`. Примером фиксированного справочника могут служить справочники валют или стран.

Справочники используются приложением в качестве типа атрибута документа и соответственно могут участвовать в процессе согласования и подписания, как и в любых других.

ReferencesService			^
POST	/grpc/references/get-all		🔒 ↓
POST	/grpc/references/get-reference-values		🔒 ↓
POST	/grpc/references/create-reference	🔗	🔒 ↓
POST	/grpc/references/update-reference		🔒 ↓
POST	/grpc/references/get-reference		🔒 ↓
POST	/grpc/references/get-all-references		🔒 ↓
POST	/grpc/references/create-reference-value		🔒 ↓
POST	/grpc/references/update-reference-value		🔒 ↓
POST	/grpc/references/bulk-upsert-reference-values		🔒 ↓
POST	/grpc/references/get-reference-value		🔒 ↓
POST	/grpc/references/get-all-reference-values		🔒 ↓
POST	/grpc/references/get-new-reference-type-id		🔒 ↓
POST	/grpc/references/migrate-reference-type		🔒 ↓
POST	/grpc/references/migrate-reference-value		🔒 ↓

Приложение 4. Иллюстрация документации Swagger. Интерфейс системы справочников

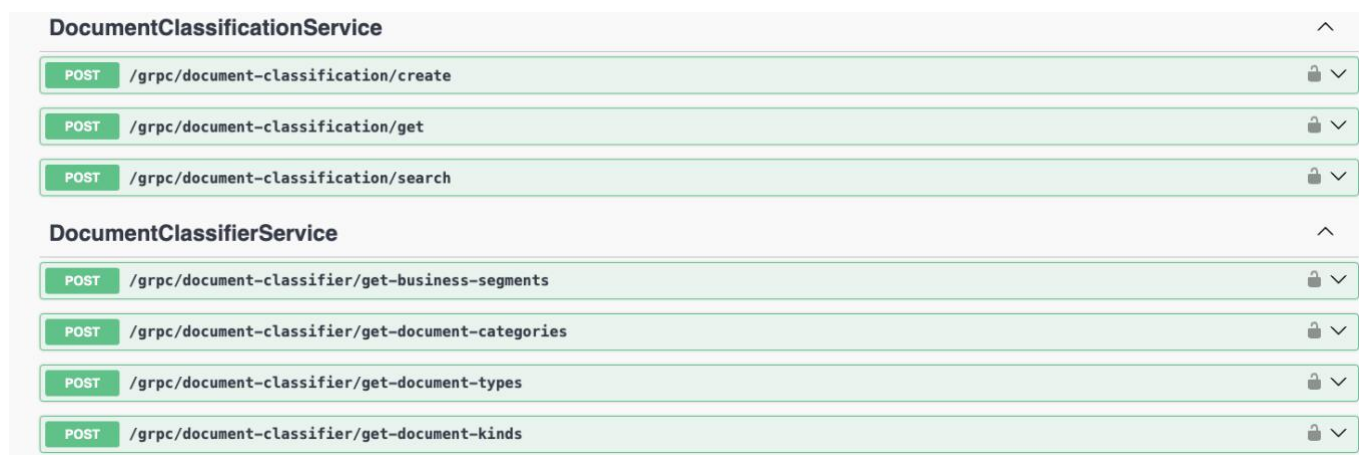
Сервис содержит в себе так же логику по работе с ролями в документе и в реестре, логика работы у них схожая с фиксированным справочниками, однако необходимы они не для использования в документах, а для настройки шаблонов и указывания на атрибуты, как на участвующие в тех или иных процессах или для вывода в реестр документов.

Так же этот сервис содержит в себе определение и описание доменов, в том числе системные атрибуты и роли, и прочая необходимая информация.

DocumentExternalLinksService			^
POST	/grpc/document-external-links/get-urls		🔒 ↓
DocumentReferencesService			^
POST	/grpc/document-references/get-types		🔒 ↓
POST	/grpc/document-references/search-values		🔒 ↓
DocumentRegistryRolesService			^
POST	/grpc/document-registry-roles/get-all-domains/v2		🔒 ↓
POST	/grpc/document-registry-roles/get-domain-by-id		🔒 ↓
POST	/grpc/document-registry-roles/get-domain-registry-roles-user-mark-allowed		🔒 ↓
POST	/grpc/document-registry-roles/get-domain-registry-roles		🔒 ↓
DocumentRolesService			^
POST	/grpc/document-roles/get-all		🔒 ↓
DocumentSystemAttributesService			^
POST	/grpc/document-system-attributes/get-all		🔒 ↓
DomainActionsService			^
POST	/grpc/domain-actions/get-domain-actions		🔒 ↓
DomainStructureService			^
POST	/grpc/document-classifier/get-domain-structure		🔒 ↓

Приложение 5. Иллюстрация документации Swagger. Интерфейс системы ролей и доменов

Одна из основных функций сервиса – функция классификатора для документов и шаблонов. При создании шаблона документа обязательны к заполнению комбинация бизнес-сегмента, категории типа и вида документа. Причем комбинаций этих может быть несколько. Далее по этой классификации можно однозначно определить шаблон документа и его документы. Необходимо это для того, чтобы пользователь мог при создании указать необходимую классификацию документа и получить созданный документ по настроенному шаблону.



DocumentClassificationService	
POST	/grpc/document-classification/create
POST	/grpc/document-classification/get
POST	/grpc/document-classification/search
DocumentClassifierService	
POST	/grpc/document-classifier/get-business-segments
POST	/grpc/document-classifier/get-document-categories
POST	/grpc/document-classifier/get-document-types
POST	/grpc/document-classifier/get-document-kinds

Приложение 6. Иллюстрация документации Swagger. Интерфейс системы классификаций

3.3. Система поиска документов и степпера

Система необходима для быстрого поиска документов. Проблема в том, что агрегат документа очень объемный и весит много, поэтому его получения или работа с ним достаточно долгая, а если этих объектов много, например если сделать поиск по сотням тысяч документов – все временные характеристики приложения нарушаются. Для этого было решено сделать систему-read-модель для документов, которая хранит данные в виде jsonb для быстрого поиска с использованием индексов в postgresql. Сервис также поддерживает расширенную фильтрацию и сортировку результата. Работает система на основе ролей в реестре, которые определяются доменом в сервисе edm-document-classifier. Весь функционал доступен как раз по указанным ролям.

Для большей разгрузки системы было также принято решение выделить степпер документа в этот же сервис и забрать эту бизнес логику с сервиса api-gateway для документов. Степпер документа это отображение маршрута ведения документа со всеми этапами, в том числе содержит подробную информацию об этапах согласования и подписания.



Приложение 7. Иллюстрация документации Swagger. Интерфейс системы поиска документов и степпера

3.4. Система шаблонов и документов

Основная часть всей системе СЭД, в этом сервисе вся логика по бизнес-процессу документооборота и концептуально делится эта система на две части: система шаблонов и система документов.

Система шаблонов необходима для конфигурации документов в целом и флоу ведения. В шаблоне конфигурируется:

- Атрибутивный состав (атрибуты могут быть базовых типов, справочниками и файлами);
- Валидаторы при обновлении атрибутивного состояния и смены состояния документа;
- Уведомления на действия с документами;
- Схема состояний, в том числе согласование и подписание;
- Доступы к документу в зависимости от его состояния теми или иными людьми;
- Формат генерации регистрационного номера.

По окончании конфигурации шаблона – отправляется запрос на создание матрицы согласования документа, и уже после конфигурации матрицы документ вводится в действие и далее по этому шаблону можно создавать документы.

Важно отметить, что атрибутивный состав шаблона, как и сам шаблон в целом, могут очень сильно меняться, могут меняться те же атрибуты, которые участвуют в согласовании. Поэтому реализовано расширенное версионирование шаблонов по критическим точкам. Для каждой критической точки существует своя версия матрицы согласования. Важно также отметить, что из каждой новой версии шаблона будет получаться совсем иной документ и изменения шаблона не ломают бизнес-процесс работы существующих документов.

DocumentsTemplatesDetailsService			^
POST	/grpc/documents-templates-details/get-details		🔒 ▼
DocumentTemplateService			^
POST	/grpc/document-templates/copy		🔒 ▼
POST	/grpc/document-templates/create		🔒 ▼
POST	/grpc/document-templates/delete		🔒 ▼
POST	/grpc/document-templates/migrate-all		🔒 ▼
POST	/grpc/document-templates/update		🔒 ▼
POST	/grpc/document-templates/get		🔒 ▼
POST	/grpc/document-templates/get-all		🔒 ▼
POST	/grpc/document-templates/get-all-documents-statuses		🔒 ▼
POST	/grpc/document-templates/process-support-requests		🔒 ▼

Приложение 8. Иллюстрация документации Swagger. Интерфейс системы шаблонов документов

Важно отметить, что атрибутивный состав шаблона, как и сам шаблон в целом, могут очень сильно меняться, могут меняться те же атрибуты, которые участвуют в согласовании. Поэтому реализовано расширенное версионирование шаблонов по критическим точкам. Для каждой критической точки существует своя версия.

Система документов тесно связано с шаблонами и по сути является логическим продолжением. Создается документ всегда по последней версии своего шаблона и получает настроенную конфигурацию и, далее, ведение документа происходит по полученной от шаблона настройке. Важно, что при изменении шаблона – схема состояний и остальные настройки уже созданных документов не изменятся. Документы хранят всю информацию о версии шаблона и используют ее для построения маршрута согласования.

В целом документ выглядит как обычная карточка, в которой можно заполнить поля и менять состояние. Однако есть некоторые виды сложных состояний: согласование, подписание и терминальное состояние активности. При запуске согласования управление состоянием уходит в систему флоу согласования и в зависимости от результата согласования в систему документов приходит запрос на модификацию. Та же самая логика реализована с подписанием. Отличительная черта статуса «Активен» – его терминальность, иными словами – активный документ оканчивает свой жизненный цикл и для любых изменений условий нужно или создавать дополнительное соглашение, или дополнительный документ, который будет перезаписывать результат.

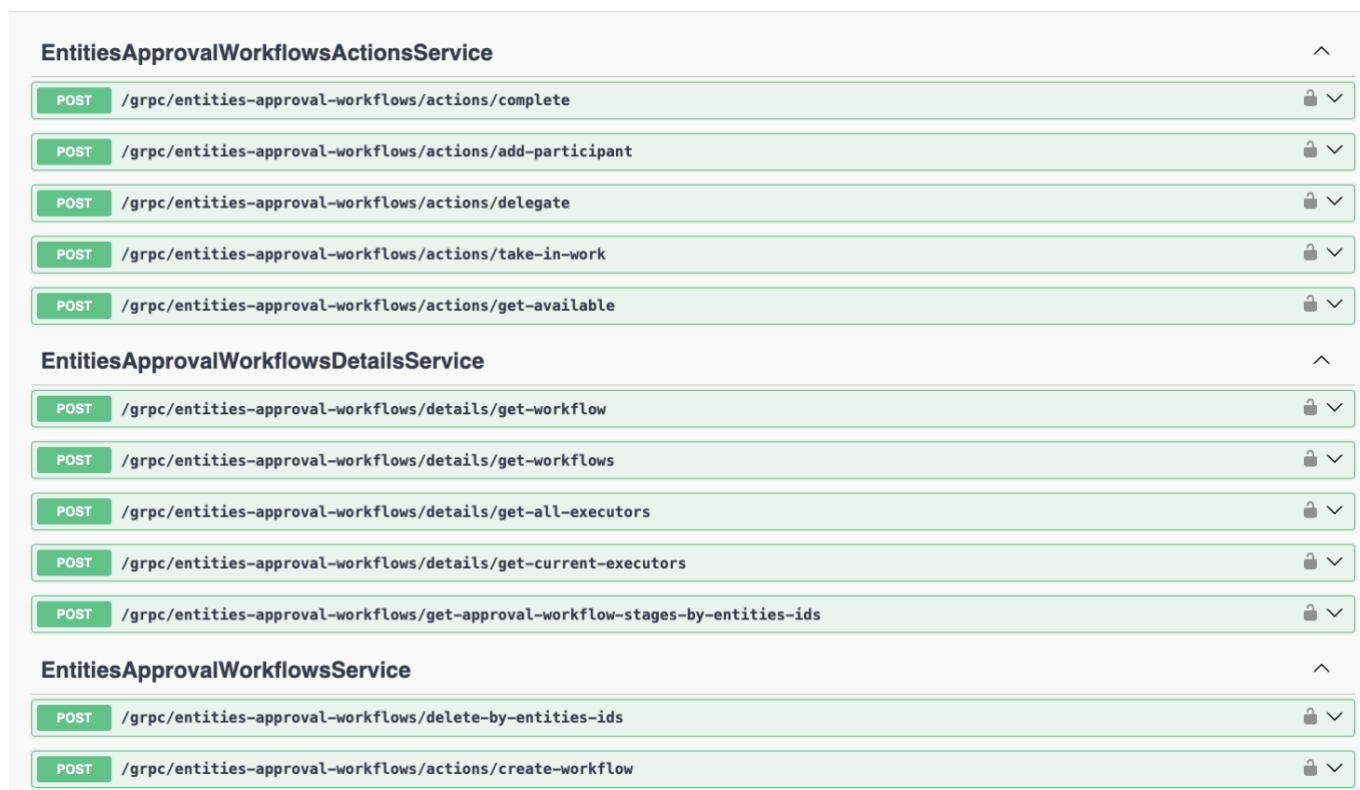
DocumentsDetailsService			^
POST	/grpc/documents-details/get-reference		🔒 ▼
POST	/grpc/documents-details/get-registration-number		🔒 ▼
DocumentsIntegrationsService			^
POST	/grpc/documents-integration/get-document_details		🔒 ▼
POST	/grpc/documents-integration/process-master-integration-request		🔒 ▼
POST	/grpc/documents-integration/process-interop-request		🔒 ▼
DocumentsService			^
POST	/grpc/documents/create-by-classification		🔒 ▼
POST	/grpc/documents/create-by-template-id		🔒 ▼
POST	/grpc/documents/create-by-template-system-name		🔒 ▼
POST	/grpc/documents/update		🔒 ▼
POST	/grpc/documents/integration-update		🔒 ▼
POST	/grpc/documents/batch-update/document-clerk		🔒 ▼
POST	/grpc/documents/get		🔒 ▼
POST	/grpc/documents/get-by-integration-id		🔒 ▼
POST	/grpc/documents/get-all		🔒 ▼
POST	/grpc/documents/get-shortest-path-to-completion		🔒 ▼
POST	/grpc/documents/validate		🔒 ▼
POST	/grpc/documents/get-documents-allowed-statuses		🔒 ▼
POST	/grpc/documents/batch-update/document-status		🔒 ▼
POST	/grpc/documents/delete		🔒 ▼

Приложение 9. Иллюстрация документации Swagger. Интерфейс системы документов

3.4. Система согласования

В систему согласования выделена бизнес логика по процессу согласования документа. При необходимости проведения согласования документа – в эту систему приходит запрос с построенным матрицей согласования маршрутом согласования. Однако это не финальная версия маршрута, потому данная система после получения маршрута его модифицирует – проводит замещения согласующих и добавляет этапы (например этап согласования заявки на согласование документа руководителем инициатора). Далее система рассылает оповещения о необходимости согласовать документ всем согласующими текущего этапа и получает запросы на передачу роли согласующего или согласование/отклонение. Передача роли согласующего может выглядеть как делегация или добавление нового согласующего именно для текущего документа. После согласования этапа – начинается следующий этап согласования следующей группой согласующих,

и так пока маршрут согласования не закончится. Далее система изменяет состояние документа в зависимости от результата визирования.



EntitiesApprovalWorkflowsActionsService	
POST	/grpc/entities-approval-workflows/actions/complete
POST	/grpc/entities-approval-workflows/actions/add-participant
POST	/grpc/entities-approval-workflows/actions/delegate
POST	/grpc/entities-approval-workflows/actions/take-in-work
POST	/grpc/entities-approval-workflows/actions/get-available
EntitiesApprovalWorkflowsDetailsService	
POST	/grpc/entities-approval-workflows/details/get-workflow
POST	/grpc/entities-approval-workflows/details/get-workflows
POST	/grpc/entities-approval-workflows/details/get-all-executors
POST	/grpc/entities-approval-workflows/details/get-current-executors
POST	/grpc/entities-approval-workflows/get-approval-workflow-stages-by-entities-ids
EntitiesApprovalWorkflowsService	
POST	/grpc/entities-approval-workflows/delete-by-entities-ids
POST	/grpc/entities-approval-workflows/actions/create-workflow

Приложение 10. Иллюстрация документации Swagger. Интерфейс системы согласования

3.5. Система подписания

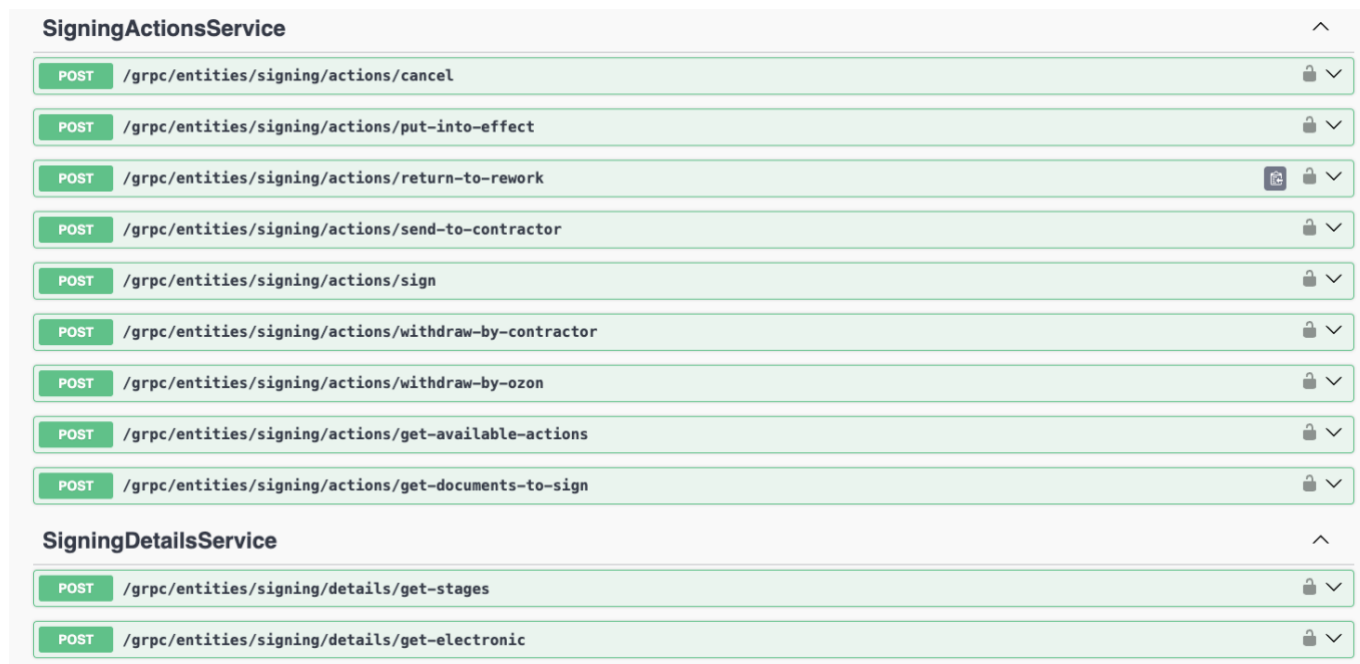
Система подписания состоит из двух компонентов: сервиса флоу подписания и сервиса для обмена с провайдером ЭДО. В рамках ВКР разработан интерфейс взаимодействия с провайдером ЭДО, но провайдер не подключен из-за сложностей с покупкой лицензии, поэтому далее пойдет речь про систему флоу подписания.

Система подписания схожа концептуально с системой подписания и так же при изменении на определенный статус документа получает запрос на начало ведения процесса, а при его окончании изменяет состояние документа в зависимости от результата.

Однако в случае подписания система получает настройки документа и его атрибутивный состав, а не построенный маршрут.

Система достаточно проста и всегда имеет два этапа – подписание компанией и контрагентом. В случае БДО (бумажное подписание) – необходимо вручную изменять состояния подписания. В случае ЭДО (электронное подписание) – состояние первого этапа компании изменяется при подписании документа с помощью цифровой печати. Переход из состояния

подписания контрагентом происходит при получении сообщения в брокере сообщений от провайдера электронных подписей.



The screenshot displays the Swagger API documentation for two services: **SigningActionsService** and **SigningDetailsService**. Each service is represented by a header with an expand/collapse arrow. Below each header is a list of API endpoints, each with a green 'POST' label, the full endpoint path, and a lock icon with a dropdown arrow.

Service	Method	Endpoint	Lock Icon
SigningActionsService	POST	/grpc/entities/signing/actions/cancel	✓
	POST	/grpc/entities/signing/actions/put-into-effect	✓
	POST	/grpc/entities/signing/actions/return-to-rework	✓
	POST	/grpc/entities/signing/actions/send-to-contractor	✓
	POST	/grpc/entities/signing/actions/sign	✓
	POST	/grpc/entities/signing/actions/withdraw-by-contractor	✓
	POST	/grpc/entities/signing/actions/withdraw-by-ozon	✓
	POST	/grpc/entities/signing/actions/get-available-actions	✓
	POST	/grpc/entities/signing/actions/get-documents-to-sign	✓
SigningDetailsService	POST	/grpc/entities/signing/details/get-stages	✓
	POST	/grpc/entities/signing/details/get-electronic	✓

Приложение 11. Иллюстрация документации Swagger. Интерфейс системы согласования

3.6. Система матрицы согласований

Матрица согласований и ее одноименная система одна из самых сложных во всем приложении и имеет отдельный Api-gateway и необходим отдельный UI под эту систему. Система отвечает за конфигурацию процесса согласования и построения матриц этого согласования. Концептуально матрица – это большой граф с вершинами, которые являются потенциальными этапами согласования со своими группами согласования, и ребрами, на которых указываются условия для построения маршрута. Таким образом матрица выглядит как дерево, и одна из его веток в последствии становится маршрутом согласования. Важно отметить, что группы тоже не фиксированы и составляются с теми же условными операторами и имеют функционал указания правил замещения сотрудников.

Что касается версионирования матриц, оно тесно связано с версионированием шаблонов, ведь при критических изменениях шаблонов может измениться процесс согласования или вовсе исчезнуть. При критическом изменении шаблона происходит попытка миграции матрицы, и, если она оканчивается успехом – создается новая матрица под текущую версию шаблона. Важно уточнить, что система матриц согласования не оперирует понятиями документов и шаблонов, а вместо них использует понятия сущностей и типов сущностей, где шаблон – это тип сущности. Под каждую версию типа сущности есть свой набор матриц согласования со своей внутренней группой версий.

В такой сложной системе очень легко ошибиться, а в процессе долгой и сложной настройки можно сломать согласование документов для текущей версии шаблона, поэтому реализована система активации и деактивации версий. При необходимости внесения изменений в матрицу согласования – создается новая не активная версия и в ней ведутся работы. Далее, когда версия матрицы настроена – она может быть активирована, и в таком случае – деактивируется версия, которая была активна. Версии можно создавать из любых существующих или в целом новую (чистый лист).

ApprovalConditionsOperatorsService			^
POST	/grpc/approval/conditions/operators/get-attributes-operators		🔒 ▼
POST	/grpc/approval/conditions/operators/get-logical-operators		🔒 ▼
ApprovalGraphsService			^
POST	/grpc/approval/graphs/create		🔒 ▼
POST	/grpc/approval/graphs/delete		🔒 ▼
POST	/grpc/approval/graphs/update		🔒 ▼
POST	/grpc/approval/graphs/get		🔒 ▼
POST	/grpc/approval/graphs/get-all		🔒 ▼
ApprovalGroupsDetailsService			^
POST	/grpc/approval/groups-details/get-foreign-groups-arguments-details		🔒 ▼
POST	/grpc/approval/groups-details/get-participant-source-attributes		🔒 ▼
ApprovalGroupsService			^
POST	/grpc/approval/groups/create		🔒 ▼
POST	/grpc/approval/groups/delete		🔒 ▼
POST	/grpc/approval/groups/update		🔒 ▼
POST	/grpc/approval/groups/get		🔒 ▼
POST	/grpc/approval/groups/get-all		🔒 ▼
ApprovalRulesDetailsService			^
POST	/grpc/approval/rules/details/get-details		🔒 ▼

Приложение 12. Иллюстрация документации Swagger. Интерфейс систем матрицы согласования: условные операторы, графы согласования, группы согласования

ApprovalRulesService			^
POST	/grpc/approval/rules/copy-entity-type		🔒 ▼
POST	/grpc/approval/rules/upsert-entity-type		🔒 ▼
POST	/grpc/approval/rules/delete-entity-type		🔒 ▼
POST	/grpc/approval/rules/find-route		🔒 ▼
POST	/grpc/approval/rules/validate-there-are-active-graphs		🔒 ▼
POST	/grpc/approval/rules/get-support-request		🔒 ▼
POST	/grpc/approval/rules/activate		🔒 ▼
POST	/grpc/approval/rules/create-new-version		🔒 ▼
POST	/grpc/approval/rules/deactivate		🔒 ▼
POST	/grpc/approval/rules/update		🔒 ▼
POST	/grpc/approval/rules/get-activation-audit		🔒 ▼
POST	/grpc/approval/rules/get-all		🔒 ▼
POST	/grpc/approval/rules/get		🔒 ▼
POST	/grpc/approval/rules/get-versions		🔒 ▼
POST	/grpc/approval/rules/search		🔒 ▼
POST	/grpc/approval/rules/migrate-support-request		🔒 ▼

Приложение 13. Иллюстрация документации Swagger. Интерфейс системы матриц согласования и их версионирования

Выводы по главе

Данная глава дает описание программной реализации серверной части системы «СЭД с расширенным версионированием». Разработанное решение реализует все заявленные функции.

Заключение

Процесс ведения документооборота может быть очень сложным и запутанным, и при недостаточной вариативности конфигурации, или прозрачности системы, которые обеспечиваются системами расширенного версионирования, может стать узким горлом бизнеса любого размера. Данный проект – хорошее решение для любой компании, которая хочет интегрировать понятную и настраиваемую систему внутреннего документооборота и учета в целом. Система позволяет не только вести договорную работу, но также настраивать работу в любом необходимом домене.

В ходе проекта был реализован весь заявленный функционал, цель была достигнута, а все задачи выполнены. Разработанный сервис представляет собой универсальный инструмент.

В качестве направления дальнейшего развития предполагается разработка пользовательских интерфейсов для систем матриц согласования и документов, а также интеграция провайдера ЭДО для электронного подписания.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Микросервисная архитектура // Microservice architecture URL: <https://microservices.io/>
- [2] 1С: Документооборот// 1С Документооборот 8 URL: <https://v8.1c.ru/doc8/>
- [3] Контур.Диалок // Официальный сайт ЭДО Диалок URL: <https://www.diadoc.ru/>
- [4] АСТРАЛ ЭДО // Астрал ЭДО для электронного документооборота URL: <https://astral.ru/products/astral-edo/>
- [5] СЭД ТЕЗИС // Систем электронного документооборота (СЭД) ТЕЗИС URL: <https://www.tezis-doc.ru/>