

Определение архитектуры решения (SAD)

**CRM системы для телекоммуникационной компании и интеграция легаси компонентов**

**Заказчик ПАО «Ростелеком»**

**01.12.2022**

**Версия: 1**

**Проектная команда №6:**

* **Шуваев Сергей**
* **Малков Юрий**
* **Прищепенко Степан**
* **Милютин Иван**
* **Махмутов Ринат**

Оглавление

* [История изменений](https://github.com/mrr000/gb-coursework#%D0%B8%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F-%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9)
* [Цель проекта/задачи](https://github.com/mrr000/gb-coursework#%D1%86%D0%B5%D0%BB%D1%8C-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8)
* [Термины и сокращения](https://github.com/mrr000/gb-coursework#%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D1%8B-%D0%B8-%D1%81%D0%BE%D0%BA%D1%80%D0%B0%D1%89%D0%B5%D0%BD%D0%B8%D1%8F)
* [Бизнес-архитектура решения](https://github.com/mrr000/gb-coursework#%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F)
  + [Функциональная модель процессов](https://github.com/mrr000/gb-coursework#%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2)
* [Информационная архитектура решения](https://github.com/mrr000/gb-coursework#%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F-%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F)
  + [Потоки данных](https://github.com/mrr000/gb-coursework#%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B8-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-data-flow-diagram)
  + [Бизнес-данные](https://github.com/mrr000/gb-coursework#%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)
* [Архитектура приложения](https://github.com/mrr000/gb-coursework#%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F)
  + [Концептуальная модель и интеграция](https://github.com/mrr000/gb-coursework#%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B8-%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F)
  + [Перечень компонентов](https://github.com/mrr000/gb-coursework#%D0%BF%D0%B5%D1%80%D0%B5%D1%87%D0%B5%D0%BD%D1%8C-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2)
  + [Выбор СУБД](https://github.com/mrr000/gb-coursework#%D0%B2%D1%8B%D0%B1%D0%BE%D1%80-%D1%81%D1%83%D0%B1%D0%B4)
  + [Архитектура БД](https://github.com/mrr000/gb-coursework#%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D0%B1%D0%B4)
  + [Выбор языка программирования](https://github.com/mrr000/gb-coursework#%D0%B2%D1%8B%D0%B1%D0%BE%D1%80-%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)
  + [Модель компонентов приложений](https://github.com/mrr000/gb-coursework#%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)
  + [Модель развертывания приложений](https://github.com/mrr000/gb-coursework#%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D1%80%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)
  + [Open Api](https://github.com/mrr000/gb-coursework#open-api)
* [Требования и решения по безопасности](https://github.com/mrr000/gb-coursework#%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%B8-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BF%D0%BE-%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D0%B8)

### История изменений

| **Номер версии** | **Дата** | **Автор изменения** | **Суть изменения** |
| --- | --- | --- | --- |
| 1.0 | 01.12.2022 |  | Первоначальная версия документа |

### Цель проекта/задачи

**Цель:**

* Повышение качества взаимодействия компании (Ростелеком) с ее клиентами за счет расширения текущих функциональных возможностей CRM

**Задачи:**

* Разработать архитектуру, реализующую требования к решению
* Интегрировать переиспользуемые функциональные возможности уже реализованных компонент
* Реализовать отсутствующие функциональные возможности путем разработки новых, доработки или замещения уже существующих в наследованных компонентах (без доработки самих компонент) в создаваемом решении
* Спроектировать интеграцию решения с другими системами

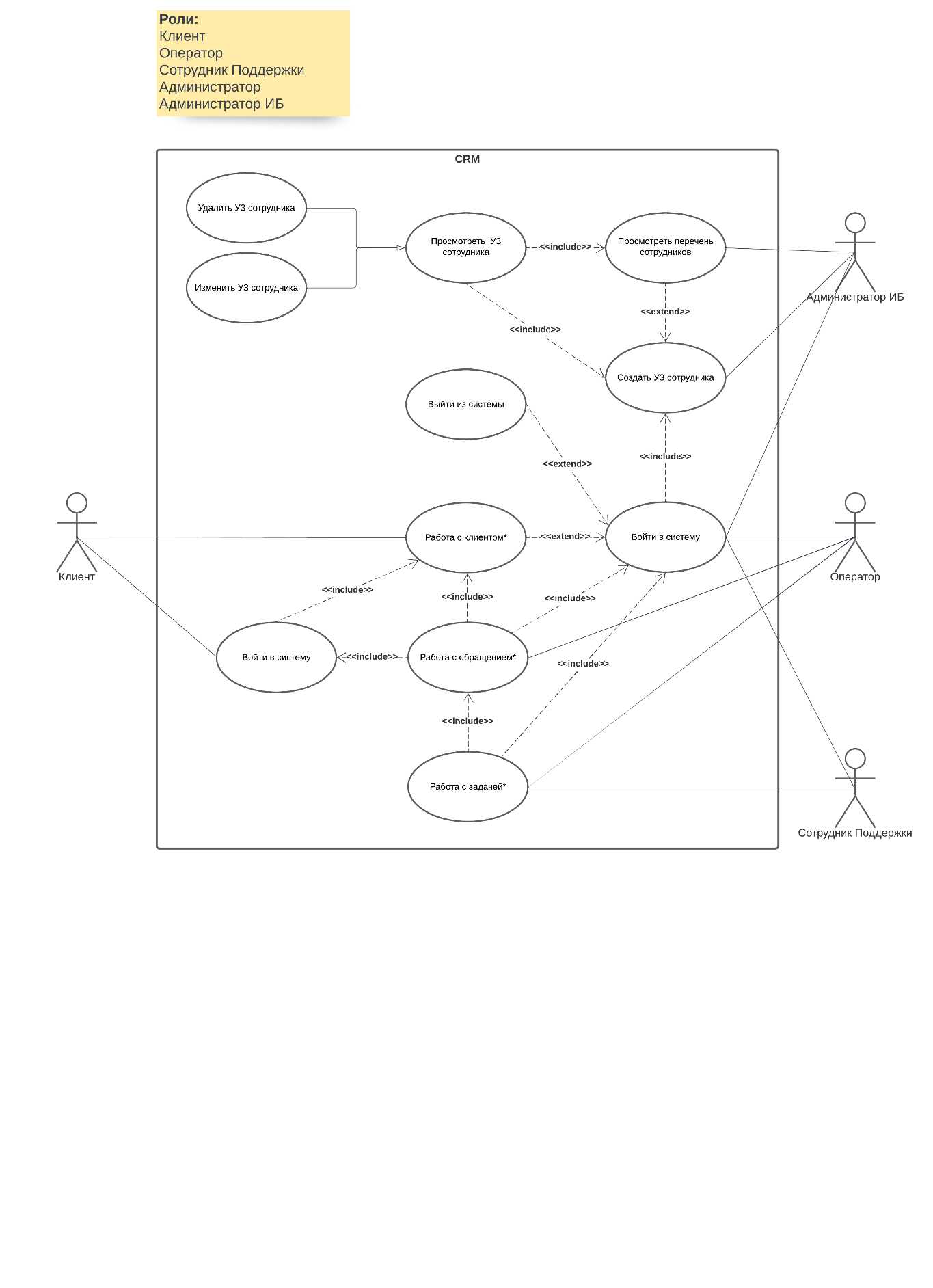
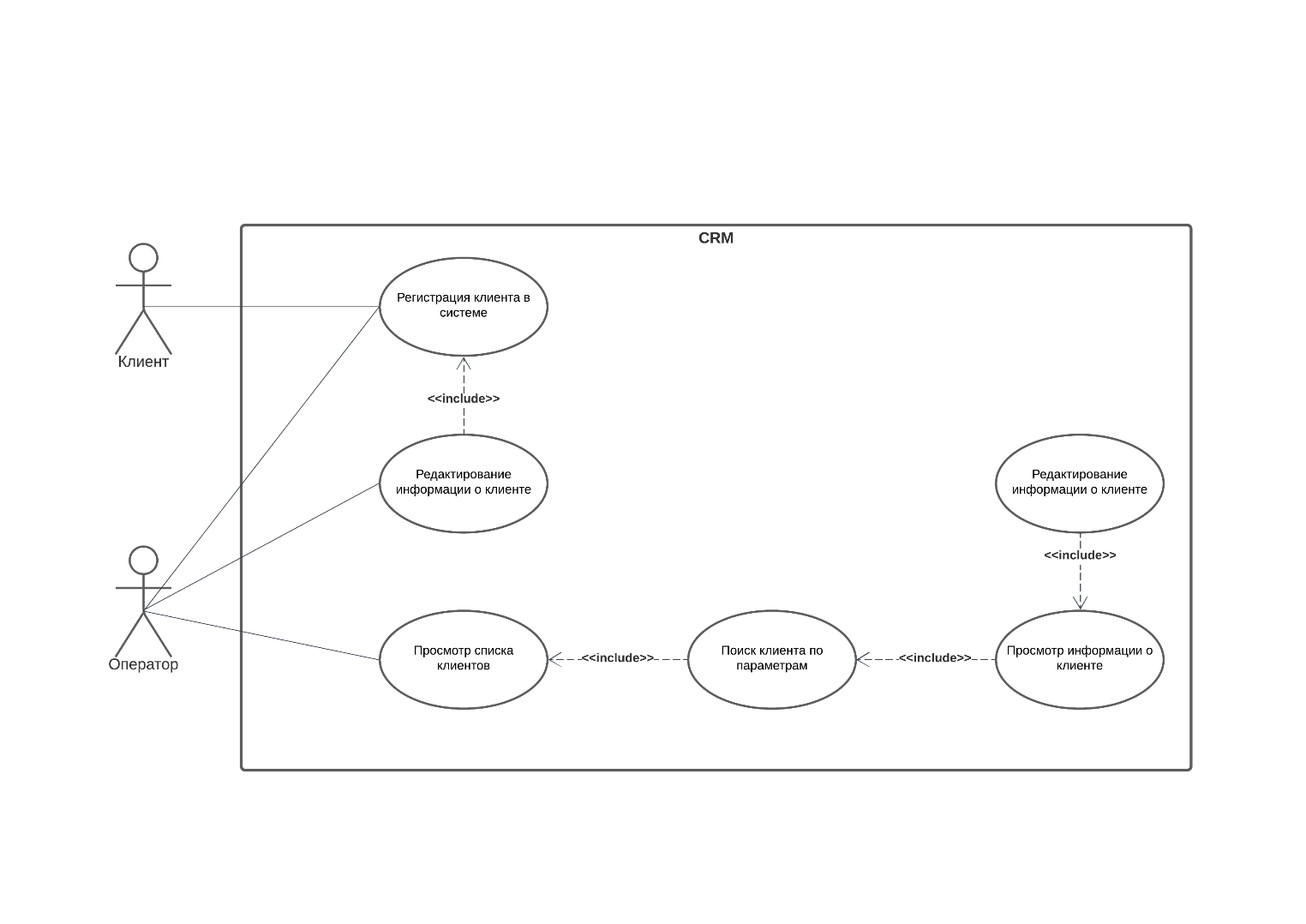
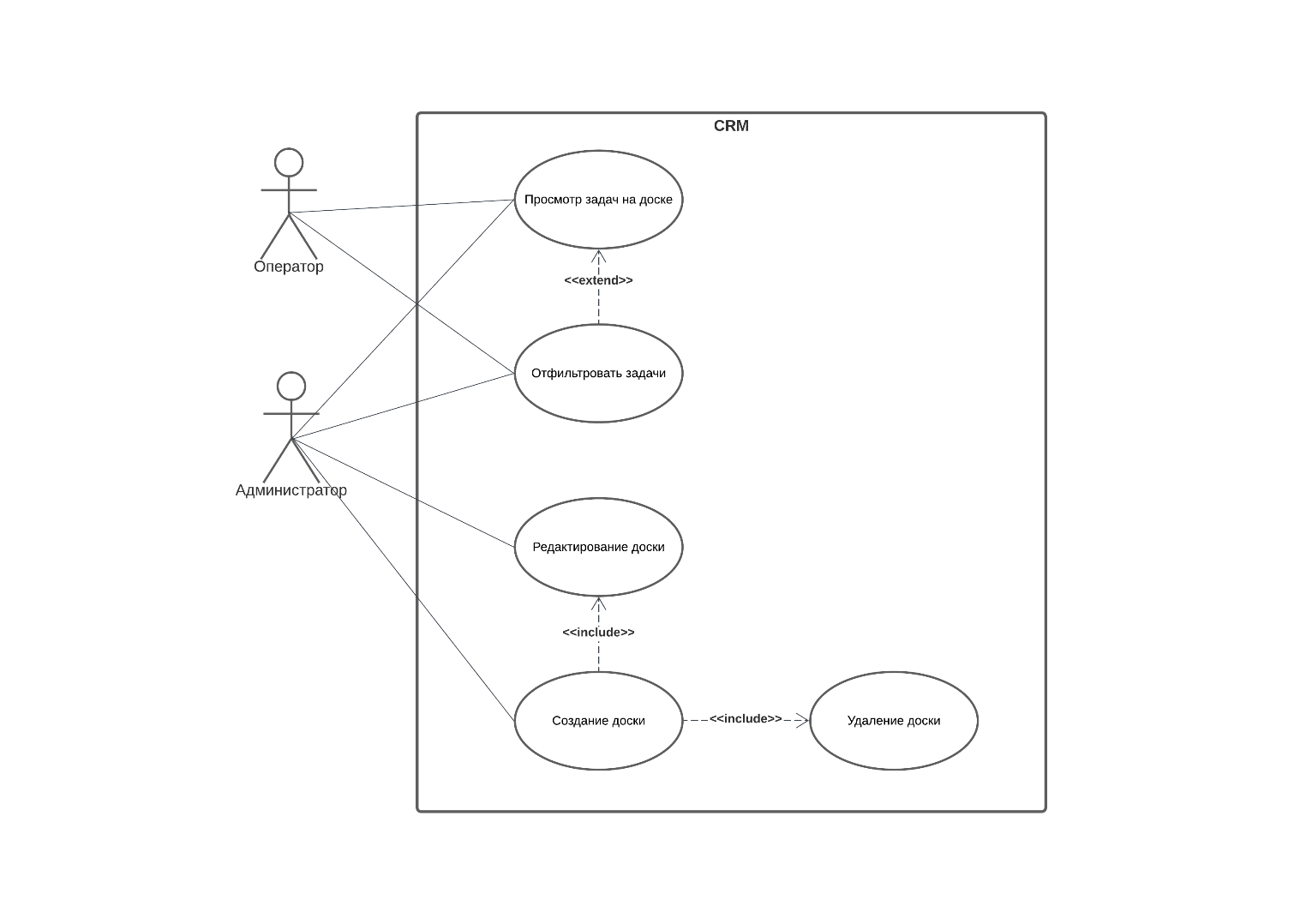
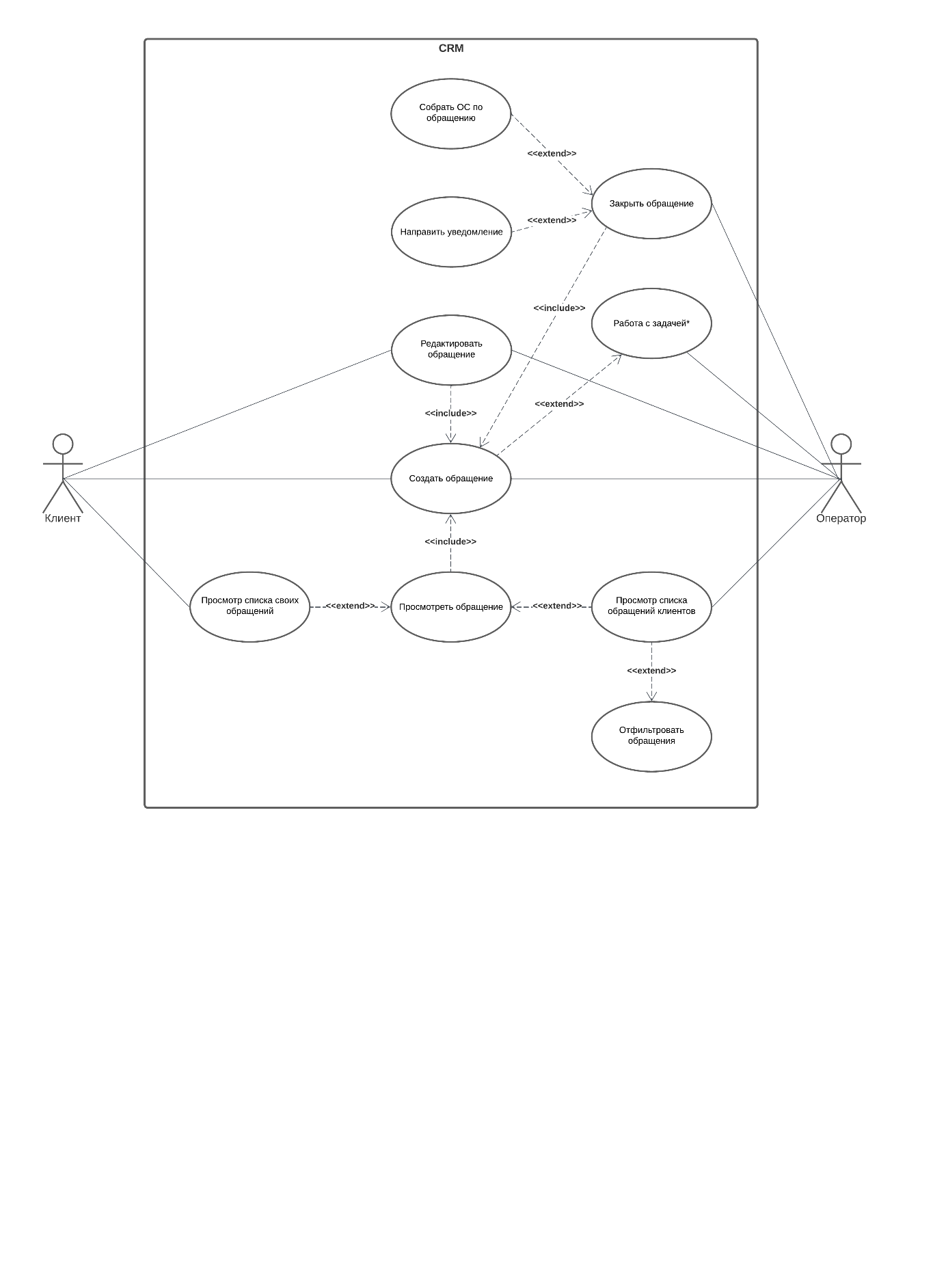
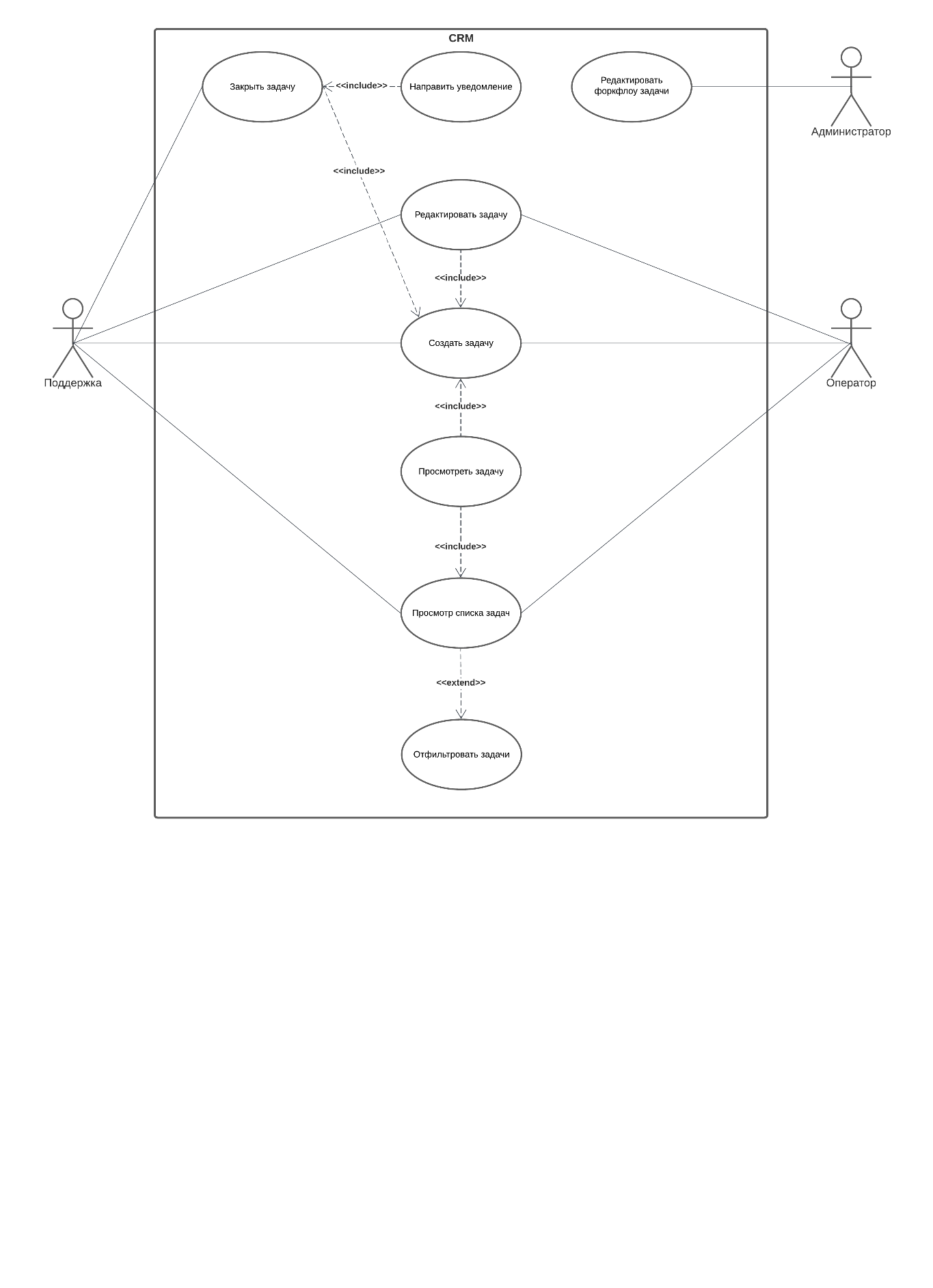
### Термины и сокращения

| **№** | **Наименование** | **Описание** | **Атрибуты** |
| --- | --- | --- | --- |
| 1 | CRM | ПО, обеспечивающее взаимодействие телеком компании и клиента посредством обращений |  |
| 2 | Компания | Телеком организация предоставляющая услуги |  |
| 3 | Пользователи | Профиль работника, обеспечивающий обработку входящих обращений клиентов | Уникальный идентификатор ФИО Отдел/подразделение Должность Корп. почта Корп. телефон Личный телефон Роль ID |
| 4 | Клиент | Физическое или юридическое лицо, обращающееся в телеком компанию за услугой | Уникальный идентификатор Юр. принадлежность ФИО Паспортные данные Адрес регистрации Адрес фактического проживания Номер телефона E-mail ИНН ОГРН Название организации Физический адре Почтовый адрес Факс |
| 5 | Договор | Документ, подтверждающий предоставление услуг клиенту | Уникальный идентификатор Клиент Номер договора Дата заключения договора Дата завершения договор Шаблон договора |
| 6 | Поставляемые услуги | Услуги предоставляемые клиенту в рамках действующих договоров | Клиент Договор Услуга Дата начала действия Дата окончания действия |
| 7 | Список услуг | Номенклатурный справочник, в котором зафиксирован весь спектр предоставляемых компанией услуг | Уникальный идентификатор Наименование услуги Платная/бесплатная Товарная/нетоварная Разовая/периодическая Срок предоставления Цена |
| 8 | Обращение | Зарегистрированная потребность клиента в получении обратной связи от компании по интересующему его вопросу | Уникальный идентификатор Номер обращения Клиент Статус обращения Дата создания Дата исполнения Дата закрытия Пользователь принявший обращение ФИО Телефон обратной связи Канал поступления Необходимость обратной связи Комментарии |
| 9 | Задача | Элемент бизнес-процесса, обеспечивающий цикл исполнения обращения клиента на не материальной основе | Уникальный идентификатор Номер задачи Номер обращения Статус задачи Исполнитель Дата создания Дата закрытия Тип задачи |
| 10 | Заказ | Элемент бизнес-процесса, обеспечивающий цикл исполнения обращения клиента на материальной основе | Уникальный идентификатор Номер задачи Номер обращения Дата заказа Дата поставки Клиент Договор Тип заказа |
| 11 | Состав заказа | Перечень предоставляемых услуг для конкретного заказа | Уникальный идентификатор заказа Уникальный идентификатор Точка поставки Услуга Цена |
| 12 | Роли | Номенклатурный справочник, в котором зафиксирован весь перечень пользовательских ролей для разграничения доступа | Уникальный идентификатор Наименование роли Описание доступных действий и возможностей |

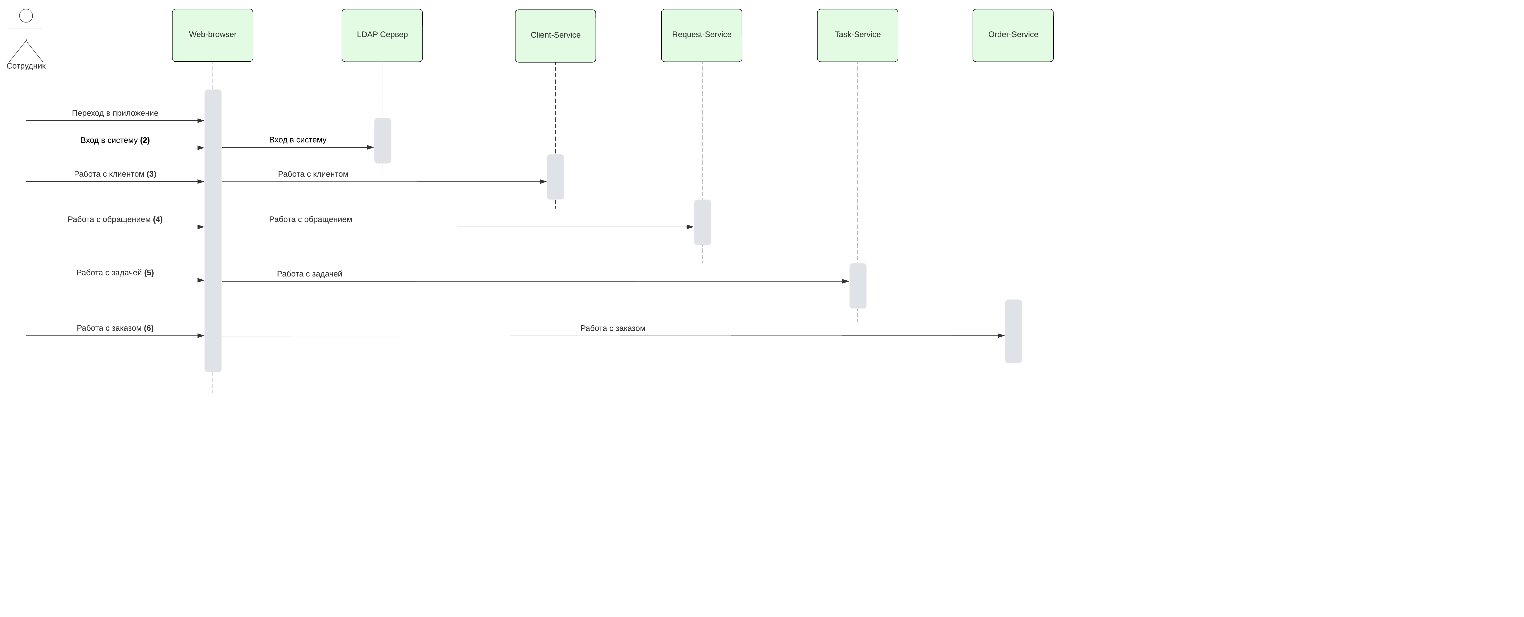
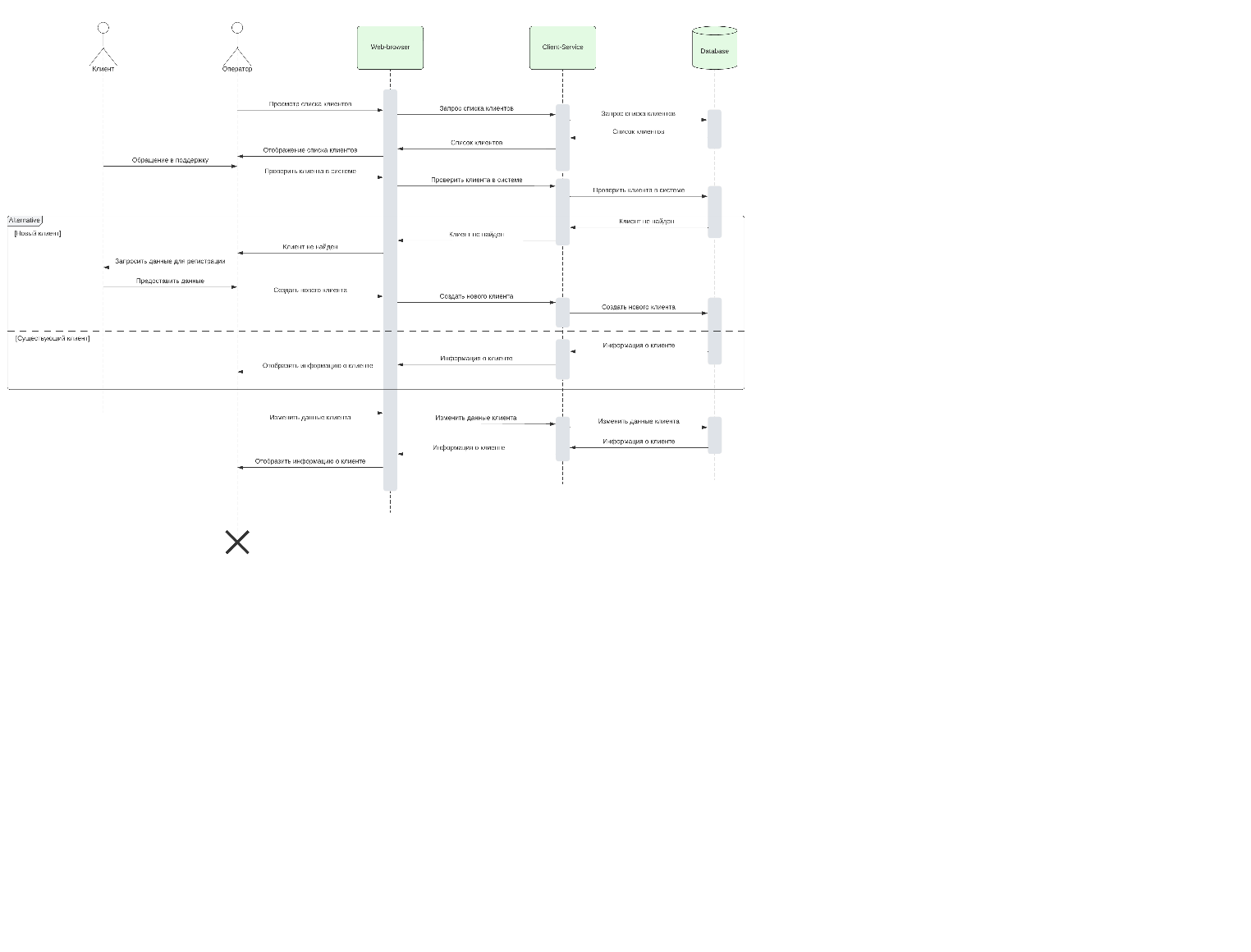
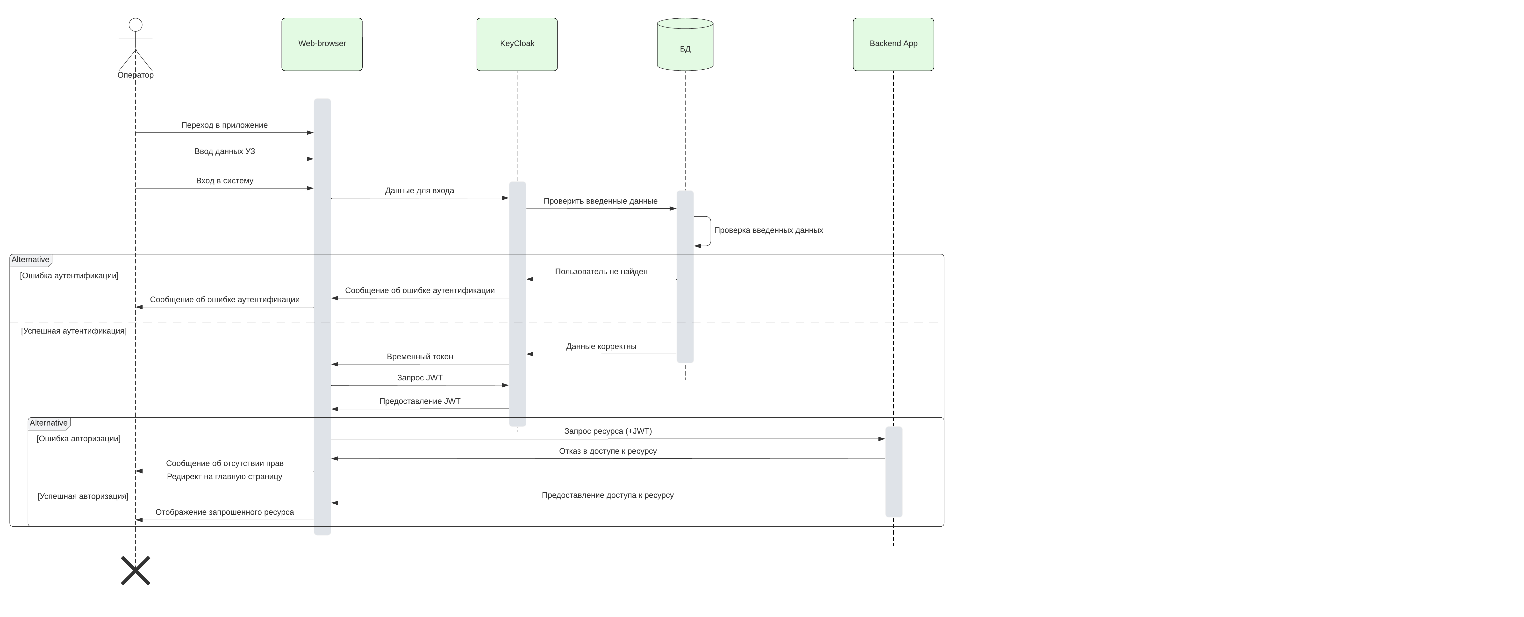
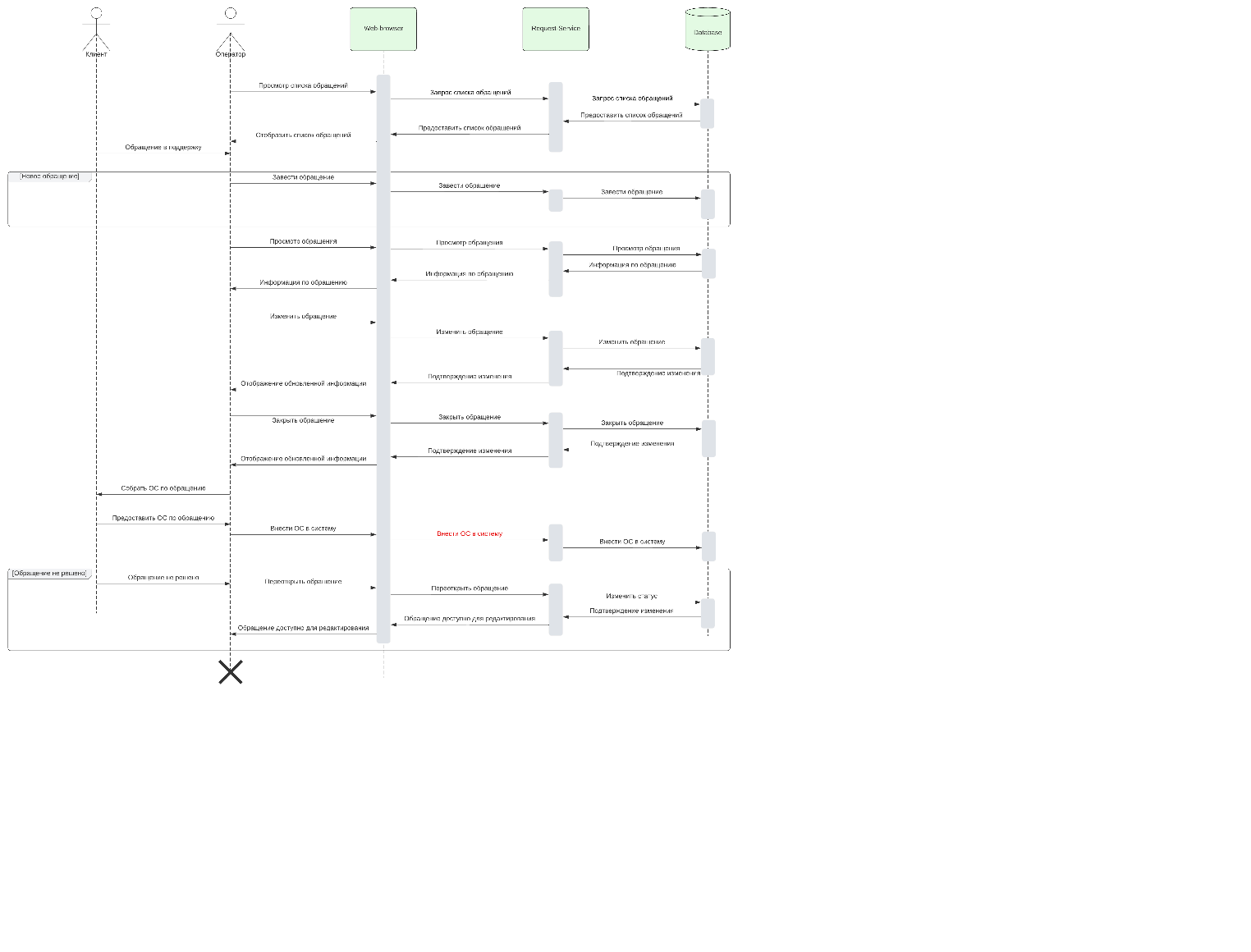
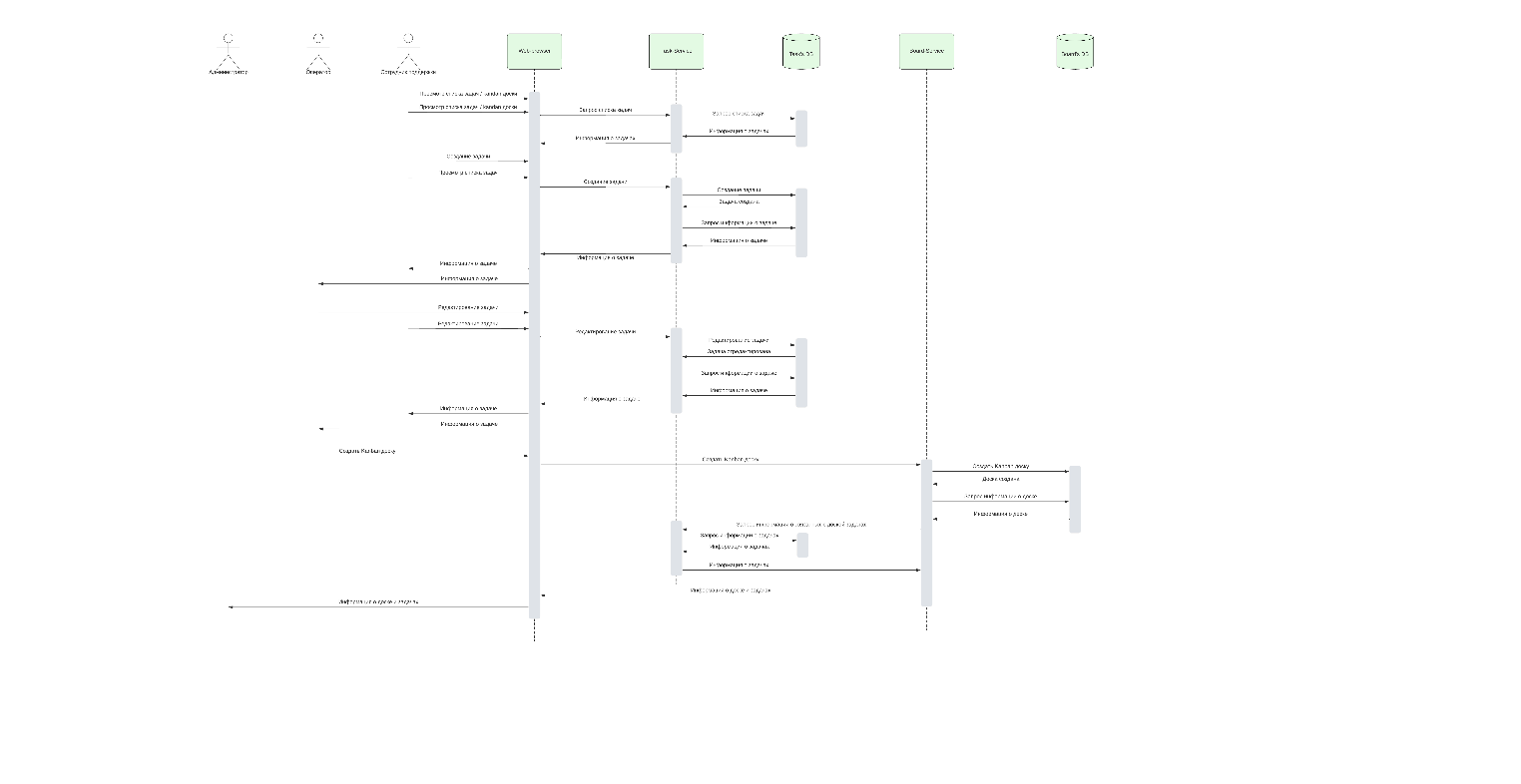
### Бизнес-архитектура решения

#### Функциональная модель процессов

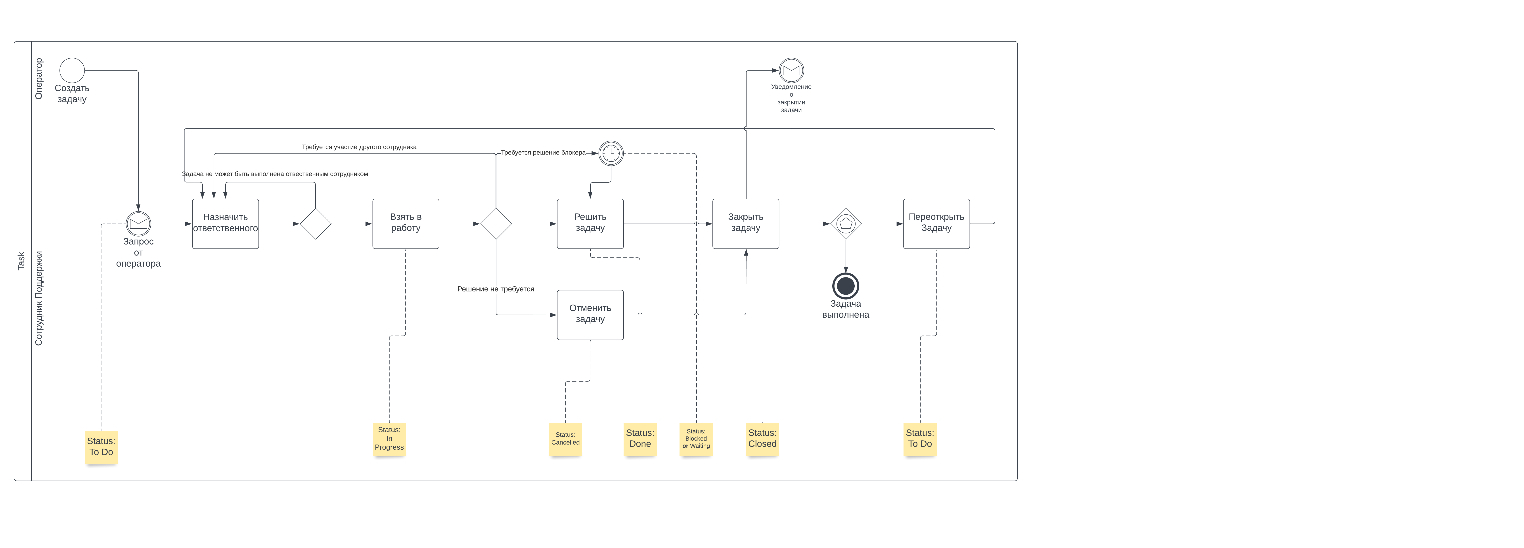
Use-case diagrams

[](https://github.com/mrr000/gb-coursework/blob/master/Use-case/Use_Overview.png) [](https://github.com/mrr000/gb-coursework/blob/master/Use-case/Use_Client.png) [](https://github.com/mrr000/gb-coursework/blob/master/Use-case/Use_Dashboard.png) [](https://github.com/mrr000/gb-coursework/blob/master/Use-case/Use_Request.png) [](https://github.com/mrr000/gb-coursework/blob/master/Use-case/Use_Task.png)

Sequence diagrams

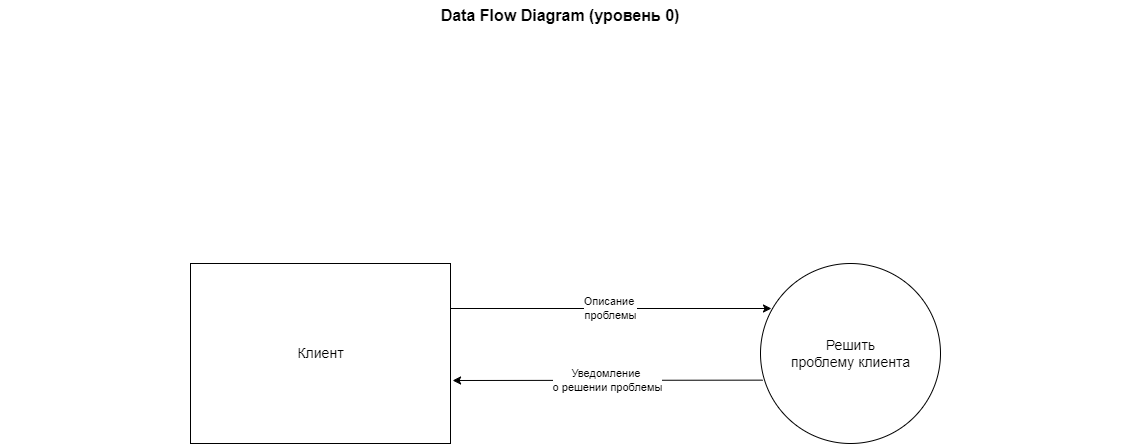
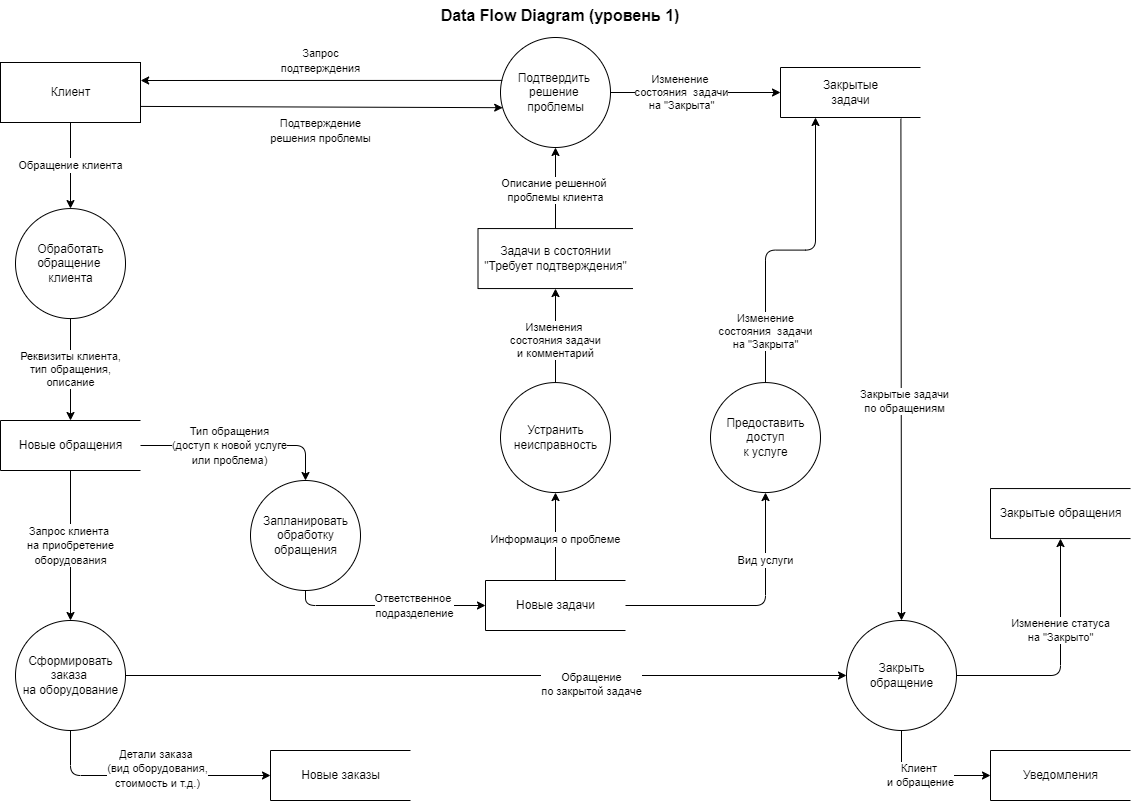
[](https://github.com/mrr000/gb-coursework/blob/master/Sequence/S_Overview.png) [](https://github.com/mrr000/gb-coursework/blob/master/Sequence/S_Client.png) [](https://github.com/mrr000/gb-coursework/blob/master/Sequence/S_Auth.png) [](https://github.com/mrr000/gb-coursework/blob/master/Sequence/S_Request.png) [](https://github.com/mrr000/gb-coursework/blob/master/Sequence/S_Task.png)

BPMN diagram

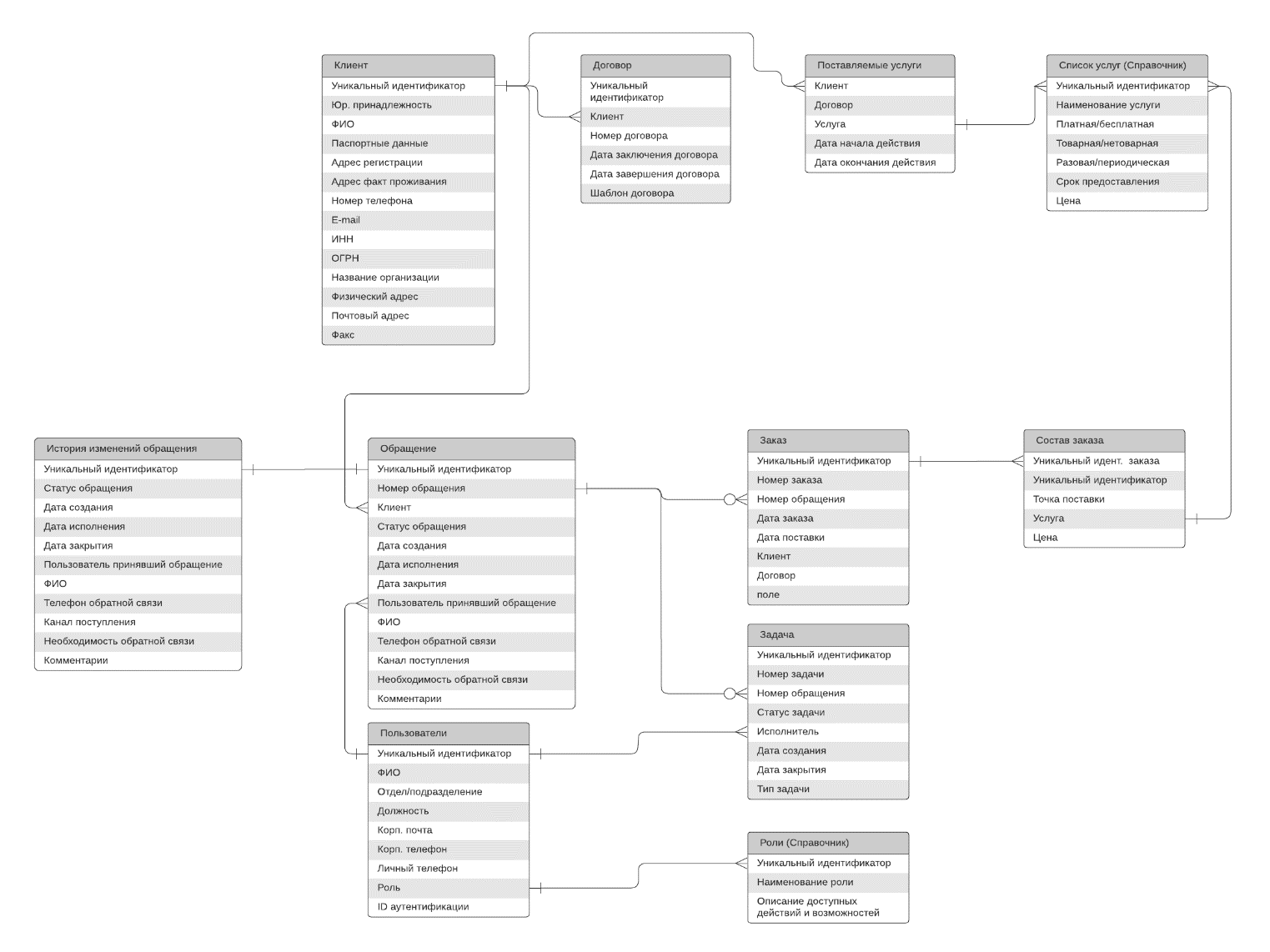
[](https://github.com/mrr000/gb-coursework/blob/master/BPMN/BPMN_Task.png)

### Информационная архитектура решения

#### Потоки данных (data flow diagram)

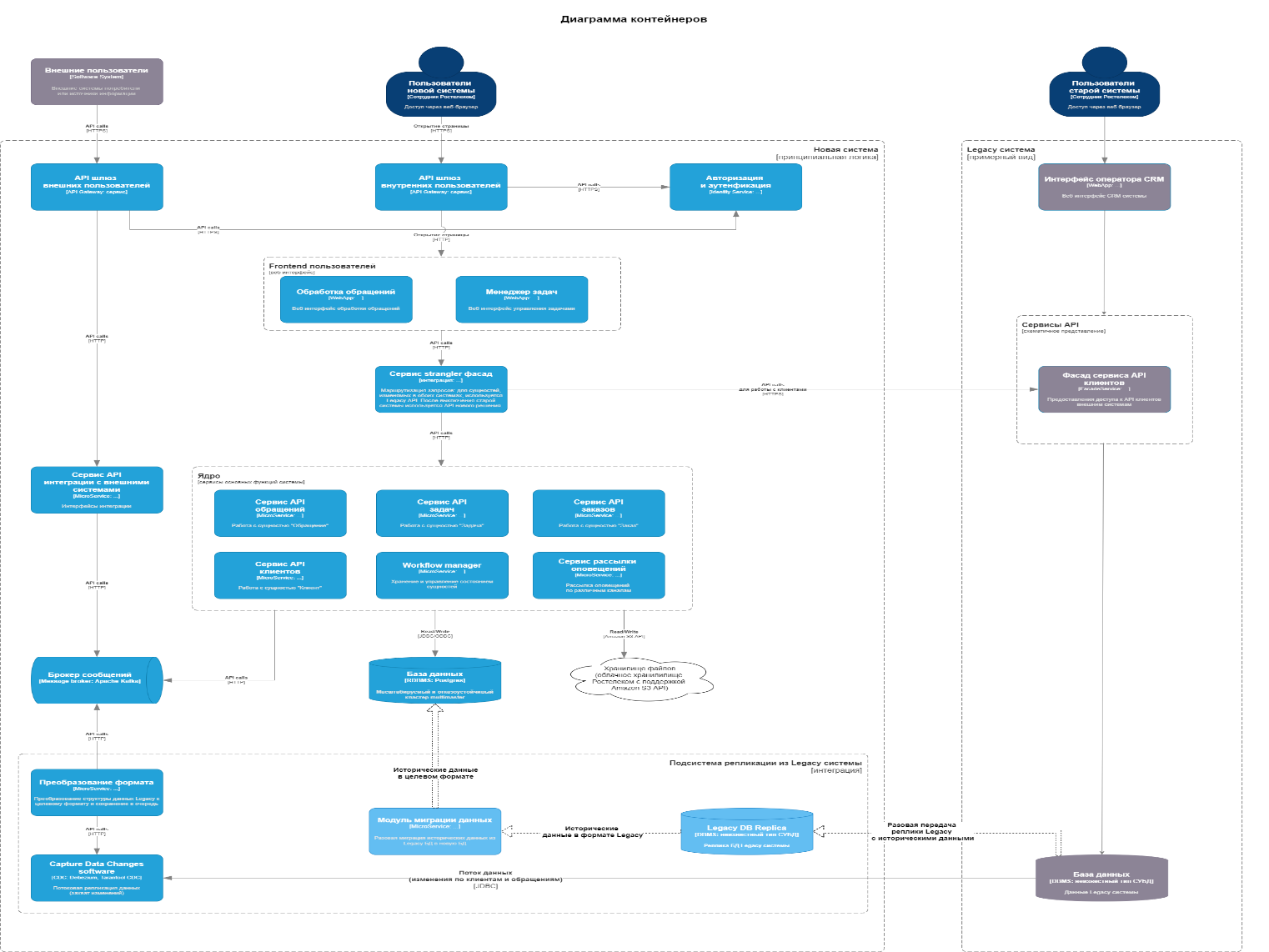
Уровень 0[](https://github.com/mrr000/gb-coursework/blob/3fa3828089952c9b50eab3b1765ad0131bea2621/DFD/DFD-level0.png)  
Уровень 1[](https://github.com/mrr000/gb-coursework/blob/3fa3828089952c9b50eab3b1765ad0131bea2621/DFD/DFD-level1.png)

#### Бизнес-данные

ER Diagram[](https://github.com/mrr000/gb-coursework/blob/7f9931ee7e4c1628a01efc47b7795f569c042c39/ERD/ER-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%20-%20%D0%A0%D0%BE%D1%81%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC.png)

### Архитектура приложения

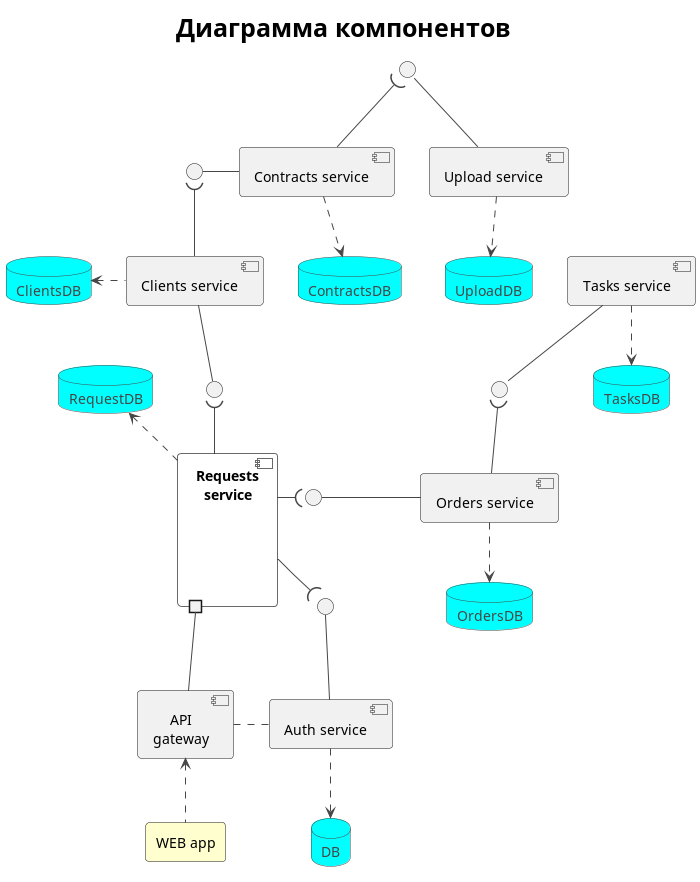
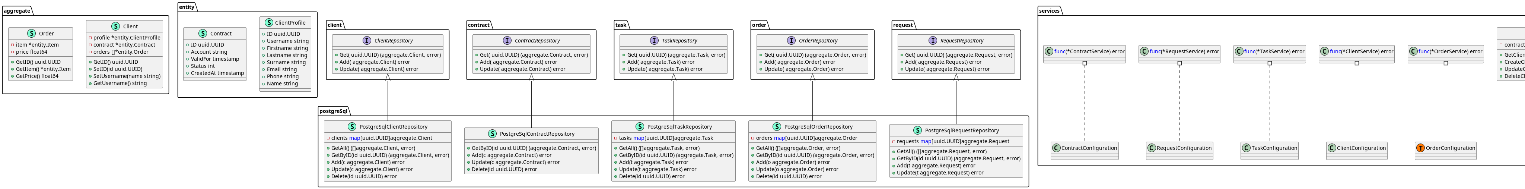
#### Концептуальная модель и интеграция

Показать[](https://github.com/mrr000/gb-coursework/blob/17964f2df2398e33a1259572e981f272d098c6c3/C4-diagram/container%20diagram.png)

#### Перечень компонентов

| **№** | **Название системы/модуля/компонента** | **Тип объекта** | **Платформа** | **Назначение** | **Зона размещения компонента** | **Статус изменения компонента** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Client-service | Сервис | RHEL/Docker/Go | Решение для управления клиентами компании | LAN | Разработка |
| 2 | Contracts-service | Сервис | RHEL/Docker/Go | Предоставление информации о договорах | LAN | Разработка |
| 3 | Dashboard-service | Сервис | RHEL/Docker/Go | Управление Kanban досками для персонала | LAN | Разработка |
| 4 | Upload-service | Сервис | RHEL/Docker/Go | Загрузка и преобразование данных из легаси системы | LAN | Разработка |
| 5 | Tasks-service | Сервис | RHEL/Docker/Go | Решение для управления задачами внутри компании | LAN | Разработка |
| 6 | Requests-service | Сервис | RHEL/Docker/Go | Решение для регистрации и ведения обращений клиентов | LAN | Разработка |
| 7 | Orders-service | Сервис | RHEL/Docker/Go | Решение для управления заказами клиентов | LAN | Разработка |
| 8 | KeyCloack | Система | Java | Решение для управления пользователями и доступом | LAN | Применение |
| 9 | Kafka | Система | Java, Scala | распределённый программный брокер сообщений | LAN | Применение |
| 10 | API Gateway | Сервис | RHEL/Docker/Go | Решения для связывания програмных компонентов | LAN | Разработка |
| 11 | Система логирования | Система | RHEL/Docker/ELK | Движок для работы с логами приложения | LAN | Применение |

#### Модель компонентов приложений

Диаграмма компонентов[](https://github.com/mrr000/gb-coursework/blob/master/components/components2.png)Диаграмма классов[](https://github.com/mrr000/gb-coursework/blob/master/classes/full.png)

#### Выбор СУБД

Для проектируемого решения предлагается использовать следующие СУБД:

* Postgres Pro Enterprise – для хранения и обработки транзакционных данных.
* Виртуальное хранилище Ростелеком – для хранения файлов.

Выбор виртуального хранилища Ростелеком обусловлен использованием уже существующей инфраструктуру заказчика и поддержкой распространенного протокола Amazon S3 API. СУБД Postgres Pro была выбрана в результате оценки критериев сравнения между наиболее распространенными реляционными СУБД (данный тип обеспечивает наиболее эффективную обработку транзакционных данных).

С учетом специфики заказчика, наиболее целесообразным является выбор СУБД Postgres, обладающей следующими преимуществами:

* отсутствие санкционных рисков использования;
* наличие вендоров осуществляющих поддержку и развитие на территории РФ;
* наличие open source версии для исключения vendor lock.

Далее среди отечественных решений на основе СУБД Postgres был выбран продукт, имеющий реализацию высокодоступного (high availability) кластера «из коробки»: СУБД Postgres Pro Enterprise. Основным преимуществом реализации кластера СУБД Postgres Pro Enterprise является:

* наличие конфигурации multimaster, позволяющей в отличии от других продуктов выполнять масштабирование операций не только на чтение, но и на запись;
* использование разработанного расширения Postgres (без использования стороннего программного обеспечения) для создания и управления кластером.

Показать критерии

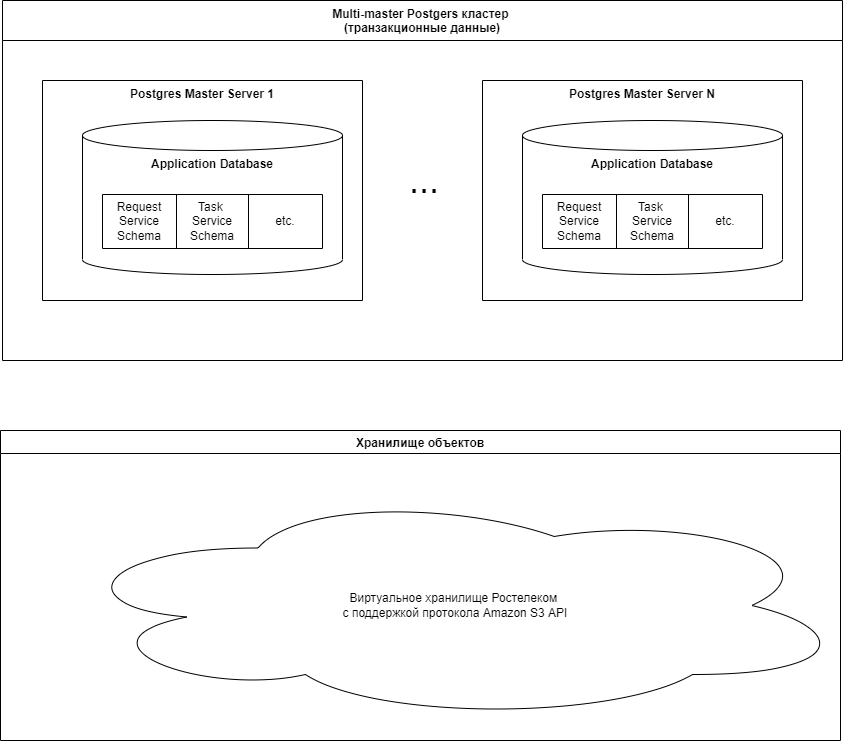
| **Группа критериев** | **№** | **Критерий** | **Оценка от 1 до 10 (10 высшая оценка)** |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Oracle | My SQL | Microsoft SQL Server | Postgres |
| Оценка поставщика и его опыт | 1 | Наличие опыта применения в крупных компания мирового масштаба | 10 | 10 | 10 | 10 |
|  | 2 | Риски по ограничению поддержки или приобретению лицензий со стороны поставщика по политическим причинам (критичный критерий) | 1 | 5 (наличие open source | 1 | 10 |
|  | 3 | Наличие вендора на территории РФ осуществляющего поддержку и развитие продукта | 1 | 1 | 1 | 10 |
|  |  | Итого: | 12 | 16 | 12 | 30 |
| Функциональность | 4 | Соответствие функциональным требованиям в целом (сумма) | 40 | 40 | 40 | 33 |
|  | 4.1 | Поддержка OLTP типа обработки данных (ACID транзакции) | 10 | 10 | 10 | 10 |
|  | 4.2 | Partitioning (разделение строк по физическим разделам) | 10 | 10 | 10 | 10 |
|  | 4.3 | Sharding (разделение строк по нодам) | 10 | 10 | 10 | 5 (ручное с расширением FDW) |
|  | 4.4 | Возможности масштабирования решения | 10 (Oracle RAC) | 10 (3rd party в бесплатной и Built-in в платной) | 10 (WSFC и технология AlwaysOn) | 8 (3rd party в бесплатной версии без масштабирования на запись и Built-in в платной) |
|  | 5 | Соответствие нефункциональным требованиям в целом (сумма) | 14 | 30 | 13 | 23 |
|  | 5.1 | Наличие Open Source версии (отсутствие зависимости от услуг конкретного поставщика) | 1 | 10 | 1 | 10 |
|  | 5.2 | Место в рейтинге RDBMS СУБД (по данным <https://db-engines.com/en/ranking_trend/relational+dbms>) | 10 | 10 | 7 | 5 |
|  | 5.3 | Оценка количества специалистов использующих СУБД (по данным <https://insights.stackoverflow.com/survey/2021#key-territories-country>) | 3 | 10 | 5 | 8 |
|  |  | ИТОГО: | 66 | 86 | 65 | 86 |

Таблица отечественных продуктов на основе СУБД Postgres

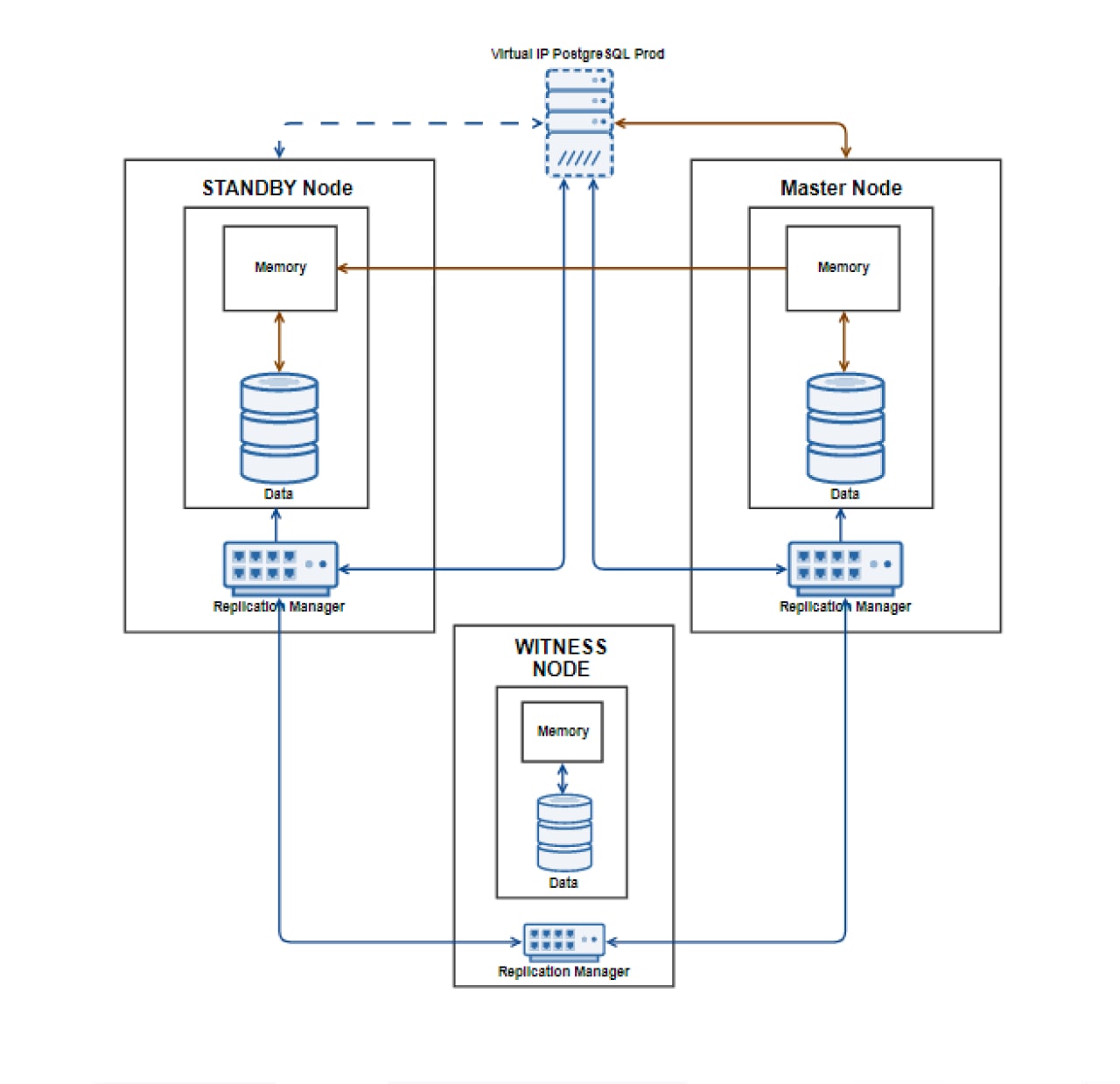
| **№п.п.** | **Наименование продукта** | **Вендор** | **Наличие кластера масштабируемого на запись и чтение** |
| --- | --- | --- | --- |
| 1 | Postgres Pro Enterprise | Postgres Pro | Да |
| 2 | СУБД Tantor | "Лаборатории Тантор" | Нет |
| 3 | СУБД Jatoba | Газинформсервис | Нет |
| 4 | Platform V Pangolin | Сбербанк-Технологии | Нет |
| 5 | СУБД “Квант- Гибрид” | О «Концерн ГРАНИТ» | Нет |
| 6 | Arenadata Postgres | Arenadata | Нет |
| 7 | СУБД "ЛИРА-Р" | НППКТ | Нет |

#### Архитектура БД

Масштабирование

[](https://github.com/mrr000/gb-coursework/blob/master/DB-architecture/DB-architecture.png)

Катастрофоустойчивость

[](https://github.com/mrr000/gb-coursework/blob/master/DB-architecture/DB_DisasterRecovery.jpg)

#### Выбор языка программирования

##### GoLang

| **Плюсы** | **Минусы** |
| --- | --- |
| Статическая типизация | Обработка ошибок требует написания док кода |
| Идеально подходит при написании микросервисов | Трудночитаемая документация |
| Компилируемый на любую платформу | Сложен для понимания при переходе с ООП |
| Конкурентность выраженная в легковесных горутинах | Нет стандартизированных рекомендаций написания кода |
| Есть дженерики |  |
| Низкий порог вхождения |  |
| Отсутствие исключений |  |
| Большое сообщество |  |
| Нет полной реализации ООП |  |
| Можно легко работать с любым протоколом |  |
| Можно спрогнозировать объем потреб ресурсов |  |
| Надежный |  |
| Возможность подключать сторонние библиотеки С/С++ |  |

##### PHP

| **Плюсы** | **Минусы** |
| --- | --- |
| Полноценное ООП | Динамическая типизация |
| Параллелизм | Не компилируемый |
| Воркеры | Низкая скорость работы |
| Обработка ошибок | Завершает процесс при каждом вызове |
| Низкий порог входа | Требует сторонний веб сервер для работы |
| Огромное сообщество | Динамическая типизация |
| Огромное количество готовых решений | Только для CLI и веб |
| Дешевая стоимость часа работы программиста | Слабо контролируемое потребление ресурсов |
| Хороший программист - редкость | Вынужденность применения фреймворков уменьшает безопасность |
| Отличная документация |  |
| Рекомендации написания кода PSR |  |

В современных проектах применяется комбинация этих двух языков - [RoadRunner](https://roadrunner.dev/), когда препроцессор PHP находится в обертке Go, таким образом процесс PHP является "не убиваемым". Но такие решения чаще распространяются на устоявшиеся проекты, которые требуют увеличения производительности. И в затраты на них учитывается работа программистов обоих стеков. Само решение "сырое" и что также рискованно к применению.

Язык Go отлично подходит для реализации небольших, стабильно работающих приложений, может быть скомпилирован на любую платформу. Писать на нем "идиоматически верно" при переходе с ООП языков трудно, что занимает время и желательно наличие грамотного (дорогого) наставника. Не рекомендуется для написания больших монолитных проектов. Не требует стороних сервисов для работы, что также облегчает разворачивание приложения в кластере k8s. Прекрасно работает с protobuf через gRpc, что облегчает взаимодействие между сервисами из-за вынужденного соблюдения интерфейсов. Что дает гарантию целостности данных и их правильного восприятия. Минимальная вероятность появления панических ошибок и надежность исполнения кода - основная причина выбора этого языка. Наличие в команде опытного специалиста - гарантия быстрого включения в проект любого специалиста при переходе с любого языка программирования, за кротчайшее время.

В то же время низкий уровень безопасности из-за необходимости использования сторонних фреймворков в PHP, возможность программирования в разных стилях, приводящее к увеличению тех долга, сложное разворачивание в кластере. Слабо контролируемое потребление ресурсов в комбинации с правилами ограничения потребления ресурсов кластера, может привести к сбоям и потерям пакетов. В результате низкие затраты на специалистов на этапе старта проекта приведет к очень дорогой поддержке в будущем, потери данных, неявным ошибкам, и большим затратам на трафик и железо.

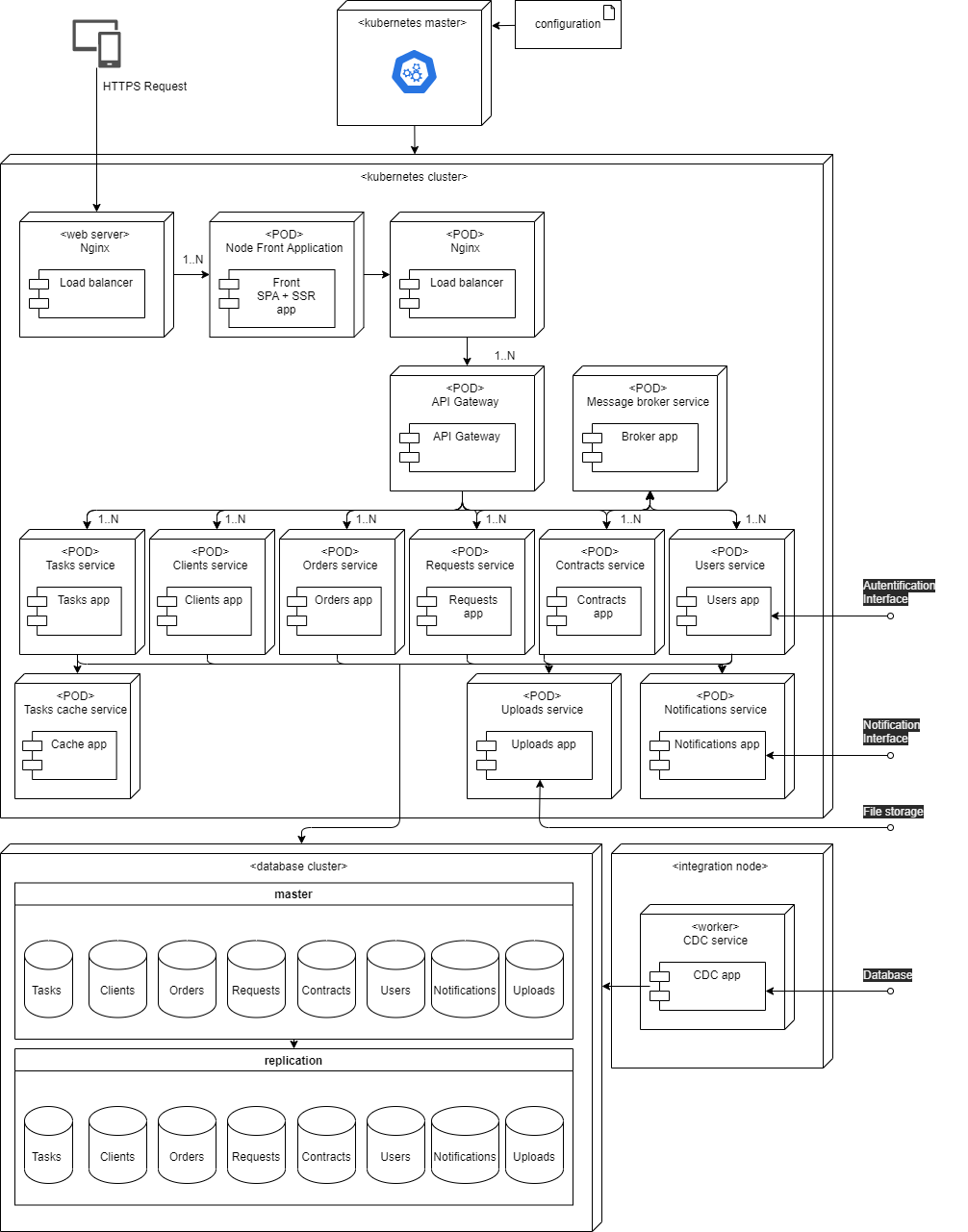
##### Сравнительная таблица производительности ЯП

Использовались данные с сайта <https://benchmarksgame-team.pages.debian.net/benchmarksgame/>

| **Тест** | **PHP 8.1.5 (cli)** | **GoLang (1.18)** |
| --- | --- | --- |
| sqrt in loop | 22.14ms | 7.42ms |
| regexp | 924ms | 297ms |
| binary-tree | 830ms | 240ms |
| file IO | 58ms | 17ms |

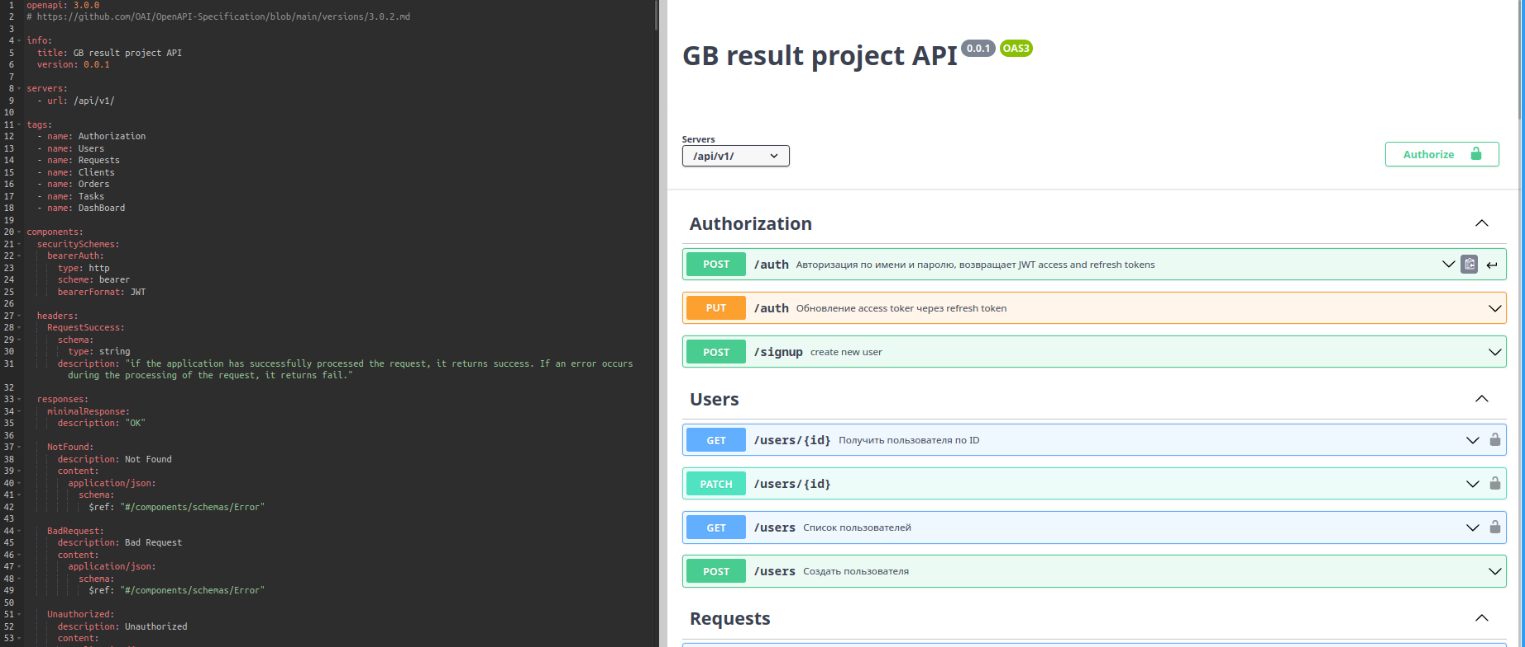
#### Модель развертывания приложений

Диаграмма развертывания

[](https://github.com/mrr000/gb-coursework/blob/master/Deployment/deployment.png)

#### Open Api

Open API

[](https://github.com/mrr000/gb-coursework/blob/master/swagger/swagger.png)

### Требования и решения по безопасности

* Система должна удовлетворять требованиям Постановления Правительства РФ от 01.11.2012 N 1119 "Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных".
* Gitlab Security Analysis не должен находить уязвимостей в компонентах системы в ходе сканирования.