

Санкт-Петербургский государственный университет

***УСМАНОВ Артур Радикович***

Выпускная квалификационная работа

# Разработка сервиса автоматизации бумажного документооборота

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование  
информационных систем»*

Основная образовательная программа *СВ.5006.2019 «Математическое обеспечение и  
администрирование информационных систем»*

Научный руководитель:  
доцент кафедры системного программирования, к.т.н. Ю. В. Литвинов

Консультант:  
менеджер разработки АО «ПФ «СКБ Контур» Д. С. Будаев

Рецензент:  
инженер-программист АО «ПФ «СКБ Контур» Д. Д. Глебов

Санкт-Петербург  
2023

Saint Petersburg State University

*Artur Usmanov*

Bachelor's Thesis

# Development of service for paper document flow automation

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5006.2019 "Software and Administration of Information Systems"*

Scientific supervisor:  
C.Sc., System Programming chair docent Y. V. Litvinov

Consultant:  
Project manager at "SKB Kontur" D. S. Budaev

Reviewer:  
Software engineer at "SKB Kontur" D. D. Glebov

Saint Petersburg  
2023

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задачи</b>	<b>6</b>
<b>2. Обзор</b>	<b>7</b>
2.1. Сервисы управления документооборотом . . . . .	8
2.2. Текущий сервис . . . . .	11
<b>3. Реализация</b>	<b>13</b>
3.1. Модуль интеграции с системой управления продажами .	13
3.2. Доступ к действиям . . . . .	18
<b>4. Апробация</b>	<b>20</b>
<b>Заключение</b>	<b>22</b>
<b>Список литературы</b>	<b>23</b>

# Введение

Компания СКБ Контур занимается разработкой сервисов для бизнеса. В процессе продажи продуктов клиенты направляются в сервисные центры (СЦ) – офисы-партнеры, для удостоверения личности и оформления необходимых документов. Затем, с определенной периодичностью, СЦ формируют посылки, наполняя их документами, и отправляют их в головной офис компании.

В офисе, в отделе обработки документов (ООД), документы проверяются и отправляются в архив. Это обусловлено законодательством: контролирующие органы используют оригиналы документов, например, для разрешения конфликтных ситуаций или для проведения проверок. Кроме того, архивация позволяет клиентам восстановить документы при их утере. При невыполнении требований на компанию могут быть наложены санкции. Поэтому важно своевременно проверять корректность каждого присланного из СЦ документа и организовывать их хранение.

Текущие процесс и инструментарий не позволяют качественно проводить бумажный документооборот. СЦ лишь проинформированы о том, какие типы документов по каким процессам им следует отправлять в компанию. Соответственно ООД не обладает пониманием, какие документы были выпущены в СЦ, не могут проверять полноту присланных посылок. Отсутствует история операций: дата выпуска и отправки, имя сотрудника, сформировавшего документ. Фактически огромному количеству независимых партнеров предоставлена полная свобода на отправку документов.

Кроме того, на данный момент проверка осуществляется с помощью сервиса, по функциональности похожего на простую электронную таблицу. Это означает, что данные документов, результат проверки, положение в архиве вбиваются вручную. При этом, из-за отсутствия инструмента, объединяющего СЦ и ООД, каждый некорректный документ порождает отдельное электронное письмо или инцидент в службе поддержки. Всё это значительно замедляет проверку документов, на-

гружая сотрудников ООД монотонным трудом.

Поэтому в компании разрабатывается продукт, интегрирующийся с различными корпоративными системами, который объединил бы СЦ, ООД, архивную группу и других сотрудников, внес прозрачность и автоматизацию в бумажный документооборот. Такой сервис должен ускорить проверку документов, предоставив специализированный пользовательский интерфейс для приемки документов, вынесения вердиктов проверки и отправки в архив. Также новый проект призван сделать процессы более понятными: указать СЦ, какие конкретно документы нужно отправить и когда, отобразить ООД какие документы нужно получить и где находится каждый отдельный документ. Это позволит проще разрешать конфликтные ситуации и быстрее вынимать документы из архива по запросу контролирующих органов.

Сейчас уже реализован сервис для автоматизации оборота документов по одному из процессов в компании. Это веб-приложение, состоящее из клиентской части, написанной на Typescript, с использованием фреймворка React, и серверной части, написанной на языке С#. Коммуникация частей приложения происходит посредством Web API. Бэкенд интегрирован с корпоративной системой управления бизнес-процессом, поэтому все необходимые документы отслеживаются в сервисе. Кроме того, реализована частично автоматическая проверка документов, что значительно снижает нагрузку на сотрудников ООД [9]. Следующий этап развития проекта – оборот документов по процессу продажи продуктов компании.

# 1. Постановка задачи

Целью данной работы является разработка сервиса автоматизации бумажного документооборота. Для её выполнения были поставлены следующие задачи.

1. Рассмотреть существующие продукты для управления документооборотом.
2. Спроектировать и разработать модуль интеграции с корпоративной системой управления продажами.
3. Доработать бизнес-логику приложения, реализовав механизм управления доступом к действиям.
4. Провести апробацию новой функциональности с помощью пилотного запуска на несколько сервисных центров.

## 2. Обзор

Данная работа является частью существующего проекта по автоматизации документооборота в компании. На выбор собственной разработки, а не готовых инструментов управления документами или использования помощи сторонних компаний повлияло множество факторов.

В первую очередь проект зарождался как MVP, без четкого понимания, какие процессы должны быть автоматизированы и как выстроить обработку документов. Делопроизводство и документооборот регулируются часто изменяющимся законодательством [11]. Процессы, связанные с документами по продажам продуктов, сильно завязаны на также изменчивые правила внутри компании. Всё это формирует высокую степень неопределенности и изменчивости проекта. Выбрать конкретный инструмент означает быть ограниченным его возможностями. Оценить пригодность сторонних решений, возможности кастомизации, без достаточного понимания требований невозможно.

Кроме того, проект обладает спецификой, в основном из-за того, что документооборот производится не между сотрудниками компании, а между компанией и множеством сервисных центров. Сейчас уже существуют системы для управления партнерами и их сотрудниками. Авторизация пользователей производится через единую учетную запись компании, сервисные центры регистрируются в одном сервисе, а соответствие сотрудников указывается в другой.

Другой специфичный фактор – расположение сервисных центров на всей территории страны. Это означает, что необходимо предусматривать отслеживание посылок в пути у транспортной компании. Из-за большого количества независимых партнеров и их территориального удаления требуется разрабатывать систему мотивации для своевременного получения всех необходимых документов.

Не менее важной является экосистема продуктов и инфраструктура компании. Партнеры привыкли пользоваться множеством корпоративных систем с единым дизайном и пользовательским опытом. Сторонние

продукты не могут использовать единые элементы дизайна, из-за чего у сотрудников могут возникать трудности в работе с сервисом. Региональные менеджеры, непосредственно коммуницирующие с сервисными центрами, привыкли использовать определенные инструменты аналитики. Техническая инфраструктура позволяет быстро запускать новые проекты, использующие общепринятые в компании решения, а интегрировать стороннюю систему в существующее окружение может оказаться непросто.

Обработка персональных данных клиентов накладывает высокие требования по безопасности к сервисам. Использование внешнего продукта, даже с размещением на серверах компании, наверняка потребует дополнительной работы от специалистов по информационной безопасности.

## **2.1. Сервисы управления документооборотом**

Тем не менее, были рассмотрены существующие продукты, которые позиционируют себя, в том числе, как сервисы управления документами. Рассматривалась функциональность, связанная с созданием документов, в том числе из внешних систем, с управлением состоянием документа, с разграничением доступа на основе групп пользователей, с архивацией бумажных документов и их поиском, с нотификацией пользователей о необходимых действиях.

### **2.1.1. Directum RX**

Directum RX [14] предоставляет комплексный сервис для управления бизнес-процессами и документами. Немаловажно, что компания является российским разработчиком и продукт фактически доступен для использования. Также большим преимуществом является предоставление двух версий сервиса: облачной и серверной, то есть с установкой программного обеспечения в сети клиента.

В общих чертах, сервис состоит из множества модулей, в том числе включая модули управления документами, делопроизводства и долго-



временной архивации. Документы в сервисе можно создавать из файла, вложения в электронном письме или с помощью поточного сканера. При этом, данные документа могут быть автоматически распознаны и сопряжены с файлом. В визуальном редакторе, с помощью блоков и переходов, можно задать схему бизнес-процесса согласования так, чтобы документ переходил от одной группы ответственных к другой. При этом на каждом шаге можно настроить уведомление необходимых пользователей. Разграничение доступа к файлам происходит на основе ролей или отдельных сотрудников. В рамках модуля долговременного архивирования сервис предоставляет возможность управления хранением документов, то есть нумерации документов, генерации описи, формирования запросов в архивную группу.

Сложные специфичные бизнес-процессы можно разрабатывать в Low-code среде разработки Directum Development Studio на языке C#. Интеграционные возможности представлены дополнительными модулями для 1С, Directory Service, Microsoft Office и другими [13]. Кроме этого представлен Directum RX REST API с использованием протокола OData 4.0.

### **2.1.2. СЭД «ДЕЛО»**

Система электронного документооборота «Дело» [15] предоставляет решение для автоматизации работы с документами. Сервис, аналогично предыдущему, позволяет создавать документы из файлов, электронных писем и с помощью сканера. Несмотря на отсутствие возможности автоматического распознавания содержимого документа, можно приложить файл в формате XML со связанной информацией.

Бизнес-процесс согласования документов задается в графическом пользовательском интерфейсе. Отдельно акцентировано внимание на возможность настройки взаимодействия нескольких офисов между собой. Так, входящий документ можно не принять и отправить обратно, с указанием причины отклонения. Это похоже на целевой процесс принятия документов от сервисных центров в отделе обработки документов.

Сервис предоставляет возможность перемещения документов в ар-

хив. Поиск по связанным данным позволяет быстро найти необходимый документ в деле. Кроме того в сервисе представлена функциональность генерации отчетов по работе сотрудников. Большим преимуществом является заявленное наличие документированного API, которое позволит настроить интеграцию с уже существующими системами. Также представлена серверная версия сервиса.

### **2.1.3. ЭЛАР Контекст**

ЭЛАР Контекст [2] – модульная система управления корпоративным контентом. Компания предлагает решение для управления документами, с возможностью располагать программное обеспечение на сервере клиента или в облаке, на различных операционных системах, с использованием разных СУБД. Система аутентификации позволяет автоматически модифицировать списки пользователей. Для описания бизнес-процессов используется язык BPMN 2.0. Сервис обладает возможностью расширять функциональность с помощью клиентских плагинов, которые могут работать как на стороне клиента, так и на стороне сервера. Создание документов возможно по событиям из других систем с помощью модулей-адаптеров. Отдельно представлена система управления архивом, которая позволяет организовать работы по инвентаризации документов и их выемке из архива, в том числе с использованием RFID-меток.

### **2.1.4. BPMN**

Отдельно хочется упомянуть нотацию для моделирования бизнес-процессов BPMN [1]. Спецификация описывает несколько категорий элементов, с помощью которых бизнес-процесс можно представить в виде диаграммы. Более того, такой диаграмме соответствует описание на языке WS-BPEL [8] или в формате BPMN 2.0 XML. Это позволяет исполнять BPMN-моделей в специализированных системах управления бизнес-процессами (например, [3], [17]).

Наглядная описание правил позволяет вносить изменения не толь-

ко разработчикам, но и аналитикам, которые лучше знакомы с бизнес-процессами. Это, в свою очередь, уменьшает количество ошибок и сокращает время на внесение изменений.

## 2.2. Текущий сервис

На данный момент реализован сервис для поддержания одного из процессов бумажного документооборота в компании. Это веб-приложение для сотрудников сервисных центров, отдела обработки документов, экспертов и архивной группы. Бэкенд сервиса разделен на несколько отдельных частей.

- Модуль интеграции, который отвечает за отслеживание событий во внешней системе и создание документов в соответствии с ними [9].
- Модуль мониторинга поточного сканера, который, с помощью класса `FileSystemWatcher`<sup>1</sup>, отслеживает созданные документы в файловой системе и отправляет их на сервер приложения.
- Модуль синхронизации архива, который сохраняет информацию о проверенных и сформированных в дела документах во внешнюю систему управления архивом.
- Множество фоновых процессов, которые пересчитывают долги сервисных центров, отправляют данные для отчётов, синхронизируют информацию о партнёрах и так далее.
- Web API для взаимодействия пользователя с сервисом. В нём реализованы возможности формирования упаковок и посылок с документами, заполнения информации об отправке, приёмки и проверки документов, делопроизводства. Информация о результате проверки в отделе обработки документов автоматически передаётся сервисному центру.

---

<sup>1</sup>`FileSystemWatcher` Class | Microsoft Learn – <https://learn.microsoft.com/en-us/dotnet/api/system.io.filesystemwatcher> (дата обращения 02.05.2023)

### 2.2.1. Хранение данных

Несмотря на разделение сервиса на части, в приложении используется одна общая база данных MongoDB<sup>2</sup>. С одной стороны это повышает связность частей между собой: изменения в схеме базы данных затрагивает сразу все модули. С другой стороны, это сильно упрощает разработку, потому что сбор данных с нескольких источников, выполнение распределенных транзакций, обработка ошибок требуют значительных трудозатрат [12], [5]. Поскольку весь сервис разрабатывает одна команда, то использование отдельных СУБД для разных частей приложения не целесообразно.

Кроме того MongoDB реплицирована на нескольких дата-центрах с одной primary нодой и двумя secondary [6]. Это повышает надежность хранения данных, а также позволяет частично распределить нагрузку. Например, запросы для формирования больших отчетов выполняются на одной из secondary нод, поэтому производительность Web API не затрагивается.

Коллекции в базе данных соответствуют основным сущностям в сервисе: **PaperDoc**, **Pack**, **Parcel**, **User**, **ServiceCenter** и так далее. Основные сущности и связи между ними изображены на ER-диаграмме 1.

Кроме этого, для хранения и передачи данных в сервисе используется корпоративная очередь сообщений. Она позволяет настроить асинхронное взаимодействие между частями приложения и применяется в основном для постепенной загрузки данных документов из внешних систем. Коммуникация реализована методом pull, то есть клиенты периодически запрашивают новые сообщения. При этом гарантируется доставка от отправителя к обработчику. Очередь предоставляется в виде сервиса, не требует настройки и развертывания, имеет клиентов для нескольких языков программирования, поддерживает стратегии отложенного исполнения задач. Поэтому для реализации интеграции с внешней системой была выбрана именно она.

---

<sup>2</sup>MongoDB Community Server – <https://www.mongodb.com/try/download/community> (дата обращения 02.05.2023)

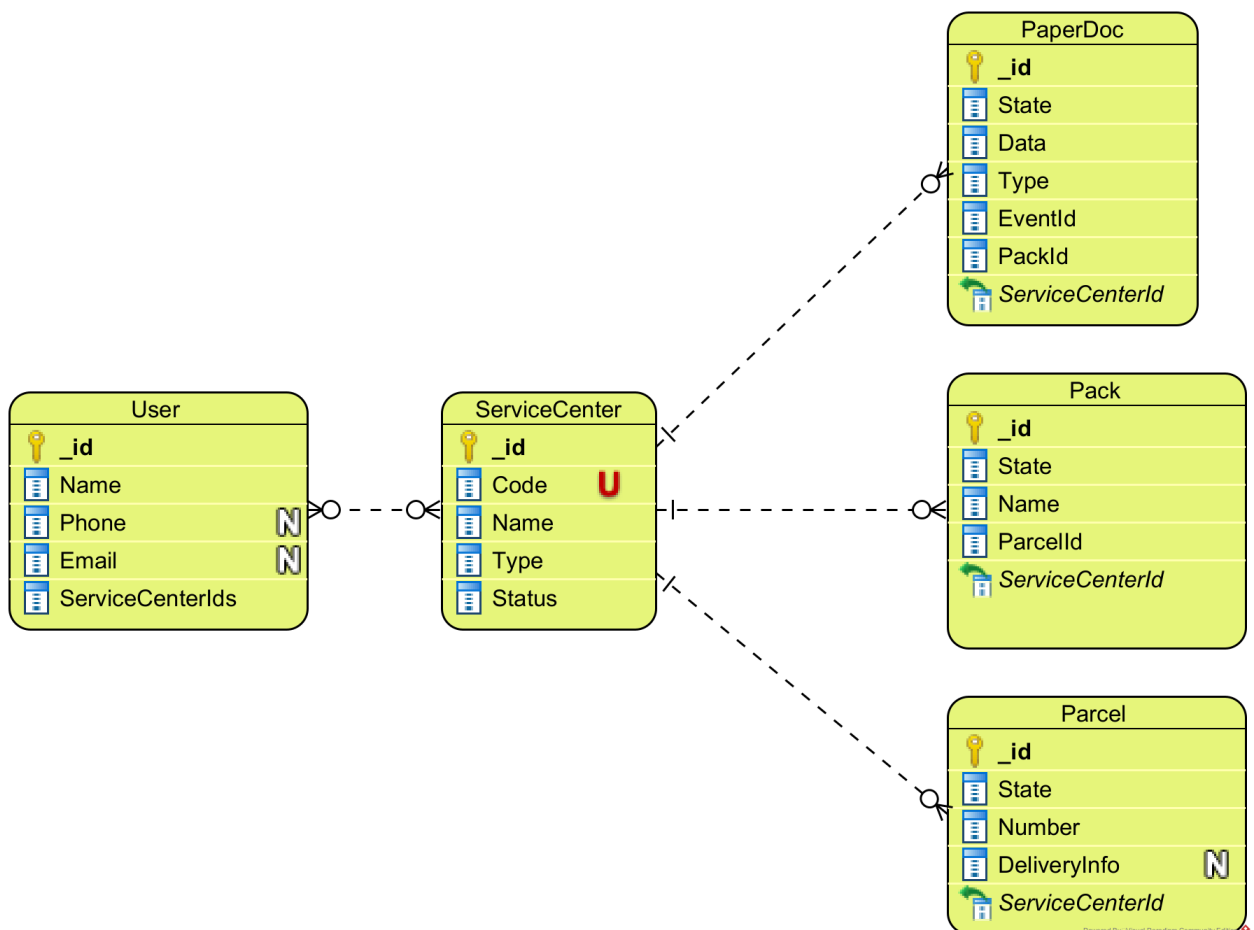


Рис. 1: ER-модель основных моделей в сервисе

## 3. Реализация

### 3.1. Модуль интеграции с системой управления продажами

Основной объём работы заключается именно в проектировании и разработке модуля интеграции сервиса документооборота с корпоративной системой продаж. Суть интеграции заключается в отслеживании действий в системе и в реакции на них, путем создания документов в сервисе. События во внешней системе представлены в виде ленты новостей. Необходимо отслеживать две ленты: по документам и по счетам. Дело в том, что сначала создается счет, к которому затем прикрепляются документы. В сервисе необходимо отображать данные как самого документа, так и счета, к которому он привязан.

Таким образом, в базу данных необходимо добавить две сущности: документ и счёт. Поскольку в базе данных уже существует коллекция с документами **PaperDocs**, решено не создавать другую модель, а модифицировать существующую, добавив новое поле **LegalData**. Хранение счетов реализовано в двух местах одновременно: в отдельной коллекции **Bills** и внутри документа в поле **LegalData.Bill**. Это позволяет искать документы по данным счёта, но требует транзакционного обновления счетов на всех документах и в коллекции **Bills**. Добавленные модели изображены на рисунке 2.

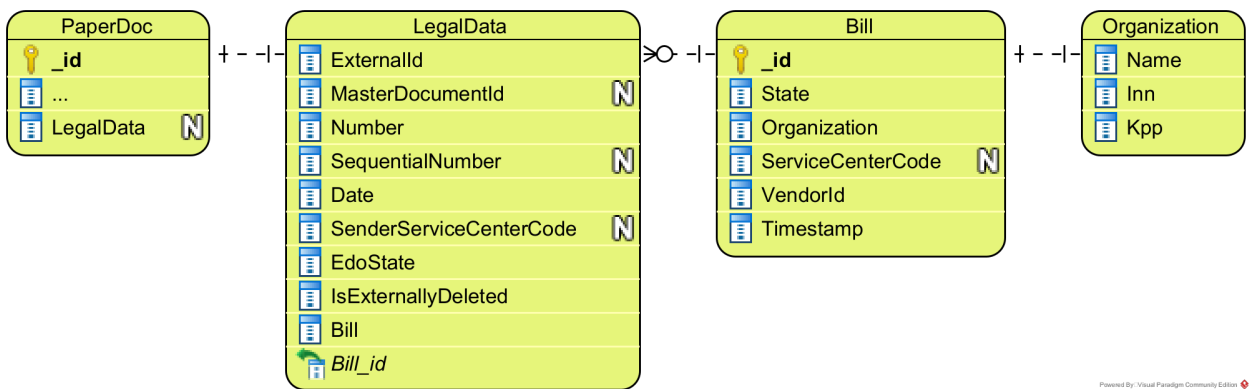


Рис. 2: ER-модель добавленных моделей

Для разделения ответственности и возможностей масштабирования, отслеживание лент разбито на два отдельных демона. Первый только читает события во внешней системе, делая необходимый вызов метода API, и создает задачи на обработку этих событий в корпоративной очереди сообщений. Второй запрашивает задачи из очереди и обрабатывает их, сохраняя в базу данных необходимую информацию. Процесс отслеживания ленты счетов изображен на рисунке 3.

Обработка новостей по документам чуть более сложная. Так как ни очередь сообщений, ни рабочие процессы чтения лент не гарантируют порядок, задача на создание документа может появиться в обработчике раньше, чем был создан счет. В таком случае документ будет создан без привязки к конкретному счёту. Это означает, что документ не принадлежит никакому сервисному центру, однако любой может его отправить, найдя его с помощью поиска. После появления счёта, документ может быть назначен на конкретный сервисный центр.

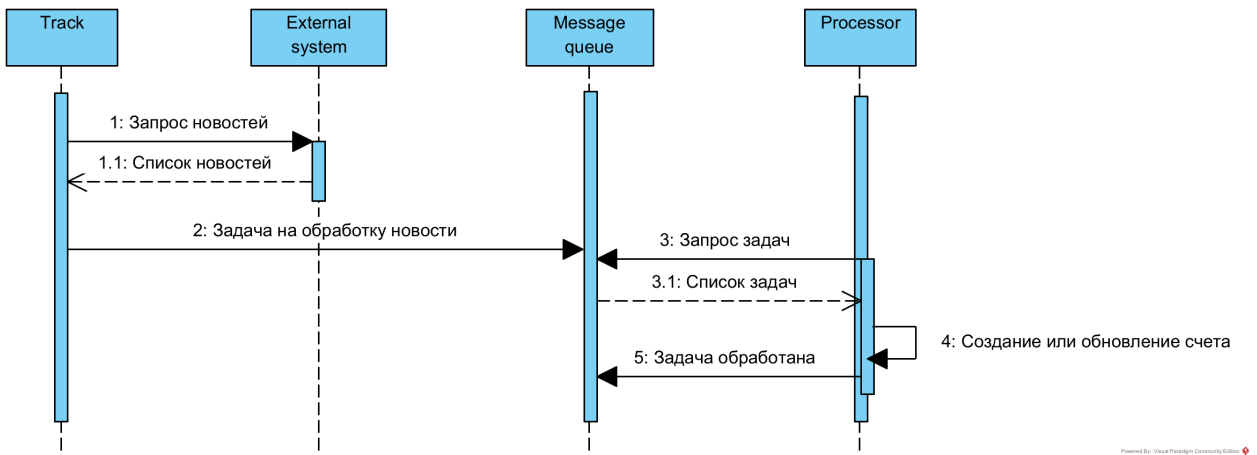


Рис. 3: Диаграмма последовательности обработки новостей по счетам

Кроме того, для некоторых документов, внешняя система отдает данные в API со ссылками на другие документы. В таком случае, чтобы получить всю необходимую информацию, требуется сделать дополнительный запрос документа по идентификатору в систему продаж. Общий случай отслеживания ленты документов изображен на рисунке 4.

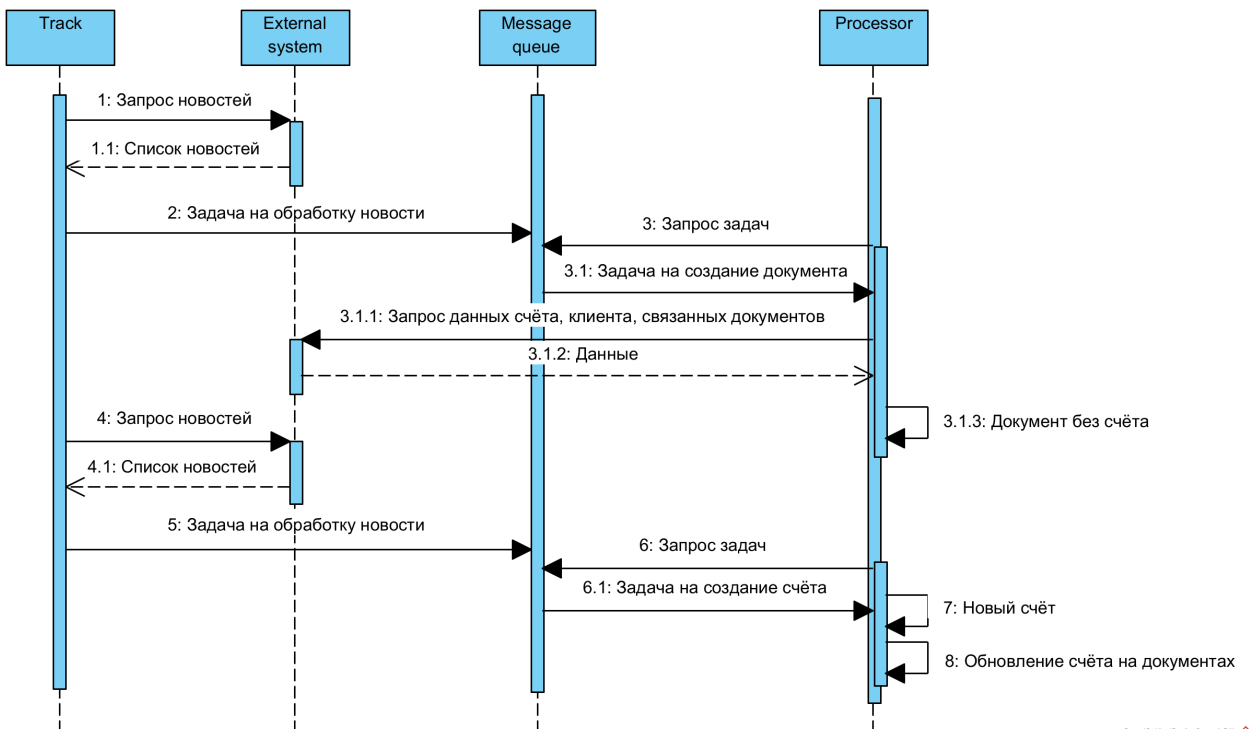


Рис. 4: Диаграмма последовательности обработки новостей по документам

В соответствии с двумя частями отслеживания лент модуль инте-

грации состоит из двух основных частей: демона с процессами чтения ленты и демона с процессами обработки задач из очереди сообщений. Кроме этого добавлен пакет с клиентами API системы продаж. Общая архитектура изображена на рисунке 5. На диаграмме синим изображены разработанные элементы, а серым уже существовавшие.

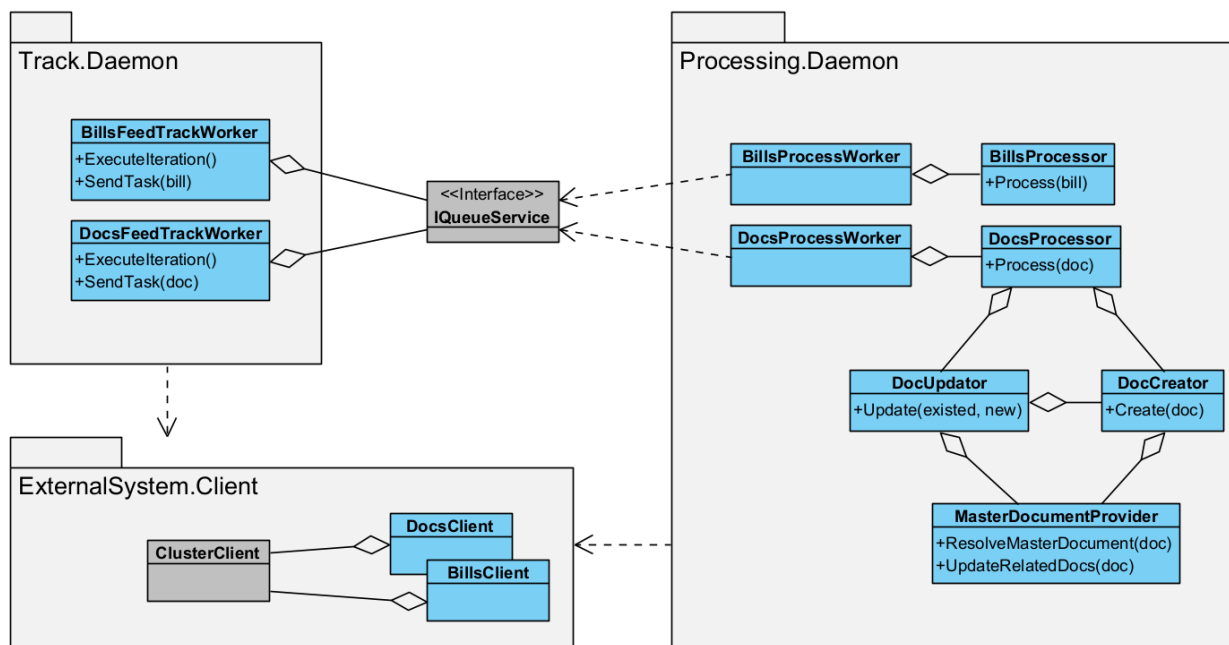


Рис. 5: Диаграмма классов модуля интеграции

Здесь пакет **Track** содержит рабочие процессы чтения ленты, которые запускаются с заданной периодичностью. С использованием реализации интерфейса **IQueueService** они отправляют задачи на обработку новостей. Их получают обработчики в модуле **Processing**, десериализуют и передают в **BillNewsProcessor** и **DocumentsNewsProcessor**, которые с использованием **DocUpdater** и **DocCreator** выполняют логику создания и обновления сущностей в сервисе. Внизу представлен модуль **Client**, содержащий клиенты различных API внешней системы.

### 3.1.1. Клиенты для внешней системы

Межсервисное взаимодействие в сети разделяют на синхронное и асинхронное [4]. В первом случае клиент выполняет запрос и дожидает-



ется ответа от другого приложения. При этом увеличивается связность сервисом между собой. Примерами протоколов такого взаимодействия являются HTTP и gRPC.

В случае асинхронного взаимодействия, вызывающая сторона публикует свой запрос и не дожидается ответа на него. Затем одно из отвечающих приложений обрабатывает запрос. В этом способе связность сервисов меньше, потому что клиенты не зависят от обработчиков. Однако появляется дополнительная инфраструктура в виде очереди сообщений, возникают сложности с гарантией доставки запроса и дублированием запросов.

В данной работе использовался синхронный метод взаимодействия с внешней системой с помощью протокола HTTP, потому что именно такое API уже было реализовано. Поскольку стороннее приложение представлено несколькими одновременно запущенными процессами — репликами, возникает задача определения IP-адреса реплик и балансировки между ними.

Для решения этих проблем клиенты, основываются на существующем модуле **Cluster Client**<sup>3</sup> из проекта Восток [16]. Он позволяет производить отправку HTTP-запросов на кластеры реплик сервисов. В нём реализованы различные стратегии отправки запросов, стратегии повторной отправки и балансировка между репликами.

Вообще есть несколько вариантов отправки запросов во внешние сервисы. Самый простой — делать запрос на внешне доступный DNS адрес. В таком подходе появляется зависимость от загрузки внешнего балансировщика, что в некоторых случаях, например, в случае DDoS-атаки, может приводить к большому времени выполнения запроса.

Другой способ — отправлять запрос на внутренний балансировщик, который самостоятельно выполняет запрос на одну или несколько реплик внешнего приложения [10]. В таком случае вместе с самим запросом передаётся ещё и название сервиса [7]. Внешняя команда управляет конфигурацией выполнения запросов, которая, в таком случае,

---

<sup>3</sup>ClusterClient.Core GitHub – <https://github.com/vostok/clusterclient.core> (дата обращения 02.05.2023)

является общей для всех клиентов. То есть у нас нет возможности переопределить стратегии отправки, время таймаутов и тому подобные настройки. С одной стороны это ограничивает возможности, с другой — централизует управление нагрузкой на внешний сервис. В нашем случае не нужны возможности тонкой настройки правил выполнения запросов, поэтому был выбран именно этот способ.

Третий вариант предполагает балансировку на стороне клиента. То есть перед отправкой запроса, с определенной периодичностью, клиентское приложение получает топологию внешней системы и проводит балансировку самостоятельно. В таком случае появляется возможность управлять всеми настройками отправки запросов. Однако балансировка в целом ухудшается, потому что, в отличие от второго способа, доступна только ограниченная статистика использования. Такой способ используется для выполнения запросов к самому внутреннему балансировщику.

## 3.2. Доступ к действиям

Основным изменением в процессе оборота документов по продажам является общий доступ сервисных центров к выпущенным договорам. Раньше между созданием заявления и его подписанием проходили минуты. Поэтому документы всегда отправляли те сервисные центры, которые их и создавали. В случае с покупкой продуктов это не так. Договор может быть выдан на ознакомление клиенту в одном сервисном центре, а подписание произойти в другом. Любой сервисный центр должен быть готов завершить продажу и собрать необходимые документы. Поэтому в ходе разработки было необходимо пересмотреть механизм управления доступа к действиям в системе.

Для этого использовался шаблон проектирования «Спецификация». Был добавлен интерфейс **IAccessSpecification** с одним методом, проверяющим, подходит ли принципал и документ под спецификацию. С помощью логических реализаций **AndSpecification**, **NotSpecification** и т.п., реализовываются конкретные спецификации для разных групп

пользователей. Например, **SenderServiceCenterSpecification** соответствует сотруднику сервисного центра, который отправил документ в головной офис. Диаграмма реализованного механизма приведена на рисунке 6.

С помощью спецификации задаются доступы к конкретным действиям в сервисе. Например, вернуть посылку обратно в сервисный центр может либо отправитель, либо сотрудник отдела обработки документов. Это соответствует следующей спецификации.

```
new OrSpecification(
    new SenderServiceCenterSpecification(),
    new OodEmployeeSpecification());
```

Таким образом достигается достаточно гибкая возможность управления доступом пользователей к действиям в сервисе.

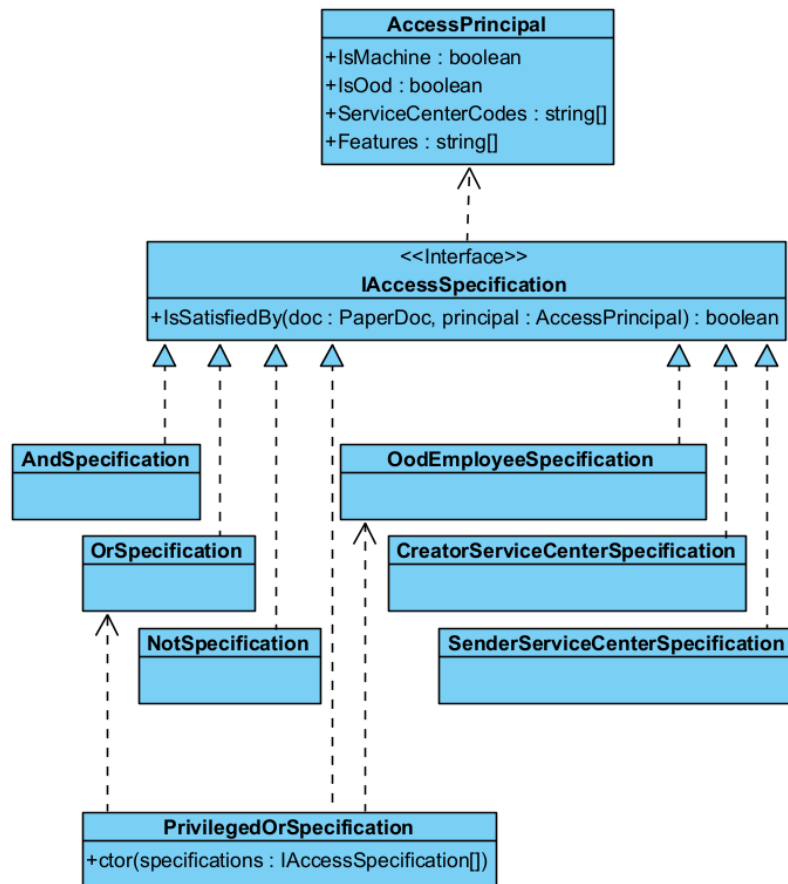


Рис. 6: Диаграмма классов модуля AccessSpecification

## 4. Апробация

В качестве апробации был произведён запуск новой функциональности в сервисе для ограниченного количества пользователей. Основная цель частичного релиза — это проверка работоспособности сервиса на лояльных сервисных центрах. Кроме этого важно собрать обратную связь сотрудников, сформулировать доработки, определить дополнительные сценарии документооборота. Основными критериями успешности пилотного запуска являются соответствие созданных документов в приложении их реальному наличию в сервисных центрах, выполнение различных процессов отправки и проверки документов несколькими сервисными центрами.

Для проведения апробации клиентская и серверная части приложения были доработаны возможностью разграничения доступа сотрудников к разным категориям документов. Интерфейс эксперта позволяет быстро подключать и отключать выбранные сервисные центры. В соответствии с этим в приложении скрываются разделы и ограничивается выполнение методов API. Всего доступ к новой функциональности получили 36 сервисных центров в пяти регионах и отдел обработки документов 8 марта 2023 года (далее все значения приводятся начиная с даты релиза и заканчивая 1 мая 2023 года не включительно).

Несмотря на доступ ограниченного числа пользователей к новой функциональности, модуль интеграции синхронизирует все документы, выпущенные во внешней системе. С одной стороны, это обусловлено единой лентой новостей для всех сервисных центров. С другой, это позволило оценить реальную нагрузку на приложение и рассмотреть значительно больше ситуаций. Новости по счетам обрабатывались с 1 января 2021 года. Такой большой интервал выбран потому что для создания документов необходимо иметь исторические данные по выставленным клиентам счетам. Документы синхронизировались только выпущенные после даты релиза. В момент первого запуска и чтения новостей из прошлого обрабатывалось суммарно около 50 новостей в секунду. Со временем это значение уменьшилось до 10 в пике в будние

дни. Независимость чтения лент и обработки новостей позволила сбалансировать нагрузку во время релиза, увеличив количество реплик с обработчиками. В итоге было создано 17.7 миллионов счетов и 50973 документа, 4608 из которых выпущены в пилотных сервисных центрах.

В ходе запуска модуля интеграции было обнаружено множество проблем в процессе обработки новостей. Некоторые были связаны с неучтенными редкими сценариями изменения документов. Другие были вызваны ошибками в коде, которые не были обнаружены на этапе тестирования. Какие-то указывали на неконсистентность данных во внешней системе. Большинство ошибок были своевременно исправлены так, что сейчас в сервисе остаются неучтенными всего около 15 документов.

За время апробации сервисными центрами было отправлено 10 посылок суммарно с 82 документами. Отдел обработки документов проверил 53 документа, из них четыре были отклонены и возвращены на исправление. В архиве сформировано четыре дела. Замечаний о критических ошибках, которые блокировали бы выполнения процессов документооборота, от сервисных центров и отдела обработки документов не поступало.

В итоге можно сказать, что запуск прошел достаточно успешно. Большинство сложностей вызвала логика обработки новостей документов. По ходу она была переосмыслена так, чтобы уменьшить зависимость от процессов во внешней системе. Тем не менее, выбранная архитектура с очередью сообщений отлично изолировала новости друг от друга, поэтому возникающие проблемы никак не влияли на работу сервиса в остальном.

## Заключение

В ходе работы разрабатывался сервис автоматизации бумажного документооборота. В приложение была добавлена функциональность, связанная с процессом оборота документов по продажам компании. Были выполнены следующие задачи.

- Рассмотрены существующие продукты для управления документооборотом, выявлены особенности существующего решения в компании.
- Спроектирован и реализован модуль интеграции с корпоративной системой управления продажами.
- Скорректирована бизнес-логика в части разграничения доступа к действиям и данным в сервисе.
- Приложение опубликовано для ограниченного количества пользователей. Проверена работоспособность сервиса, оценена его производительность, исправлены ошибки, начата работа над доработками.

Исходный код проекта закрыт.

## Список литературы

- [1] Business Process Model and Notation. — URL: <https://www.omg.org/spec/BPMN/2.0/PDF> (online; accessed: 12.12.2022).
- [2] ЕСМ система ЭЛАР Контекст | Платформа для организация делопроизводства и документооборота в организации. — URL: <https://elar-context.ru/> (online; accessed: 12.12.2022).
- [3] ELMA365 - Low-code система для автоматизации внутренних бизнес-процессов и CRM. — URL: <https://elma365.com/ru/> (online; accessed: 12.12.2022).
- [4] Interservice communication in microservices - Azure Architecture Center. — URL: <https://learn.microsoft.com/en-us/azure/architecture/microservices/design/interservice-communication> (online; accessed: 02.05.2023).
- [5] Patterns for distributed transactions within a microservices architecture | Red Hat Developer. — URL: <https://developers.redhat.com/blog/2018/10/01/patterns-for-distributed-transactions-within-a-microservices-arc> (online; accessed: 02.05.2023).
- [6] Replication — MongoDB Manual. — URL: <https://www.mongodb.com/docs/manual/replication> (online; accessed: 02.05.2023).
- [7] Service Discovery in a Microservices Architecture - NGINX. — URL: <https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/> (online; accessed: 02.05.2023).
- [8] Web Services Business Process Execution Language Version 2.0. — URL: <https://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf> (online; accessed: 02.05.2023).

- [9] В.М. Залкин. Разработка модуля автоматической проверки документов для системы учёта посылок и документов // XI конгресс молодых ученых. — Университет ИТМО, 2022. — Vol. 2. — P. 43–49.
- [10] Как Контур балансирует нагрузку в микросервисах / Хабр. — URL: <https://habr.com/ru/companies/skbkontur/articles/539410/> (online; accessed: 02.05.2023).
- [11] Новые правила работы с электронной подписью в 2022 году. Изменения в 63-ФЗ. — URL: [https://ca.kontur.ru/articles/20993-novyepravila\\_raboty\\_s\\_ep\\_2022](https://ca.kontur.ru/articles/20993-novyepravila_raboty_s_ep_2022) (online; accessed: 12.12.2022).
- [12] Паттерн Outbox: как не растерять сообщения в микросервисной архитектуре / Хабр. — URL: <https://habr.com/ru/companies/lamoda/articles/678932> (online; accessed: 02.05.2023).
- [13] Решения электронного документооборота Directum. — URL: <https://www.directum.ru/solutions?tags=14> (online; accessed: 02.05.2023).
- [14] Система управления документооборотом и бизнес-процессами Directum RX. — URL: <https://www.directum.ru/products/directum> (online; accessed: 02.05.2023).
- [15] Система электронного документооборота (СЭД) «ДЕЛО». — URL: [https://eos.ru/eos\\_products/eos\\_delo/](https://eos.ru/eos_products/eos_delo/) (online; accessed: 12.12.2022).
- [16] Технологии в Контуре / Восток. — URL: <https://tech.kontur.ru/vostok> (online; accessed: 02.05.2023).
- [17] Цифровая платформа для разработки бизнес-приложений – Comindware. — URL: <https://www.comindware.ru/platform/> (online; accessed: 12.12.2022).