

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«Смарт-Ком»

«Opti-Track TMS – система управления транспортом»
Описание архитектуры

Версия 1.0

Аннотация

Настоящий документ содержит описание архитектурного решения автоматизированной информационной системы управления перевозками OptiTrack TMS, предназначенной для повышения эффективности и автоматизации ключевых бизнес-процессов организации в области грузоперевозок.

Документ разработан в соответствии с РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов» и предназначен для архитекторов, разработчиков, администраторов и сопровождающих специалистов системы OptiTrack TMS.

СОДЕРЖАНИЕ

Аннотация	2
СОДЕРЖАНИЕ	3
1 ОБЩИЕ ПОЛОЖЕНИЯ.....	5
1.1 Полное наименование системы	5
1.2 Назначение документа	5
1.3 Назначение системы	5
1.4 Перечень нормативных документов.....	5
1.5 Перечень сокращений и терминов.....	6
2 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ.....	7
2.1 Контекст системы.....	7
2.2 Состав контекста	9
2.3 Контейнерная диаграмма (C4 Container)	10
2.3.1 Основные элементы диаграммы	10
2.3.2 Состав контейнеров.....	12
2.4 Состав платформы ProSpace	13
2.5 Состав сервисов системы OptiTrack TMS.....	14
2.6 Структура потоков данных.....	17
2.7 Общая коммуникационная структура	18
3 ТРЕБОВАНИЯ К ИНФРАСТРУКТУРЕ.....	19
3.1 Состав компонентов инфраструктуры	19
3.2 Характеристики компонентов инфраструктуры	20
4 ОПИСАНИЕ ПЛАТФОРМЫ PROSPACE.....	22
4.1 Назначение платформы	22
4.2 Принципы межсервисного взаимодействия	22
4.3 Структура пакетов платформы	23
4.4 Компоненты ядра платформы	23
4.5 Структура вызова через API Gateway	25
4.6 Структура межсервисного вызова	27
4.7 Аутентификация внешних вызовов.....	28
4.7.1 Внешний аутентификатор	28
4.7.2 Внутренний аутентификатор	30
5 АРХИТЕКТУРА PROSPACE FRONTEND WEBAPP	33

5.1	Общая структура	33
5.2	Структура ядра приложения	35
5.3	Невизуальные компоненты	35
6	ПРОЦЕДУРА РАЗВЁРТЫВАНИЯ ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ	36
6.1	Подготовка ядра проекта	36
6.2	Сборка и выпуск основного проекта	38
6.3	Подготовка инфраструктуры	40
6.3.1	Исходное состояние	40
6.3.2	Подготовка и конфигурирование элементов инфраструктуры	40
	Лист регистрации изменений	42

1 ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Полное наименование системы

Полное наименование: автоматизированная информационная система управления перевозками OptiTrack TMS.

Сокращённое наименование: OptiTrack TMS, TMC.

1.2 Назначение документа

Настоящий документ описывает архитектурное решение автоматизированной информационной системы управления перевозками, предназначенной для повышения эффективности и автоматизации ключевых бизнес-процессов организации в области грузоперевозок.

1.3 Назначение системы

Система OptiTrack TMS предназначена для:

- Автоматизации диспетчеризации и координации перевозок.
- Управления мастер-данными по транспортным средствам, маршрутам и контрагентам.
- Проведения электронных аукционов и тендеров на выполнение перевозок.
- Управления электронными транспортными накладными (эТРН) и интеграции с ГИС ЭПД.
- Планирования железнодорожных, контейнерных и автомобильных перевозок.
- Управления временными слотами погрузки/разгрузки.
- Расчёта KPI работы транспортно-экспедиционных компаний (ТЭК).
- Взаимодействия с внешними системами SAP ERP, SAP EWM, РЖД ЭТРАН, ГИС ЭПД.

1.4 Перечень нормативных документов

При разработке настоящего документа использованы следующие нормативные документы:

- ГОСТ 34.602–2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание (развитие или модернизацию) автоматизированной системы».

- РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов».
- ГОСТ Р 59792–2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем».

1.5 Перечень сокращений и терминов

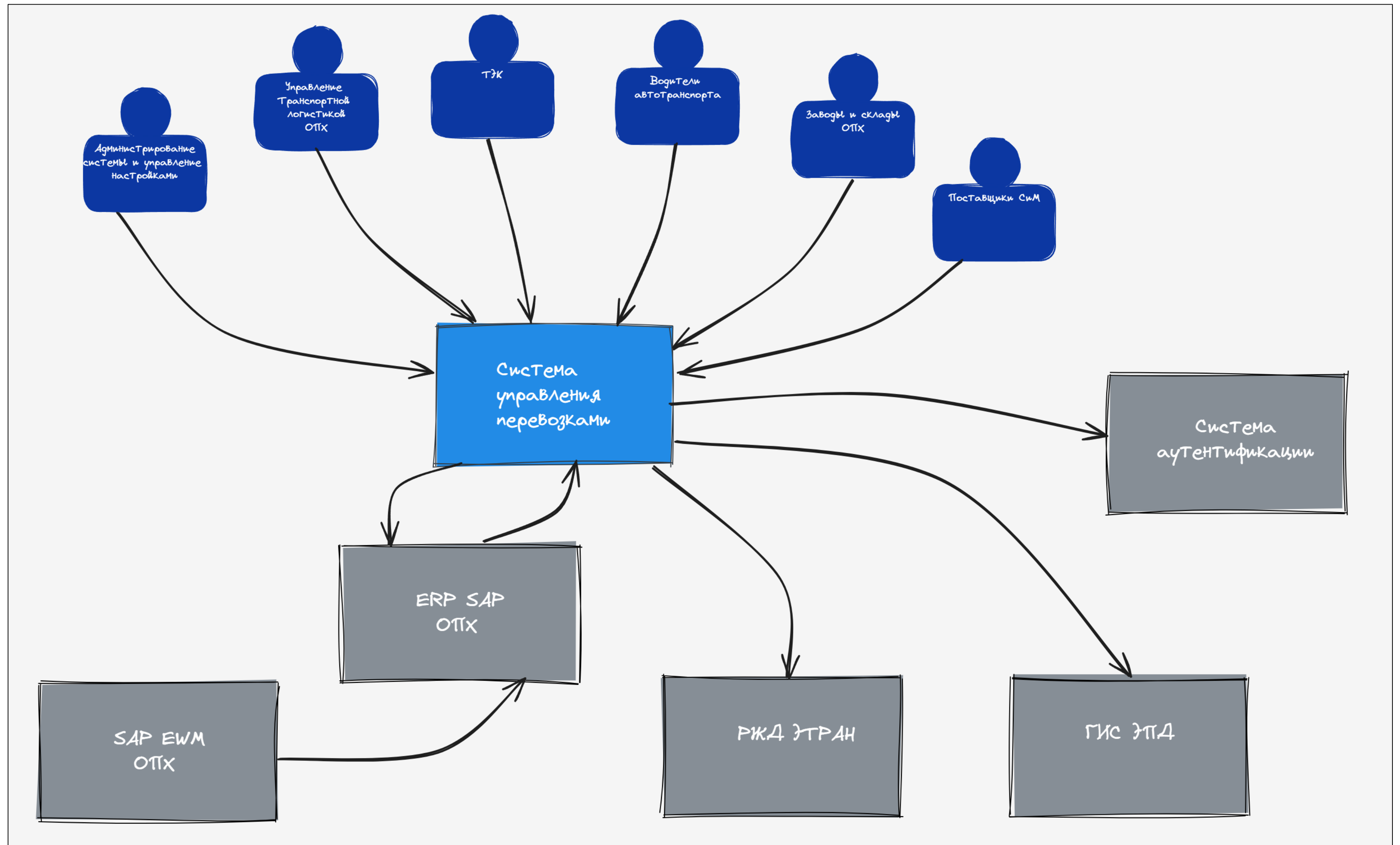
Сокращение	Расшифровка
API	Application Programming Interface — программный интерфейс приложения
API Gateway	Шлюз API — точка входа всех внешних обращений к системе
ADFS	Active Directory Federation Services
BI	Business Intelligence
CDC	Change Data Capture
DFD	Data Flow Diagram (диаграмма потоков данных)
ERP	Enterprise Resource Planning
EWM	Extended Warehouse Management
KPI	Key Performance Indicator
OData	Open Data Protocol
OSM	OpenStreetMap
S3	Simple Storage Service (объектное хранилище)
SAP	Информационная система класса ERP/EWM от SAP SE
ТЭК	Транспортно-экспедиционная компания
TMC / TMS	Transportation Management System
ЭТРН	Электронная транспортная накладная
ГИС ЭПД	Государственная информационная система электронных перевозочных документов
ОС	Операционная система
СУБД	Система управления базами данных

2 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ

2.1 Контекст системы

Архитектура системы состоит из ряда взаимосвязанных компонентов, каждый из которых выполняет определённую роль в обеспечении функциональности всей системы.

Контекстная диаграмма представляет пути передачи данных и взаимодействия между различными компонентами системы и внешними объектами. Каждая связь несёт информацию о направлении потока данных и характере взаимодействия.



2.2 Состав контекста

На контекстной диаграмме отражены следующие элементы:

Контекст системы:

- OptiTrack TMS — центральный элемент диаграммы, представляющий основную систему, с которой взаимодействуют другие компоненты.

Внешние системы:

- SAP EWM — внешняя система управления складом.
- SAP ERP — корпоративная ERP-система организации.
- РЖД «ЭТРАН» — система сопровождения процесса железнодорожных перевозок.
- ГИС ЭПД — государственная информационная система электронных перевозочных документов, используемая для электронного документооборота ЭТРН.
- Система аутентификации — внешний провайдер аутентификации сотрудников компании.

Пользователи:

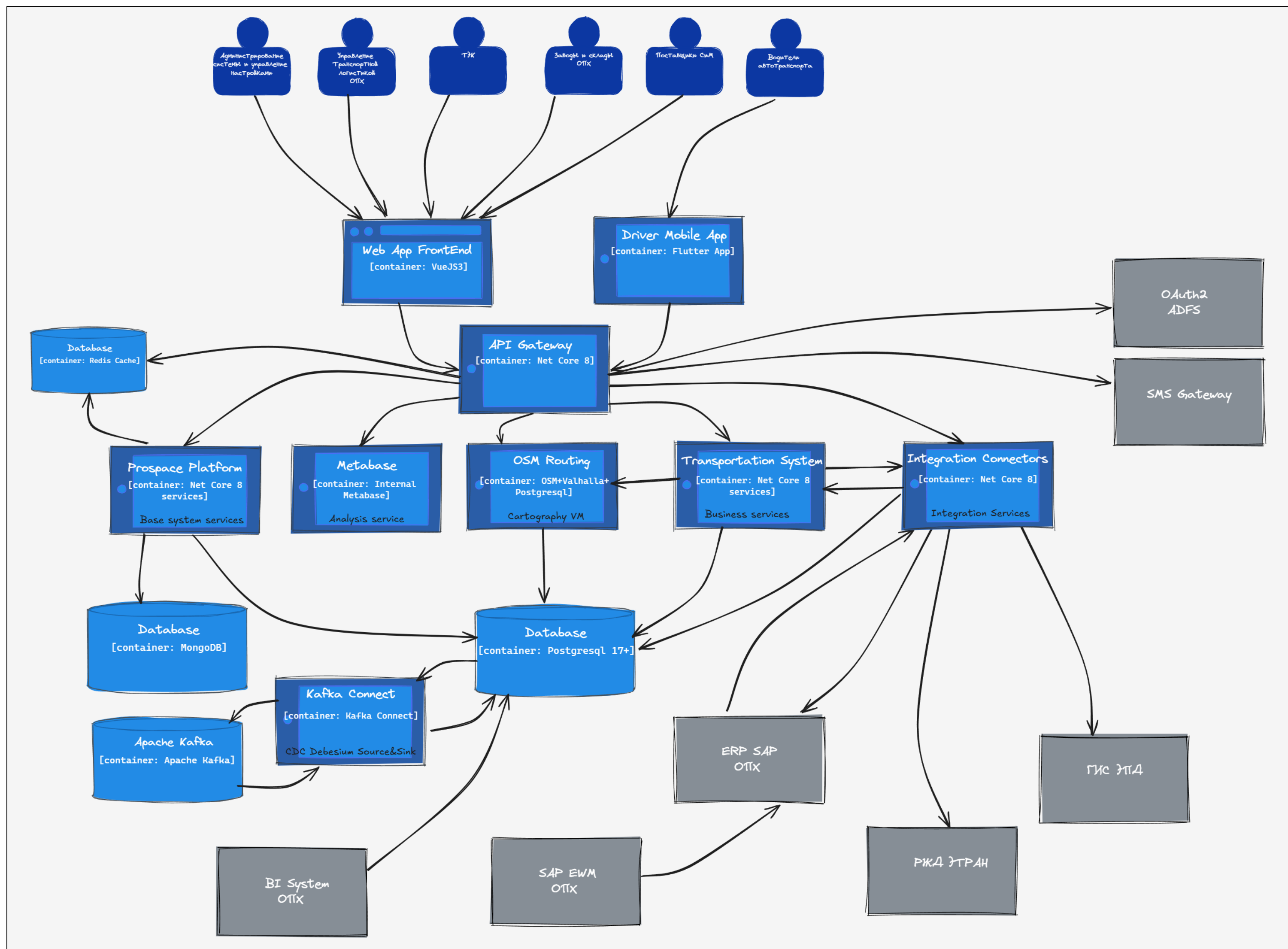
- Сотрудники компании, управляющие транспортной логистикой, — выполняют функции диспетчеризации и координации перевозок и являются основными потребителями механизмов автоматизации планирования.
- Администраторы и специалисты по управлению настройками — управляют доступом, системными настройками, базовыми типами, нотификациями и интеграциями.
- Пользователи транспортной компании — представители транспортно-экспедиционных компаний.
- Водители автотранспорта — водители, осуществляющие перевозки автомобильным транспортом, зафиксированные в системе.
- Заводы и склады — пользователи, обеспечивающие операции на точках погрузки и выгрузки.
- Поставщики сырья — представители поставщиков сырья и материалов в системе.

2.3 Контейнерная диаграмма (C4 Container)

Диаграмма контейнеров (container diagram) в C4-модели архитектуры программного обеспечения используется для визуализации высокоуровневой структуры системы и показывает компоненты («контейнеры»), взаимодействующие друг с другом.

2.3.1 Основные элементы диаграммы

- Контейнер — компонент уровня приложения или службы (веб-приложение, база данных, микросервис, API-шлюз и т. п.). Изображается прямоугольником со скруглёнными углами; внутри указывается название и используемая технология.
- Связи между контейнерами — обозначаются стрелками с указанием типа взаимодействия (синхронный HTTP-запрос, асинхронная очередь сообщений, обращение к БД).
- Внешние системы — обозначены отдельными контейнерами с соответствующей меткой.
- Описание технологии — для каждого контейнера указывается стек реализации (Net Core 8, PostgreSQL, Vue.js 3 и др.).



2.3.2 Состав контейнеров

На контейнерной диаграмме отражены следующие элементы:

1. Фронтенд и мобильные приложения:

- Web App Frontend (container: Vue.js 3).
- Driver Mobile App (container: Flutter).

2. Шлюзы и API:

- API Gateway (container: .NET Core 8).

3. Сервисы маршрутизации и транспорта:

- OSM Routing (container: .NET Core 8 + OpenStreetMap).
- Transportation System (container: .NET Core 8).

4. Интеграционные сервисы:

- Integration Connectors (container: .NET Core 8)

5. Аналитика и метрики:

- Metabase (container: Internal services).

6. Базы данных и кэш:

- Database (основная) (container: PostgreSQL 17+).
- Database (токены, настройки) (container: MongoDB).
- Database cache (container: Redis).

7. Система CDC (фиксация изменений в данных):

- Apache Kafka (container: Apache Kafka).
- Kafka Connect (container: Kafka Connect).

8. Внешние системы:

- BI System — система бизнес-аналитики OptiTrack.
- SAP EWM — внешняя система управления складом.
- РЖД ЭТРАН — система сопровождения железнодорожных перевозок.
- ГИС ЭПД — система электронных перевозочных документов.

9. Системы аутентификации и коммуникации:

- OAuth2 / ADFS (container: внешний провайдер).
- SMS Gateway (container: внешний шлюз).

10. Базовая платформа:

- ProSpace Platform (container: системные сервисы).

2.4 Состав платформы ProSpace

В состав платформы ProSpace, помимо ядра платформы, подключаемого к каждому сервису системы, входят готовые к эксплуатации платформенные сервисы:

Сервис платформы	Назначение
Security Admin Service	Администрирование и управление доступом
Base Type Service	Управление типами и классификаторами
Notification Service	Управление и маршрутизация уведомлений
Scheduler Service	Управление фоновыми задачами и системными службами
File Store Service	Регистрация и управление файлами в системе

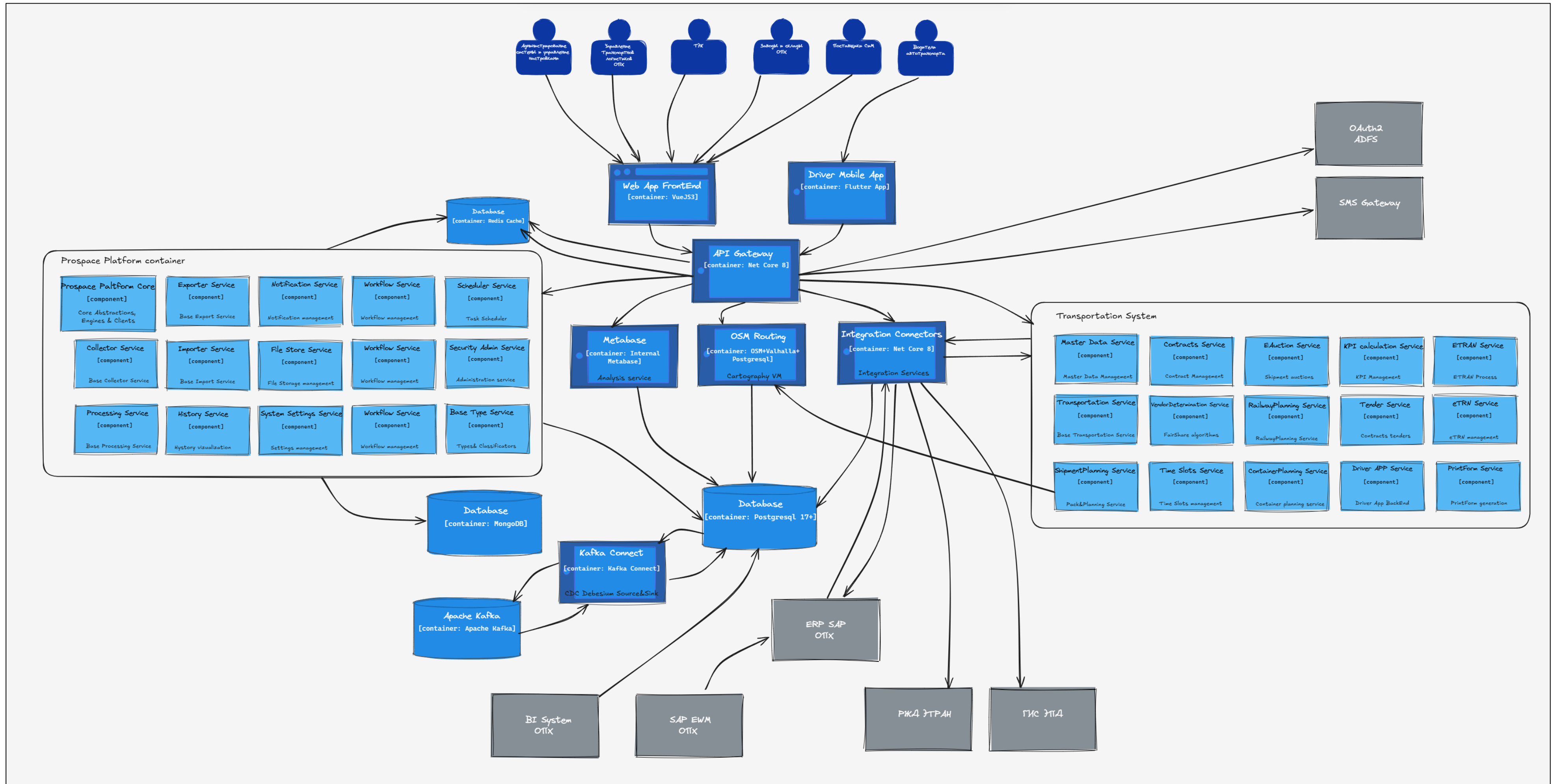
Сервис платформы	Назначение
System Settings Service	Управление системными настройками
History Service	Чтение и визуализация истории изменений данных
Workflow Service	Управление согласованиями и статусными переходами
Collector Service	Базовая регистрация интеграционных сообщений
Processing Service	Базовая обработка интеграционных сообщений
Exporter Service	Базовый экспорт данных из системы
Importer Service	Базовый импорт данных в систему

2.5 Состав сервисов системы OptiTrack TMS

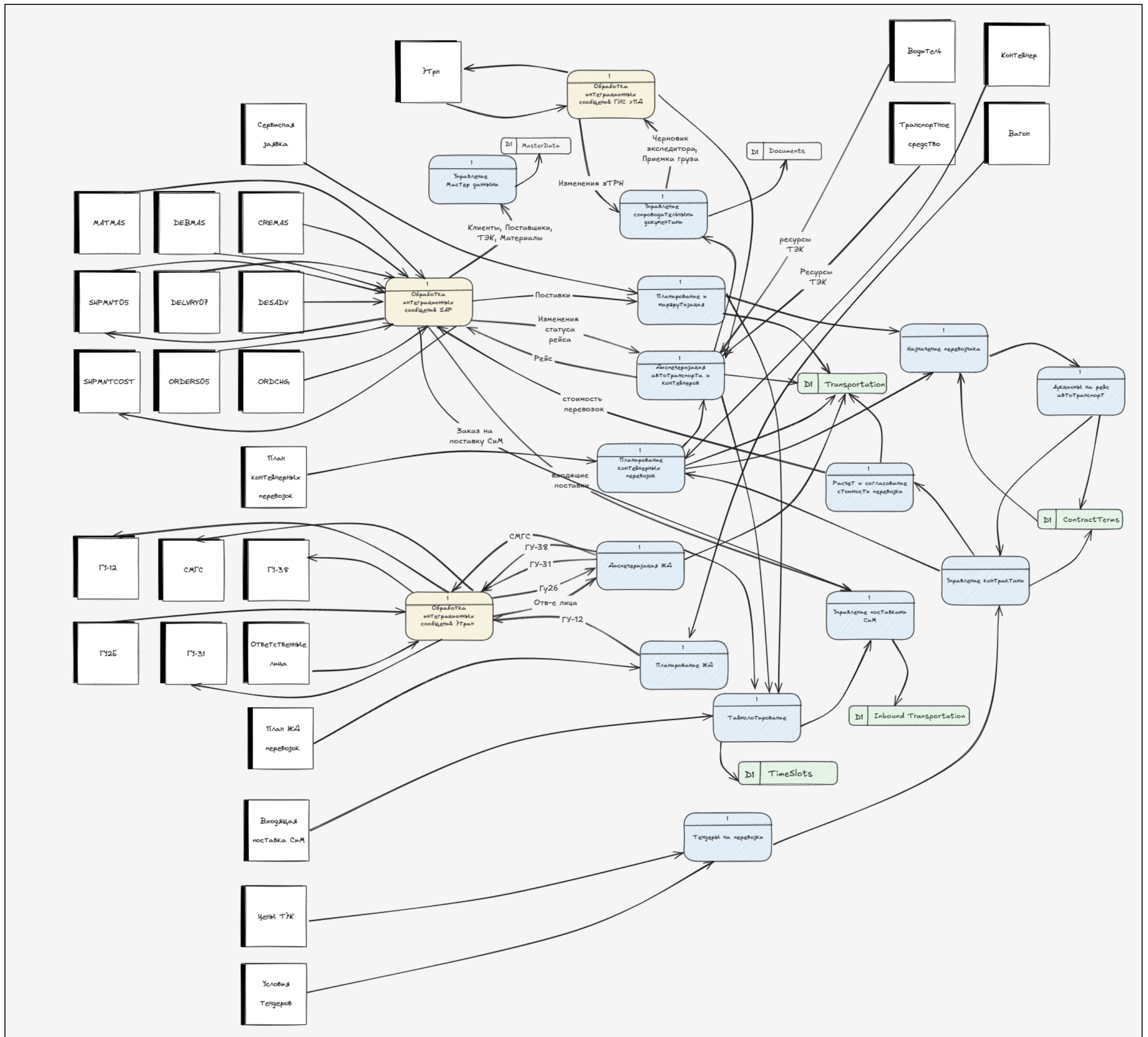
Система OptiTrack TMS включает следующие высокоуровневые бизнес-сервисы:

Сервис	Назначение
Master Data Service	Управление мастер-данными (транспортные средства, маршруты, ключевые справочники)
Contracts Service	Управление контрактами с перевозчиками — создание, редактирование, отслеживание
E-Auction Service	Проведение аукционов на выбор перевозчиков (Shipment Auction)
KPI Calculation Service	Расчёт и управление KPI для оценки работы ТЭК
ETRAN Service	Управление процессом обмена данными с РЖД ЭТРАН
Transportation Service	Базовый сервис управления транспортными операциями
Vendor Determination Service	Алгоритм справедливого распределения перевозок (FairShare)
Railway Planning Service	Планирование железнодорожных перевозок
Tender Service	Управление тендерами на заключение контрактов
eTRN Service	Управление электронными транспортными накладными
Shipment Planning Service	Планирование перевозок
Time Slots Service	Управление временными слотами погрузки/разгрузки

Сервис	Назначение
Container Planning Service	Планирование контейнерных перевозок
Driver App Service	Бэкэнд мобильного приложения водителя
PrintForm Service	Генерация печатных форм документов перевозок
Pack&Planning Service	Планирование упаковки и погрузки

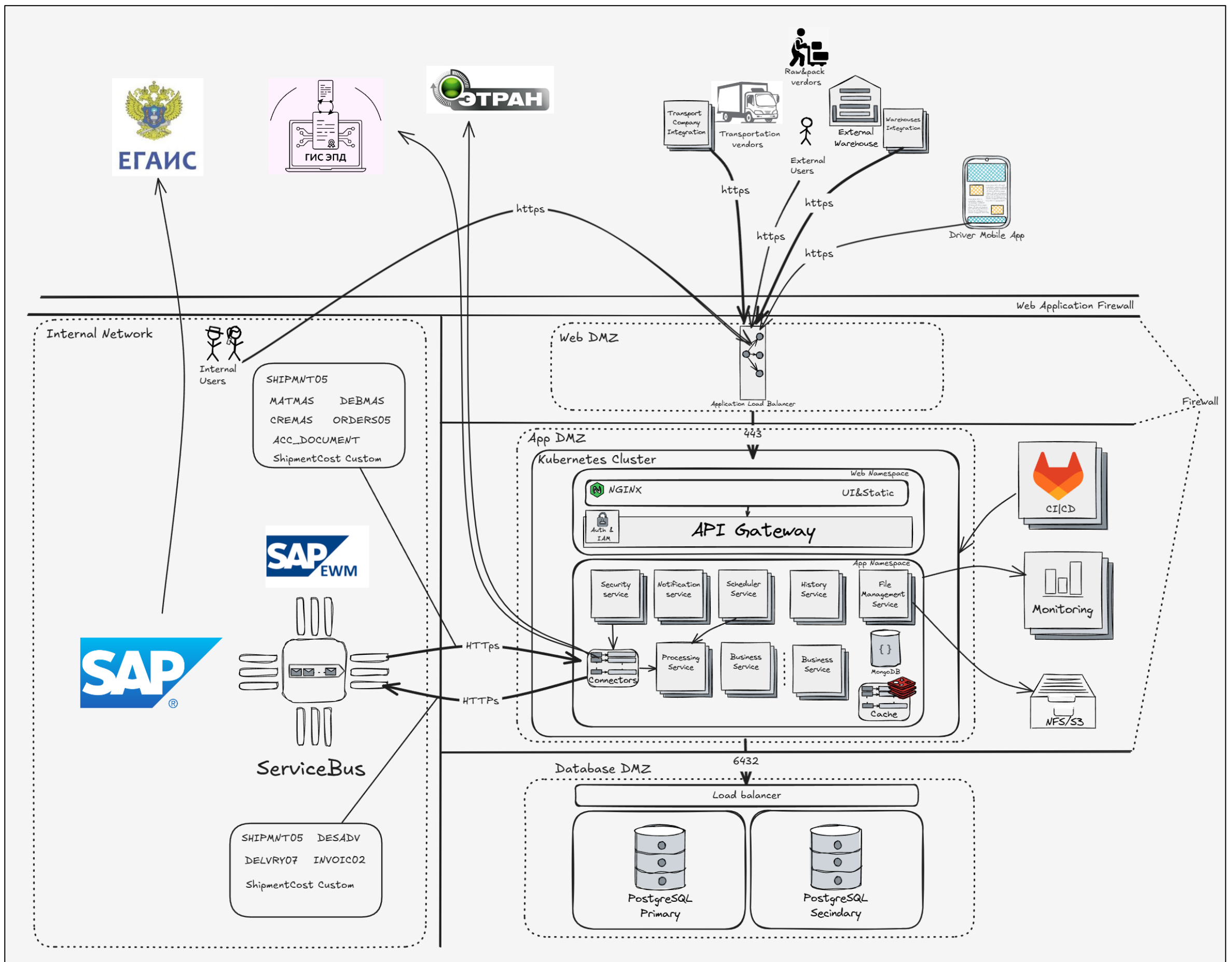


Структура кода системы представлена в виде диаграммы.



27 Общакommунuкационная структура

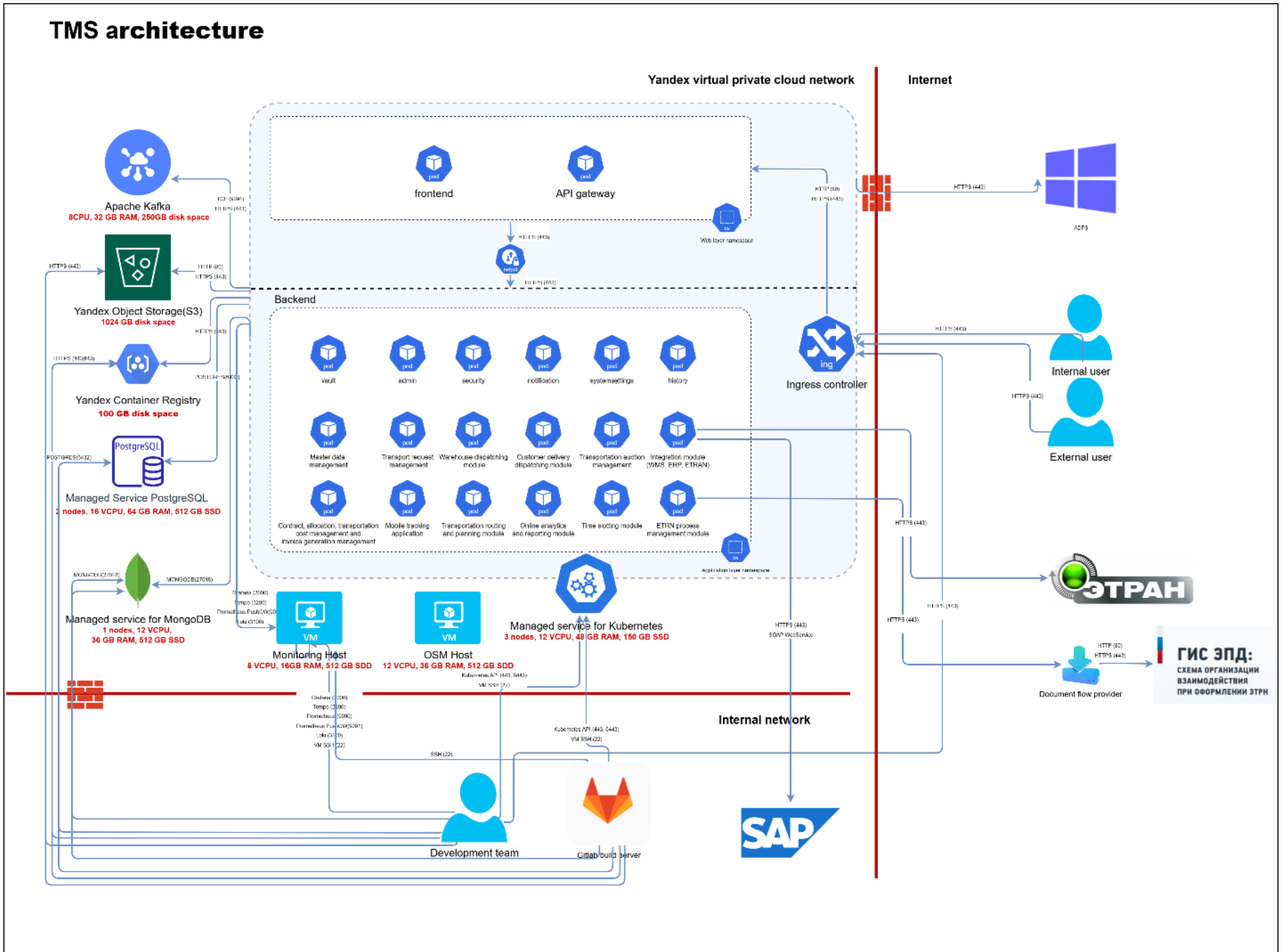
Общакommунuкационная структура системы отражает разделение конечных точек по сетевым сегментам (Internal Network, Web DMZ, App DMZ, Database DMZ), а также внешние интеграции с системами SAP, РЖД, ГРАНИС, ЭДИАИС.



3 ТРЕБОВАНИЯ КИНФРАСТРУКТУРЕ

3.1 Состав компонентов инфраструктуры

Инфраструктура системы OptiTrack™ работает в облачном сервисе Яндекс.Облако (Yandex Cloud) и включает следующие компоненты:



3.2 Характеристики компонентов инфраструктуры

Ниже приведены параметры конфигурирования компонентов инфраструктуры в Yandex Cloud, соответствующие текущему проектному решению.

Наименование характеристики	Значение характеристики
Кластер Kubernetes	
Тип мастера	Высокодоступный
Платформа	Intel lake (Compute Optimized)
Гарантированная доля vCPU	100%
Количество vCPU	12
Объем RAM	48 ГБ
Диск	Network-hdd
Размер диска	150 Гб
Количество узлов	3
PostgreSQL	
Платформа	Intel Ice Lake
Тип хоста	standard
Класс хоста	S3-c16-m64 (vCPU 16 cores/ Память 64 Гб)
Тип хранилища	Network-ssd
Размер хранилища	512 Гб
Количество хостов в кластере	2
MongoDB	
Платформа	Intel Ice Lake
Тип хоста	standard

Наименование характеристики	Значение характеристики
Класс хоста	S3-c8-m32 (vCPU 8 cores/ Память 32 Гб)
Тип хранилища	Network-ssd
Размер хранилища	512 Гб
Количество хостов в кластере	1
Monitoring Host	
Платформа	Intel Cascade Lake
Гарантированная доля vCPU	100%
Количество vCPU	8
Объем RAM	16 Гб
Диск загрузочный	Network-ssd
Размер диска	245 Гб

4 ОПИСАНИЕ ПЛАТФОРМЫ PROSPACE

4.1 Назначение платформы

Платформа ProSpace задаёт единую архитектуру всех бизнес-сервисов, реализуемых в рамках разработки на её базе, закрепляет общий принцип управления безопасностью, стандартизирует подход к внутренним межсервисным вызовам, обеспечивает единый механизм обработки запроса, а также вносит требования к описанию доменных моделей и моделей API, задавая контракты на вызовы, доступные внутри системы.

Наименование характеристики	Значение характеристики
Apache Kafka	
Платформа	Intel Ice Lake
Тип хоста	standard
Класс хоста	S3-c8-m32 (vCPU 8 cores/ Память 32 Гб)
Тип хранилища	Network-ssd
Размер хранилища	250 Гб
Количество хостов в кластере	1

4.2 Принципы межсервисного взаимодействия

В рамках платформы стандартизирован подход к реализации и интеграции в систему вертикального пайплайна обработки запроса; запрос данных выполняется с использованием конструкций стандарта OData.

Межсервисное взаимодействие построено на основе платформенного компонента с реализацией абстракции ядра платформы «Client». Такой подход позволил обеспечить разделение сервисов как горизонтально (по функциям бизнес-контекста), так и вертикально (с выделением части работы с данными на уровне доменного репозитория).

Общие абстракции, компоненты и middleware, используемые в структуре платформенного сервиса, находятся в ядре платформы и являются основой для реализации базовой архитектуры. Все готовые платформенные сервисы построены по единому принципу, соответствующему общему подходу к разработке.

4.3 Структура пакетов платформы

Платформа ProSpace поставляется в виде единого NuGet-пакета, но архитектурно состоит из нескольких логических пакетов, представленных на диаграмме пакетов с выделенными уровнями абстракции.

Наименование характеристики	Значение характеристики
OSM Host	
Платформа	Intel Cascade Lake
Гарантированная доля vCPU	100%
Количество vCPU	12
Объем RAM	36 ГБ
Диск загрузочный	Network-ssd
Размер диска	512 Гб

Структурно пакеты состоят из компонентов ядра, каждый из которых представляет определённый уровень абстракции — от базовых абстракций Prospace.Infrastructure.Abstraction до уровня прикладных компонентов (Cache, History и т. д.).

4.4 Компоненты ядра платформы

Компоненты ядра платформы и их интерфейсы представлены на диаграмме компонентов.

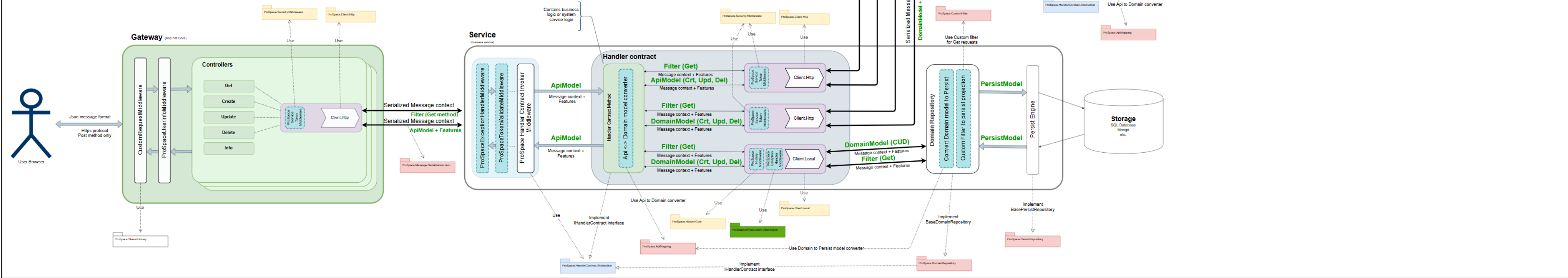
Наименование характеристики	Значение характеристики
File Storage (S3)	
Тип хранилища	Стандартное
Размер хранилища	1024 Гб

Detailed communication process

Основной подход использования заложенный в платформе

Определения

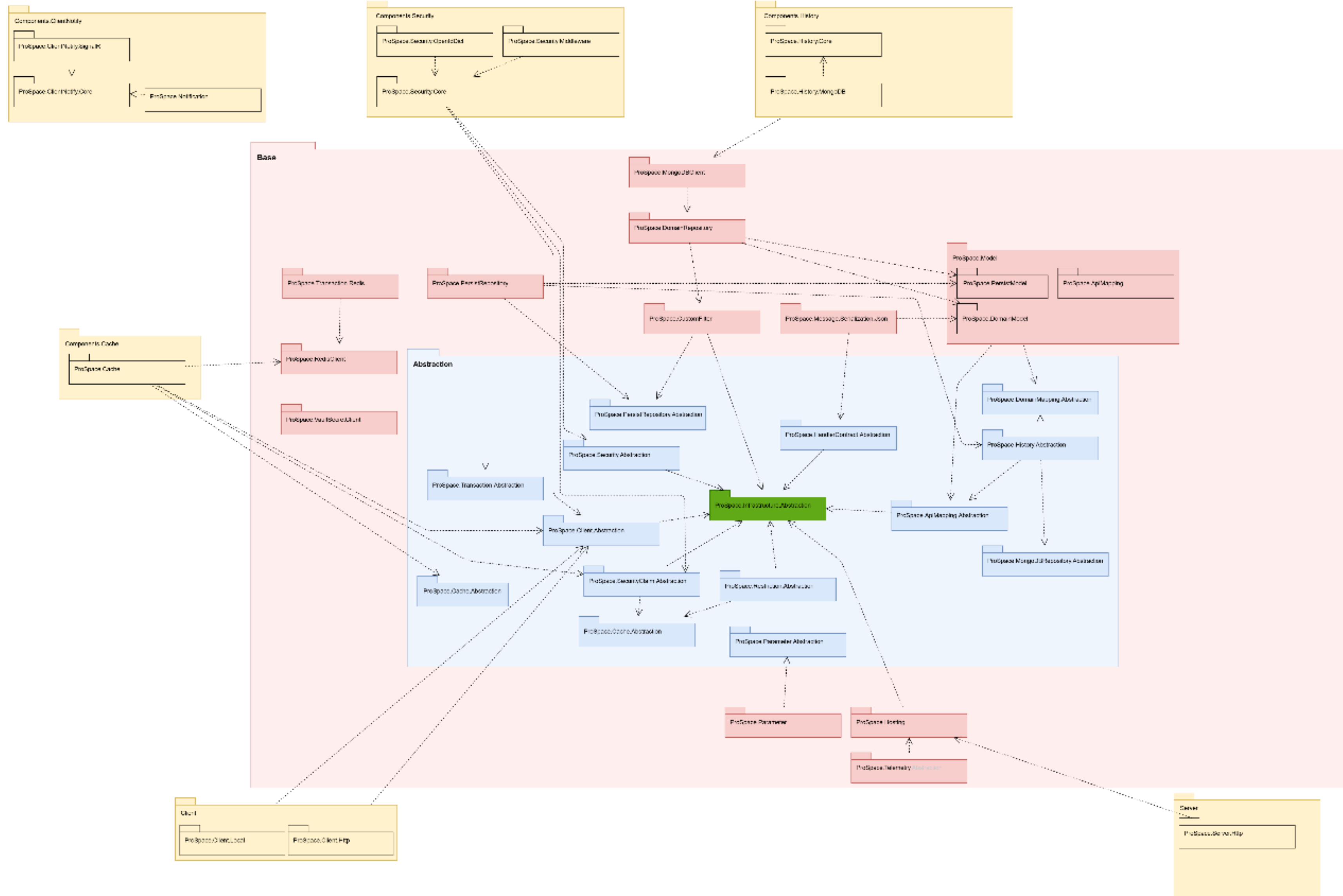
- Компонент расширения. Может быть присутствовать/отсутствовать или иметь перегрузку базовых методов реализации
- Обязательный компонент. Отсутствие компонента делает работу платформы не возможной



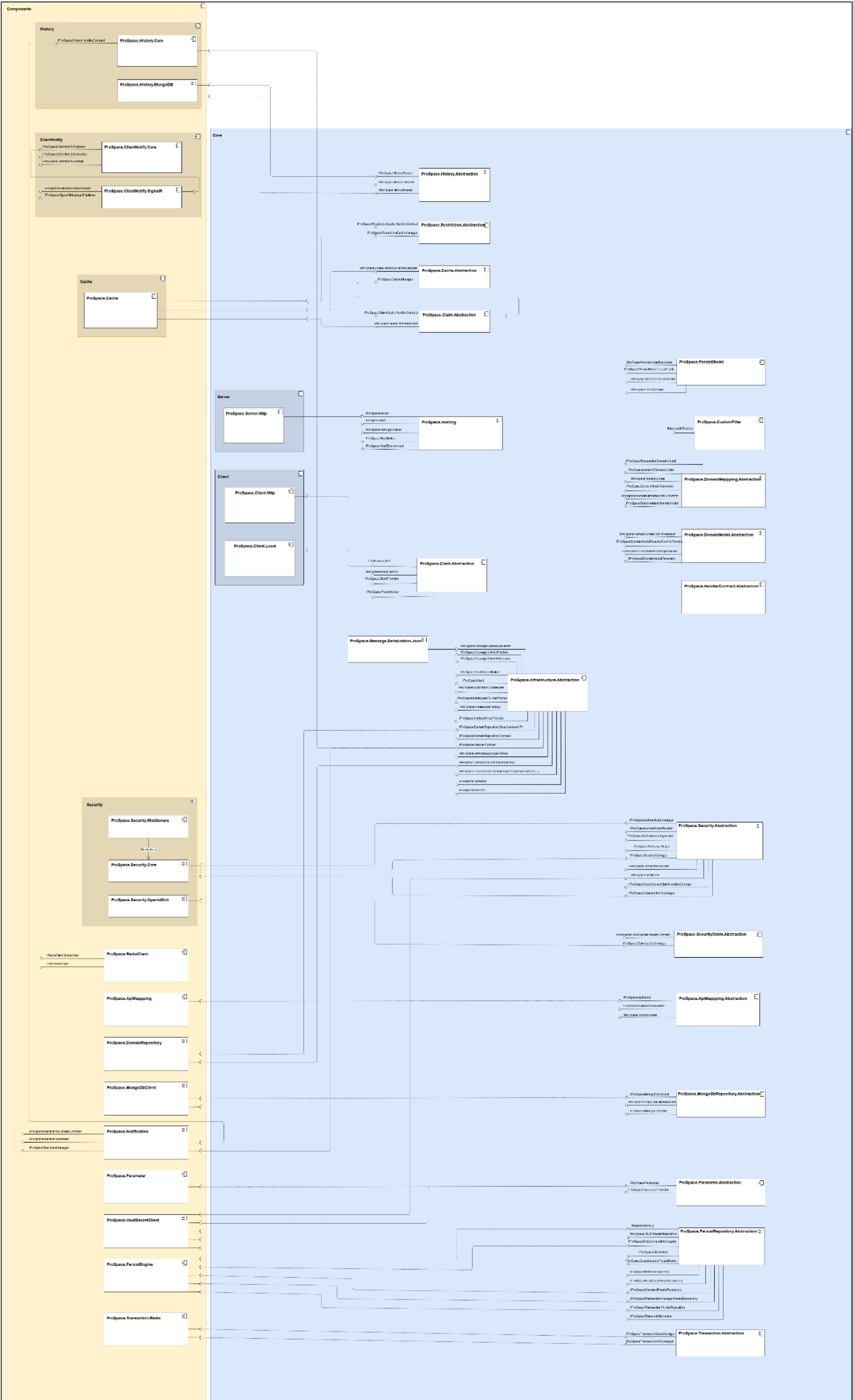
4.5 Структура вызова через API Gateway

При использовании абстракций платформы в межсервисном взаимодействии жёстко регламентируется сериализация структур данных при внутренних и внешних вызовах. Внешние вызовы поступают строго на API Gateway, который является модификацией стандартного WebAPI-приложения ASP.NET Core. Базовая структура Gateway также находится в ядре платформы, а набор платформенных Middleware обеспечивает аутентификацию и авторизацию всех обращений к системе, десериализацию параметров вызова и передачу вызова к внутренним сервисам с использованием абстракции «Client» ядра платформы.

Components



Внутренний восток API в автоматическом соответствии с другими компонентами



При выполнении вызовов внутри системы используется упрощённый механизм аутентификации, ориентированный на идентификацию сервиса-отправителя, с длительным периодом действия токена аутентификации.

4.7 Аутентификация внешних вызовов

При внешнем вызове различают вызовы с использованием внешнего и внутреннего (локального) аутентификатора.

4.7.1 Внешний аутентификатор

Вызов с использованием внешнего аутентификатора имеет следующий поток исполнения.

Gateway invoke messsge structure

<Наименование сервиса>/<Контроллер>/<Метод>

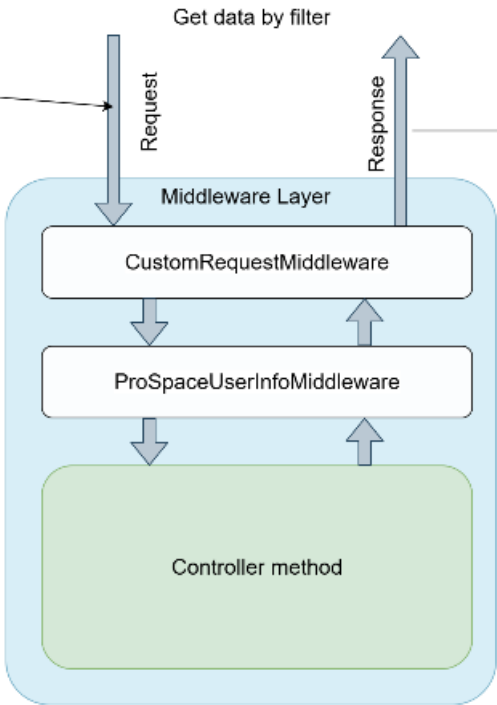
Header:

Authorization: Bearer E-1Q5qLKqTn4U42AoUNT-uBmXN4gR-RDe2fH273mORY

Body:

```
{
  "filter": "$filter=RoleName eq 'Administrator'&$top=50"
}
```

Структура сообщения (Пример)



```
{
  "success": true,
  "data": [{
    "id": 1,
    "code": "Perviy.user:1",
    "status": "ACTIVE",
    "firstName": "Alexander",
    "secondName": null,
    "lastName": "Perviy",
    "login": "alexander.Perviy",
    "email": "alexander.Perviy@email.com",
    "language": "EN",
    "disabled": false,
    "disableDate": null,
    "jobTitle": null,
    "comments": null,
    "timeZoneld": "GMT Standard Time",
    "dateFormat": "dd.MM.yyyy",
    "timeFormat": "HH:mm",
    "lastLoginDate": null,
    "securityLoginAuthenticateProvider": "Microsoft",
    "password": null,
    "createdBy": "SYSTEM",
    "updatedBy": null,
    "createdAt": "2023-03-06T10:26:55.472399",
    "lastUpdated": "2023-03-06T10:26:55.472399",
    "stamp": null,
    "additionalFields": []
  }, {
    "id": 2,
    "code": "Sergey.Kochetov:2",
    "status": "ACTIVE",
    "firstName": "Andrey",
    "secondName": null,
    "lastName": "Kochetov",
    "login": "Sergey.Kochetov",
    "email": "Sergey.Kochetov@google.com",
    "language": "EN",
    "disabled": false,
    "disableDate": null,
    "jobTitle": null,
    "comments": null,
    "timeZoneld": "GMT Standard Time",
    "dateFormat": "dd.MM.yyyy",
    "timeFormat": "HH:mm",
    "lastLoginDate": "2023-04-13T08:13:29.91074",
    "securityLoginAuthenticateProvider": "Microsoft",
    "password": null,
    "createdBy": "SYSTEM",
    "updatedBy": "Sergey.Kochetov:2",
    "createdAt": "2023-03-06T10:26:55.472399",
    "lastUpdated": "2023-04-13T08:13:29.917744",
    "stamp": "RdRb9/Y72wg=",
    "additionalFields": []
  }
  ],
  "errors": null
}
```

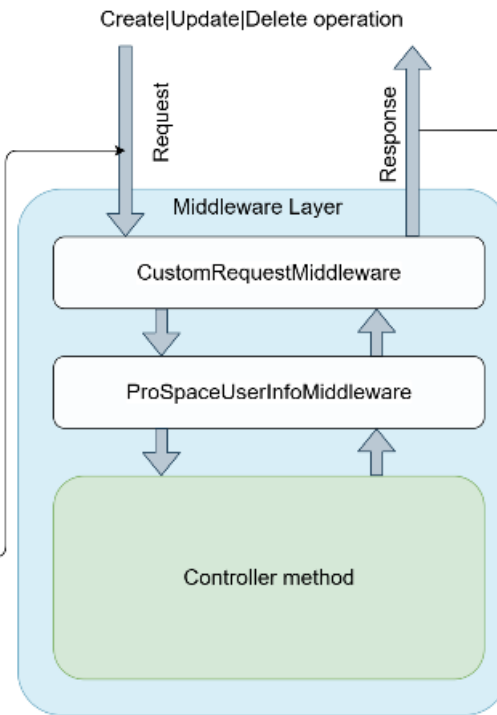
Header:

Authorization: Bearer E-1Q5qLKqTn4U42AoUNT-uBmXN4gR-RDe2fH273mORY

Body:

```
{
  "data": [{
    "id": 11,
    "code": "alexey.pashkevich:11",
    "status": "ACTIVE",
    "fullName": "Pashkevich Alexey",
    "firstName": "Alexey",
    "secondName": null,
    "lastName": "Pashkevich",
    "login": "alexey.pashkevich",
    "email": "alexey.pashkevich@smartcom.software",
    "language": "EN",
    "disabled": false,
    "disableDate": null,
    "jobTitle": null,
    "comments": null,
    "timeZoneld": "GMT Standard Time",
    "dateFormat": "DD.MM.YYYY",
    "timeFormat": "HH:mm",
    "lastLoginDate": "2023-04-17T12:47:54.540588",
    "securityLoginAuthenticateProvider": "Microsoft",
    "password": null,
    "orgLinkId": 65,
    "orgLinkType": "Distributor",
    "orgLinkMarsOrgId": null,
    "orgLinkDistributorId": 1,
    "orgLinkDistributorOrgId": 30,
    "orgLinkMarsOrgName": null,
    "orgLinkDistributorName": "ЗАО Андреева, Щербаков and Зиновьев",
    "orgLinkDistributorOrgName": "Сотрудник отдела закупки",
    "createdBy": "SYSTEM",
    "updatedBy": "alexey.pashkevich:11",
    "createdAt": "2023-01-26T07:51:12.04119",
    "lastUpdated": "2023-04-17T12:47:54.542911",
    "stamp": "TbmNb0U/2wg=",
    "additionalFields": []
  }
  ]
}
```

Структура сообщения (Пример)



```
{
  "success": true,
  "data": [{
    "id": 11,
    "code": "alexey.pashkevich:11",
    "status": "ACTIVE",
    "fullName": "Pashkevich Alexey",
    "firstName": "Alexey",
    "secondName": null,
    "lastName": "Pashkevich",
    "login": "alexey.pashkevich",
    "email": "alexey.pashkevich@smartcom.software",
    "language": "EN",
    "disabled": false,
    "disableDate": null,
    "jobTitle": null,
    "comments": null,
    "timeZoneld": "GMT Standard Time",
    "dateFormat": "DD.MM.YYYY",
    "timeFormat": "HH:mm",
    "lastLoginDate": "2023-04-17T12:47:54.540588",
    "securityLoginAuthenticateProvider": "Microsoft",
    "password": null,
    "orgLinkId": 65,
    "orgLinkType": "Distributor",
    "orgLinkMarsOrgId": null,
    "orgLinkDistributorId": 1,
    "orgLinkDistributorOrgId": 30,
    "orgLinkMarsOrgName": null,
    "orgLinkDistributorName": "ЗАО Андреева, Щербаков and Зиновьев",
    "orgLinkDistributorOrgName": "Сотрудник отдела закупки",
    "createdBy": "SYSTEM",
    "updatedBy": "alexey.pashkevich:11",
    "createdAt": "2023-01-26T07:51:12.04119",
    "lastUpdated": "2023-04-17T12:47:54.542911",
    "stamp": "VAaZ9kE/2wg=",
    "additionalFields": []
  }
  ],
  "errors": null
}
```

Для реализации цепочки используется библиотека OpenIddict, обеспечивающая высокий уровень надёжности, безопасности и гибкости при работе с различными аутентификаторами; библиотека не имеет ограничений на использование и является полностью открытой.

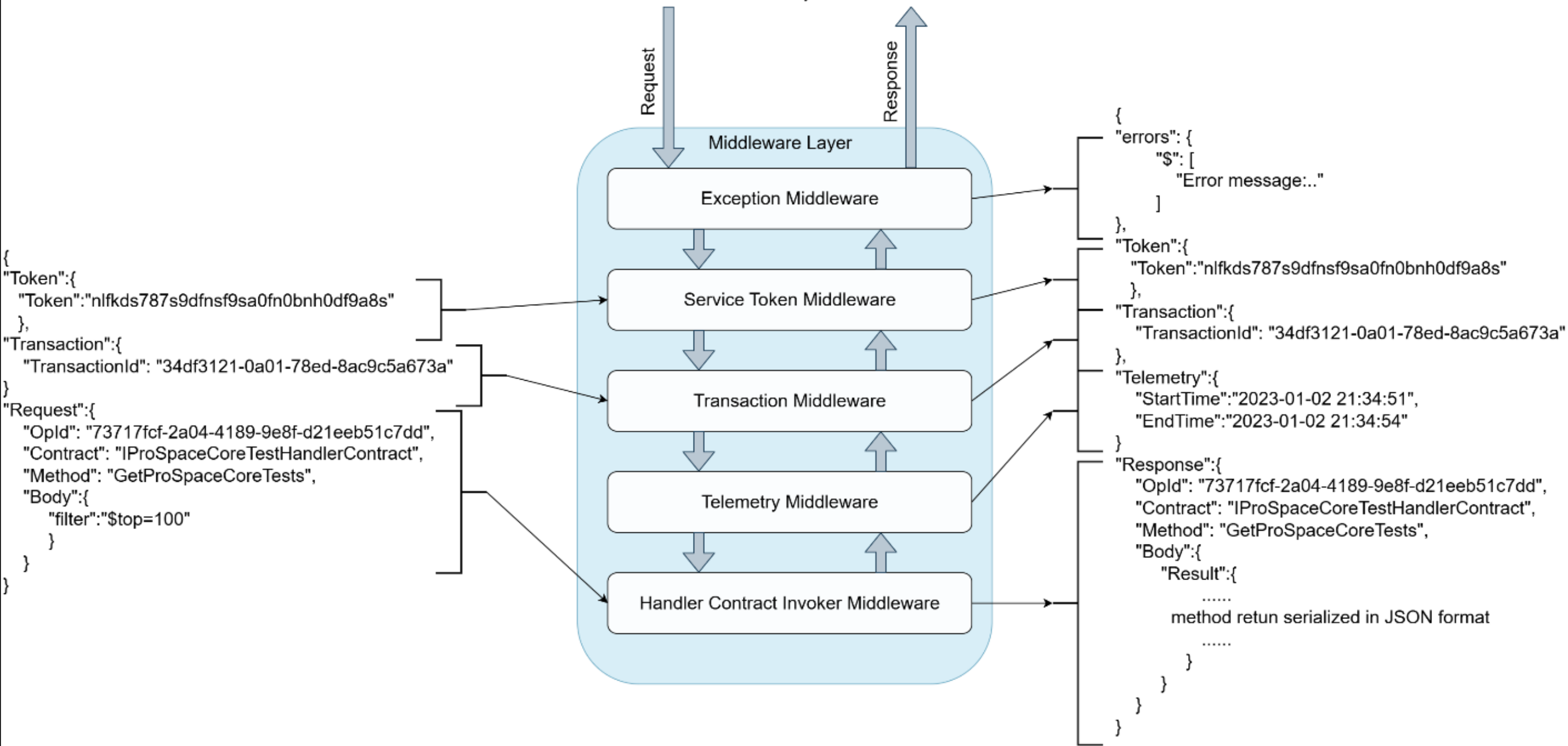
4.7.2 Внутренний аутентификатор

Service invoke message structure

<Наименование сервиса>
Контракт и метод указывается в разделе Request

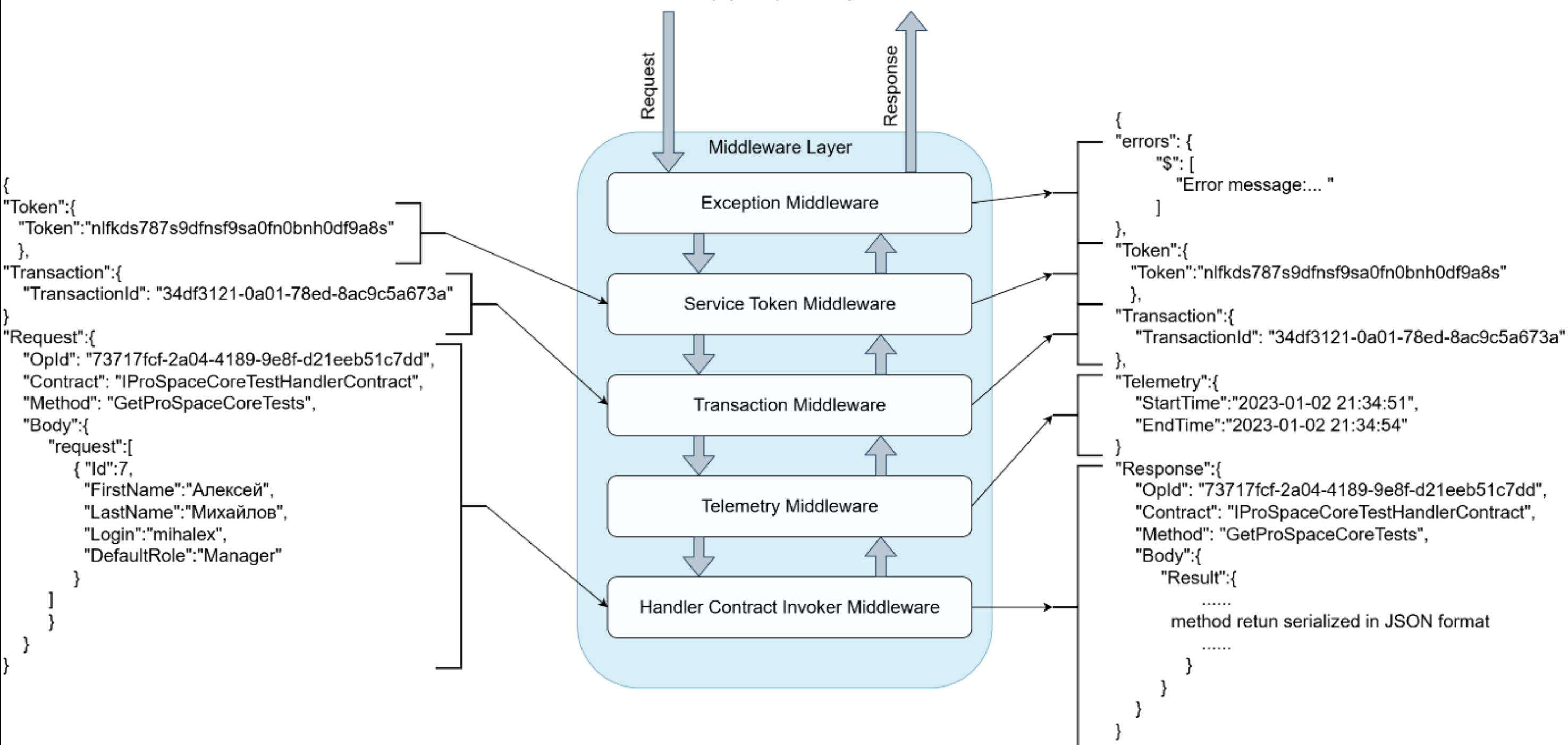
Структура сообщения (Пример)

Get data by filter



Структура сообщения (Пример)

Create|Update|Delete operation



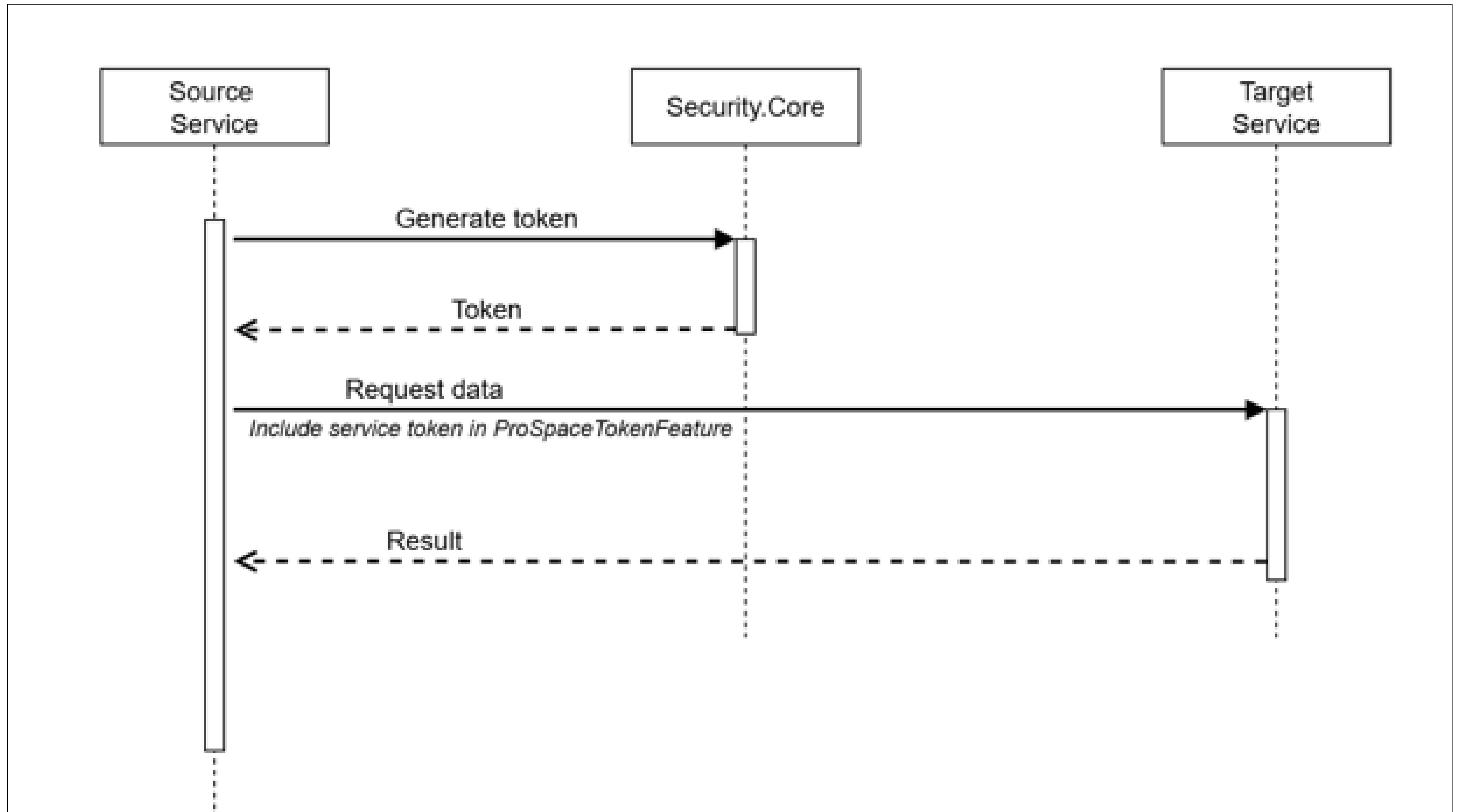
Вызов с использованием внутреннего аутентификатора в целом аналогичен внешнему, но отличается использованием встроенного в платформу механизма идентификации пользователя по паролю. Ротация пароля и соблюдение правил по длине, сложности и повторяемости обеспечивается встроенной платформенной службой, поставляемой в составе сервиса Security Admin.

5 АРХИТЕКТУРА PROSPACE FRONTEND WEBAPP

5.1 Общая структура

Платформенное решение для реализации FrontEnd WebApp представляет собой набор из трёх npm-пакетов: центрального ядра приложения, библиотеки визуальных компонентов на базе открытой библиотеки PrimeVue 4, а также библиотеки невизуальных компонентов для разметки страницы и генерации данных для визуализации в формах.

Реализация библиотек полностью выполнена на языке TypeScript с проверкой типизации. Интерфейс стандартного представления сущностей и моделей системы стандартизирован, но может быть изменён за счёт использования сторонних визуальных компонентов и механизмов наследования основных элементов архитектуры ядра.



5.2 Структура ядра приложения

Ядро приложения включает в себя несколько уровней базовых реализаций:

Слой	Назначение
API	прокси-слой обмена данных с API Gateway, средства аутентификации и системные вызовы
Auth	реализация client flow аутентификации для встроенного и сторонних аутентификаторов по протоколу OAuth2
Services	слой сервисов для работы с бизнес-моделями; экранирование приложения от особенностей расположения данных, фильтрация через FilterBuilder, массовые операции и обработка ошибок
FilterBuilder	инструмент формирования фильтров — как на основании цепочки выражений, так и через прямую интеграцию с визуальными компонентами через Transformers
Transformers	стандартизированные методы преобразования данных между визуальными и не визуальными компонентами
Brokers	основной компонент, реализующий иерархию страниц и содержащий всю frontend-бизнес-логику операций WebApp; формируется от корневого элемента — держателя главного окна приложения; использует один или более сервисов для обращения к данным

5.3 Невизуальные компоненты

Невизуальная часть включает в себя три основных размещения:

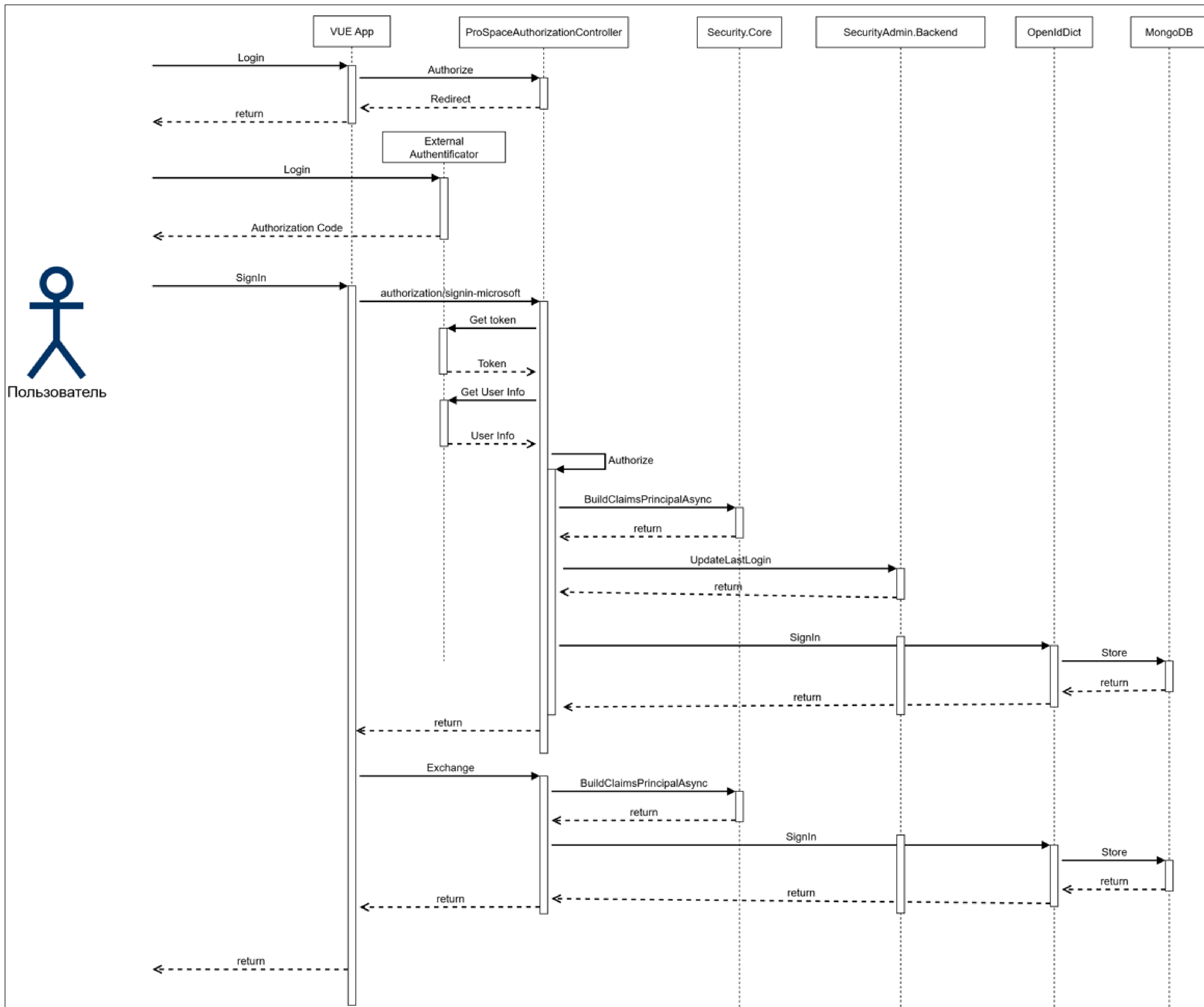
- GEOLayout — размещение в виде пяти зон (север, юг, восток, запад, центр).
- H-Layout — тиражирование по горизонтали.
- V-Layout — тиражирование по вертикали.

Также в библиотеку не визуальных компонентов включён построитель структуры для формирования различных форм для API-моделей — FormBuilder. FormBuilder представляет собой конфигурируемый инструмент, который формирует ViewModel для любого вида API-модели. Такая модель может быть использована в компонентах *Drawer из UI-библиотеки компонентов для отображения формы чтения или ввода данных.

6 ПРОЦЕДУРА РАЗВЁРТЫВАНИЯ ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ

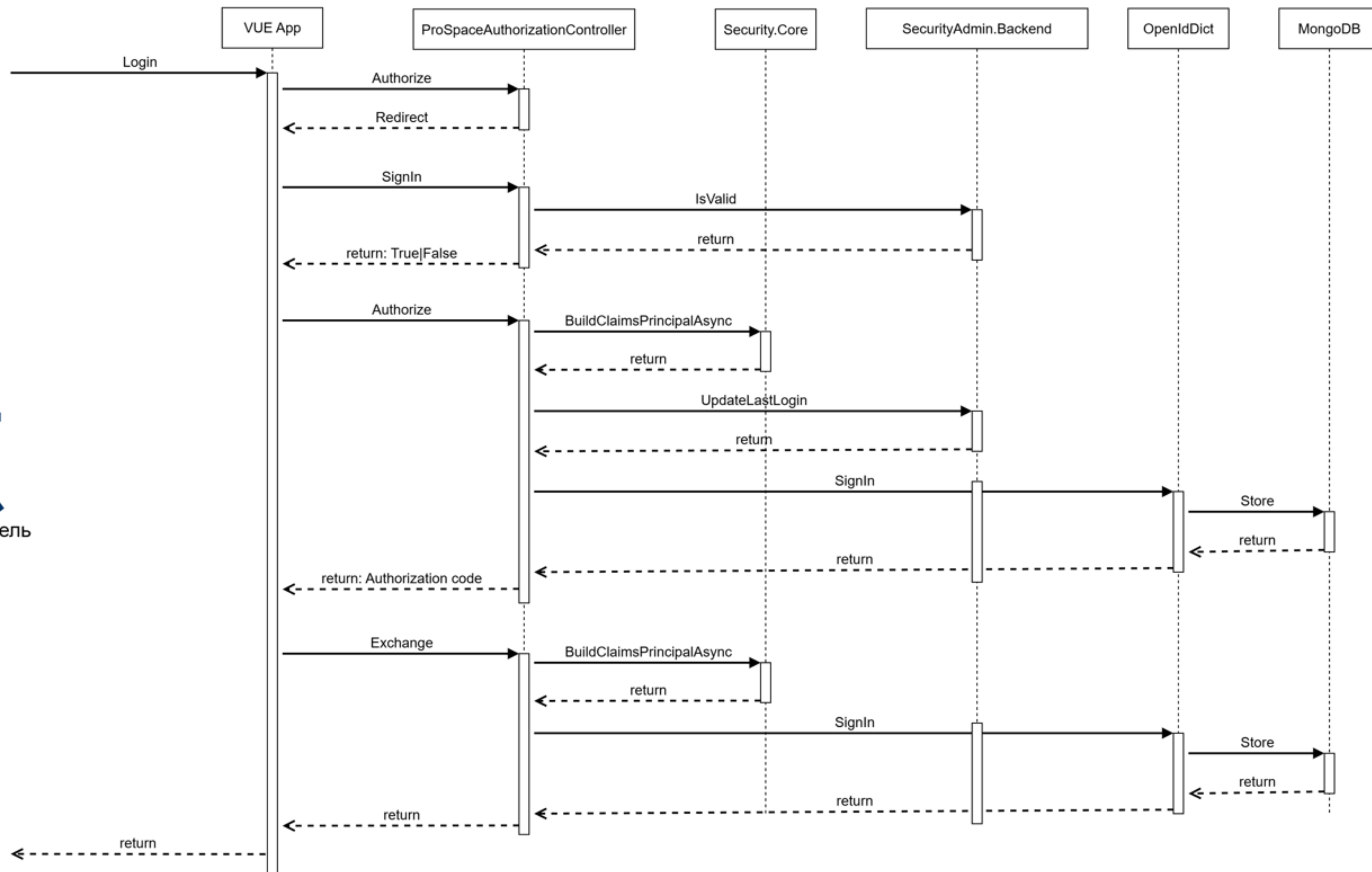
6.1 Подготовка ядра проекта

Ядро проекта (его базовые сервисы) разрабатывается и дорабатывается в отдельном репозитории. В результате компиляции получаются NuGet- и nrm-пакеты, которые в дальнейшем используются в основном проекте системы. Пакеты публикуются в Package Registry.



6.2 Сборка и выпуск основного проекта

Основной проект использует пакеты ядра в качестве основного функционала платформы; дополнительная бизнес-логика разрабатывается в виде отдельных сервисов. По итогам компиляции каждый сервис собирается в отдельный Docker-образ и публикуется в Container Registry. Релизный pipeline использует эти образы для установки в Kubernetes с помощью Helm-чартов.



6.3 Подготовка инфраструктуры

6.3.1 Исходное состояние

Процедура написана с допущением, что инфраструктура подготовлена и развёрнута в частном облаке заказчика на платформе Yandex Cloud, а именно:

- GitLab, GitLab Runner, GitLab Container Registry.
- Managed Service for Kubernetes.
- Yandex Object Storage.
- Managed Service for PostgreSQL.
- Managed Service for MongoDB.
- Две виртуальные машины (Monitoring и Vault).
- Созданная Application Group в ADFS заказчика.
- Настроены все необходимые сетевые подключения.
- Получены все необходимые артефакты инфраструктуры (сервисные аккаунты, SSL-сертификаты, DNS-имена).

6.3.2 Подготовка и конфигурирование элементов инфраструктуры

Запуском pipeline развёртывания в GitLab устанавливаются базовые компоненты, в частности выполняется установка системы мониторинга и OpenBao с последующей инициализацией.

Следующим шагом — также через запуск pipeline — выполняется настройка Kubernetes, OpenBao и компонентов мониторинга: загружаются необходимые инфраструктурные настройки для работы приложения.

FrontEnd UI

DataTable

Filter

Filter Items

NavigationPanel

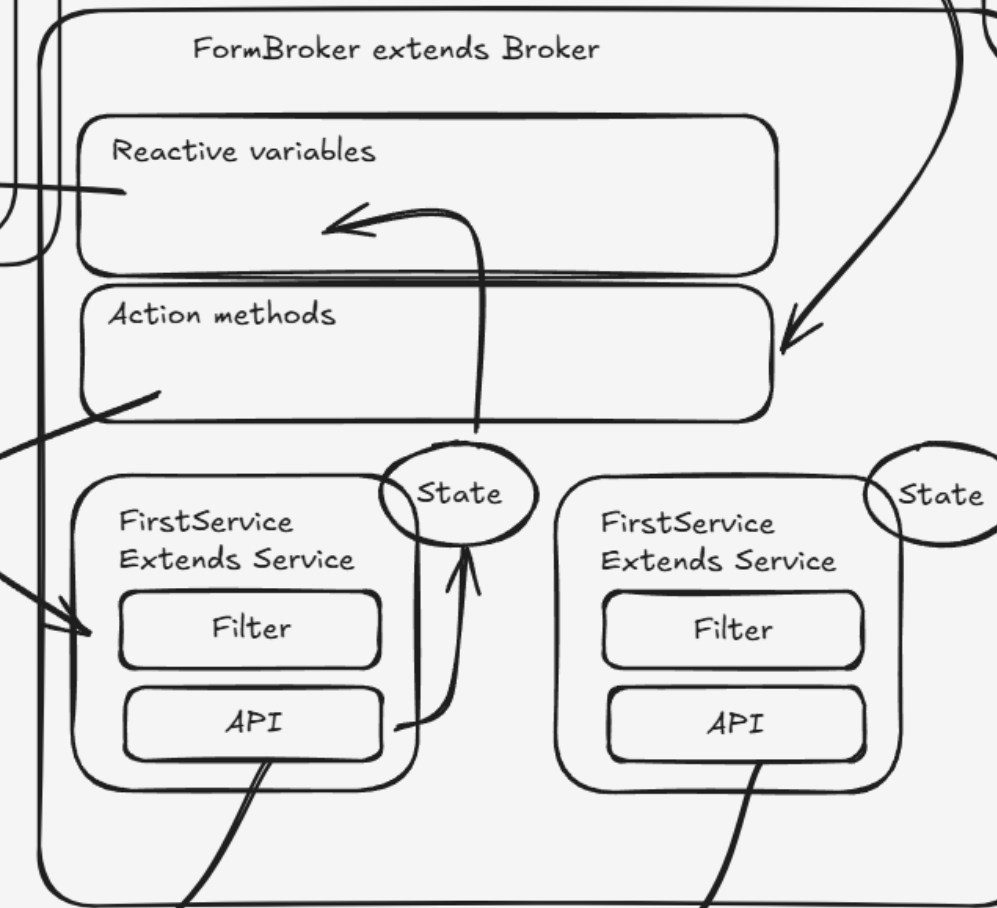
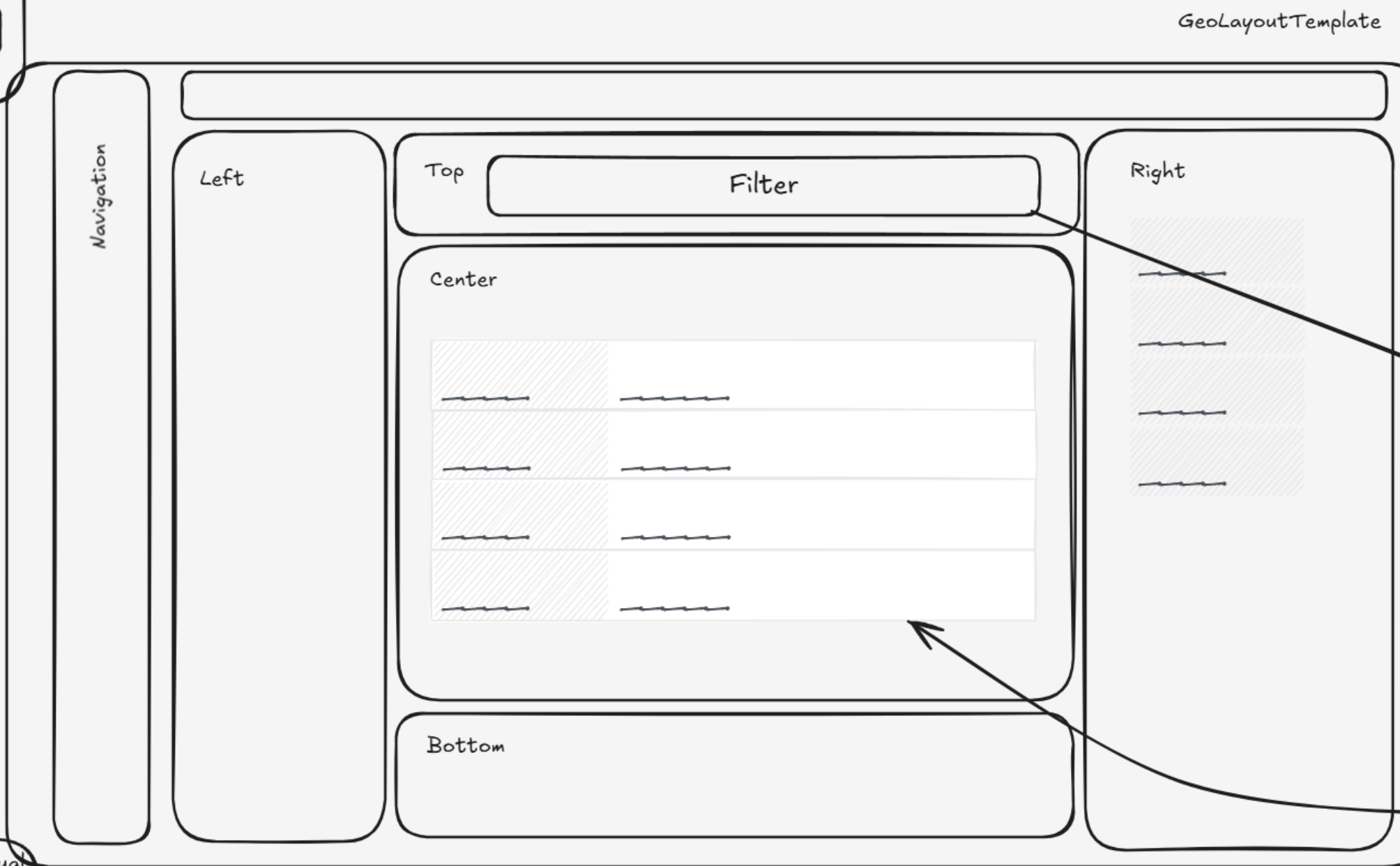
Context Menu

.....

FrontEnd NonVisual

Layouts

Form Builder



FrontEnd Core

API proxy

Auth

Service

Filter

Broker

Transformers

Gateway Endpoint

Лист регистрации изменений

[illegible]