

ПРОГРАММНЫЙ КОМПЛЕКС  
«ДАЛЬПЛАН»

**Описание архитектуры ПО**

ЦВАУ.01185-01 91

Листов 28

Инв.№ подл.	Подл. и дата	Бзм. инв. №	Инв.№ дубл.	Подл. и дата

2025

## СОДЕРЖАНИЕ

<b>Аннотация .....</b>	<b>3</b>
<b>1 Назначение программы .....</b>	<b>4</b>
1.1 Наименование, обозначение и назначение .....	4
1.2 Возможности.....	4
<b>2 Описание программного обеспечения .....</b>	<b>6</b>
2.1 Технологический стек.....	6
2.2 Программная и аппаратная совместимость.....	6
2.3 Сторонние зависимости.....	7
2.3.1 <i>Фронт</i> .....	7
2.3.2 <i>Обработка GeotIFF</i> .....	8
2.3.3 <i>Pipeline-gpu</i> .....	9
2.3.4 <i>Backend</i> .....	12
2.4 Средства и инструменты, используемые при разработке ПО .....	14
2.4.1 <i>Инструмент для хранения и управления репозиториями GitLab, билд-сервер для обеспечения непрерывной интеграции</i> .....	14
2.4.2 <i>Технические средства Guardant DL</i> .....	15
<b>3 Инфраструктура под размещение ПО .....</b>	<b>16</b>
<b>4 Компонентный состав .....</b>	<b>17</b>
4.1 Назначение компонентов.....	17
4.2 Frontend-сервисы .....	19
4.3 Сервис authapi .....	19
4.4 Сервис Integration-gate .....	19
4.5 Сервис tile_prepare.....	19
4.6 Сервис terracotta.....	20
4.7 Сервис geo-content-service .....	20
4.8 Сервис pipeline-gpu .....	22
4.9 Сервис notification-service .....	22
4.10 Схема взаимодействия сервисов.....	22
4.10.1 <i>Получение списка задачей</i> .....	22
4.10.2 <i>Создание задания</i> .....	23
4.10.3 <i>Получение и отображение результатов обработки</i> .....	23
4.10.4 <i>Изменение геопривязки снимка на карте</i> .....	24
4.10.5 <i>Взаимодействие с подложками карт</i> .....	24
<b>5 Перечень принятых сокращений .....</b>	<b>25</b>
<b>6 Термины и определения .....</b>	<b>26</b>

## АННОТАЦИЯ

Настоящий документ предназначен для ознакомления с функциями и принципами работы программного продукта «Программный комплекс «Дальплан» ЦВАУ.01185-01 (далее – программный комплекс «Дальплан», программный комплекс).

В документе изложены сведения о компонентах и сервисах, их назначении, взаимодействии и функциональных возможностях, а также информация о технологиях, используемых для разработки программного комплекса.

## 1 НАЗНАЧЕНИЕ ПРОГРАММЫ

### 1.1 Наименование, обозначение и назначение

Наименование программного продукта: «Программный комплекс «Дальплан».

Краткое наименование программного продукта: Программный комплекс «Дальплан».

Обозначение: ЦВАУ.01185-01.

Изготовитель программного продукта: ООО «ЦРТ».

Программный комплекс «Дальплан» предназначен для выполнения следующих функций:

- автоматический поиск и классификация объектов, анализ изображения (вторичная обработка) для выделения и распознавания объекта, определения его координат;
- ретроспективный анализ изменений на местности: выявление изменений в качественном и количественном аспекте за определенные периоды времени, которые зависят от тематики исследования.

### 1.2 Возможности

Программное обеспечение (далее – ПО) программного комплекса «Дальплан» обеспечивает следующие возможности:

- загрузка снимков в систему в ручном режиме;
- универсальный программный интерфейс для автоматического приема снимков от БПЛА;
- автоматическое обнаружение и классификация объектов;
- получение координат объектов;
- наложение снимка и объектов на картографическую подложку;

- ручная корректировка привязки снимков к геоподложке;
- проверка и редактирование результатов детекции объектов оператором;
- ввод информации оператором по результатам обработки снимков в ручном режиме;
- анализ снимка по зонам интереса;
- измерение линейных размеров объектов;
- нанесение дополнительных геометрических элементов на снимок и карту: линии, полигоны, точки;
- экспорт файлов с координатами объектов в ручном режиме.

Программный комплекс «Дальплан» имеет клиент-серверную архитектуру. В соответствии с указанной архитектурой практически все данные хранятся и обрабатываются на стороне сервера. Клиентские рабочие места взаимодействуют с сервером по сетям Intranet с использованием стандартного стека протоколов TCP/IP.

Для хранения информации используются реляционные и объектно-реляционные базы данных. Доступ к данным обеспечивается только авторизованным пользователям.

## 2 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 2.1 Технологический стек

Ниже перечислен набор технологий, являющихся основой для разработки ПО программного комплекса «Дальплан»:

C#, .NET 8.x, Python, React JS.

Архитектура: микросервисы, тонкий web-клиент.

Межсервисное взаимодействие: HTTPS REST API, RabbitMq, Terracotta.

Аутентификация пользователей и сервисов: JWT Bearer.

Конфигурирование: локально appsettings.json, docker-compose (env, yml).

Мониторинг и логирование: ELK.

Масштабирование: горизонтальное.

### 2.2 Программная и аппаратная совместимость

ПО программного комплекса «Дальплан» совместимо со следующим программным и аппаратным обеспечением:

Серверная операционная система: Astra Linux SE 1.7, 1.8, Debian 12, Ubuntu 24.04.

Поддержка аппаратного обеспечения архитектуры x86/x64.

СУБД: PostgreSQL версии 16.3.X.

Поддержка СХД: локальная FS, iSCSI, NFS.

Поддержка сред виртуализации: VMware/KVM.

Поддержка браузеров Google Chrome, Chromium.

Поддержка работы пользовательского веб-интерфейса на компьютерах с любой ОС.

## 2.3 Сторонние зависимости

Дополнительные библиотеки, требуемые для работы ПО программного комплекса «Дальплан», представлены ниже.

### 2.3.1 Фронт

Название	Описание
angular	Фреймворк для создания динамических и интерактивных веб-приложений на языках программирования TypeScript и JavaScript.
deck.gl	Фреймворк для визуализации данных на базе WebGL с открытым исходным кодом.
kurkle/color	Библиотека для быстрого разбора и манипуляции цветами CSS.
loaders.gl	Библиотека для визуализации данных, которая предоставляет модули загрузки (парсеры) для разных форматов.
nextgis/ol-map-adapter	Библиотека из набора NextGIS Frontend – набора браузерных JavaScript-библиотек для разработки собственных картографических веб-приложений на базе сервисов NextGIS.
ngrx	Библиотека управления состоянием для приложений Angular, которая реализует принцип работы Redux для Angular-приложений.
chart.js	Библиотека на JavaScript для визуализации данных, позволяющая создавать диаграммы для веб-страниц.
compare-versions	Библиотека для сравнения строк версий в формате «major.minor.patch».
core-js	Модульная стандартная библиотека для JavaScript, которая предоставляет полифиллы (добавляет недостающие функции) для новейших стандартов ECMAScript.
crypto-js	Библиотека JavaScript, которая предоставляет криптографические функции для веб-приложений.
date-fns	Библиотека для работы с датами в JavaScript. Она предоставляет набор функций для манипулирования, форматирования и сравнения дат.
file-saver	Библиотека для сохранения файлов в браузере, которая позволяет сохранять файлы разных типов (текстовые, изображения, PDF и др.) без использования сервера.
hex-to-rgba	Конвертер, который преобразует шестнадцатеричный (HEX) код цвета в значения RGBA (красный, зелёный, синий, альфа).
i	Библиотека для работы с текстовыми строками в Node.js, позволяющая изменять форму слов
jszip	Библиотека JavaScript для создания, чтения и редактирования ZIP-файлов прямо в браузере или в Node.js.
npm	Менеджер пакетов JavaScript.
ol	Библиотека на JavaScript, предназначенная для создания карт. Она позволяет создавать веб-интерфейсы для

<b>Название</b>	<b>Описание</b>
	отображения картографических данных, представленных в различных форматах и расположенных на различных серверах.
ol-ext	Набор расширений, элементов управления и взаимодействий для библиотеки OpenLayers (ol).
primeicons	Библиотека шрифтов для иконок, предназначенная для использования с библиотеками компонентов Prime UI и самостоятельными приложениями.
primeng	Библиотека компонентов пользовательского интерфейса для Angular с открытым исходным кодом.
proj4	Библиотека для преобразования координат из одной системы отсчёта в другую. Она включает как картографические проекции, так и геодезические преобразования.
ramda	Библиотека для функционального программирования в JavaScript.
rxjs	Библиотека для реактивного программирования в JavaScript с использованием наблюдаемых значений. Она облегчает создание асинхронного кода или кода, основанного на обратных вызовах.
simplify-ts	Пакет для упрощения 2D/3D полилиний в TypeScript, порт библиотеки Simplify.js для JavaScript. Это комбинация базового определения радиального расстояния и алгоритма Дугласа-Пекера.
tslib	Библиотека для работы с TypeScript, которая содержит вспомогательные функции, обычно используемые при компиляции кода на этом языке.
worker-plugin	Пакет, который автоматически объединяет и компилирует модули, загруженные в веб-работники (Web Workers), в рамках Webpack.
zone.js	Библиотека для JavaScript, которая обнаруживает и перехватывает асинхронные операции (обработку событий, таймеры, запросы на сервер и другие). Она создаёт контекст выполнения («зону») для этих операций, что помогает управлять ими в разных частях приложения.

### **2.3.2 Обработка GeoTIFF**

<b>Название</b>	<b>Описание</b>
pika	Клиентская библиотека RabbitMQ (AMQP 0-9-1) для Python.
boto3	Клиентская библиотека S3 для Python.
marshmallow	Независимая от ORM/ODM/фреймворка библиотека для преобразования сложных типов данных, таких как объекты, в собственные типы данных Python и обратно.
pillow	Библиотека обработки изображений на Python.
numpy	Базовый пакет для научных вычислений на Python.
shapely	Пакет Python с лицензией BSD для работы с плоскими геометрическими объектами и их анализа.

Название	Описание
sqlalchemy	Набор инструментов Python для работы с SQL и объектно-реляционное отображени.
terracotta	Библиотека для работы с данными тайного сервера Terracotta.
GDAL	Библиотека-транслятор с открытым исходным кодом и лицензией MIT для работы с растровыми и векторными форматами геопространственных данных.
pyproj	Библиотеки картографических проекций и преобразований координат.
fastapi	Веб-фреймворк для создания WEB API на Python.
uvicorn	Реализация веб-сервера ASGI для Python.
requests	Библиотека для выполнения HTTP-запросов в языке Python. Она позволяет отправлять GET, POST, PUT, DELETE и другие типы запросов с минимальным количеством кода.
lxml	Библиотека для обработки XML и HTML на языке Python.
pykml	Пакет Python для создания, анализа и обработки документов KML.
psycopg2	Адаптер базы данных PostgreSQL для языка программирования Python. Он позволяет приложениям на Python взаимодействовать с данными, хранящимися в базе PostgreSQL, и выполнять SQL-запросы.

### 2.3.3 Pipeline-gru

Название	Описание
boto3	Клиентская библиотека S3 для Python.
rasterio	Библиотека Python для чтения, записи и анализа геопространственных растровых данных. Она построена на основе библиотеки GDAL (Geospatial Data Abstraction Library) и предоставляет интерфейс для работы с наборами растровых данных, такими как спутниковые снимки, цифровые модели местности (DEMs) и другие данные с привязкой к сетке.
shapely	Пакет Python с лицензией BSD для работы с плоскими геометрическими объектами и их анализа.
affine	Пакет для создания матрицы аффинного преобразования плоскости в среде Anaconda.
pandas	Пакет Python, предоставляющий быстрые, гибкие и выразительные структуры данных, которые упрощают и делают интуитивно понятной работу с «реляционными» или «маркированными» данными.
tqdm	Библиотека для Python, которая позволяет добавлять индикаторы выполнения (прогресс-бары) к циклам и повторяющимся объектам.
botocore	Библиотека низкого уровня S3.
pydantic	Библиотека для Python, предназначенная для валидации и трансформации данных.

Название	Описание
gunicorn	Веб-сервер на языке Python, который использует стандарт WSGI (Web Server Gateway Interface) для запуска веб-приложений.
uvicorn	Реализация веб-сервера ASGI для Python.
geopandas	Инструменты Python для обработки географических данных
pyproj	Интерфейс Python для PROJ (библиотеки картографических проекций и преобразований координат).
pyogrio	библиотека для работы с векторными данными на Python, ориентированная на источники данных GDAL/OGR. Она предоставляет API для чтения и записи данных из источников, которые обычно имеют геометрию (точки, линии, полигоны) и связанные записи с данными в нескольких столбцах.
requests	Библиотека для выполнения HTTP-запросов в языке Python. Она позволяет отправлять GET, POST, PUT, DELETE и другие типы запросов с минимальным количеством кода.
onnx	Стандартный формат для представления моделей машинного обучения с открытым исходным кодом. Он позволяет разработчикам перемещать модели между различными фреймворками и платформами, обеспечивая совместимость.
onnxruntime-gpu	Версия пакета ONNX Runtime, предназначенная для работы в видеопамяти и на графическом процессоре (кроссплатформенный ускоритель машинного обучения для вывода и обучения).
pytest	Фреймворк для тестирования программного обеспечения на языке Python.
pytest-xdist	Плагин для pytest, обеспечивающий параллельное выполнение тестов на нескольких машинах.
fastkml	Библиотека для чтения, записи и манипулирования файлами формата KML (Keyhole Markup Language) в Python.
lxml	Библиотека для обработки XML и HTML на языке Python
python-dateutil	Сторонний модуль в Python, который расширяет функциональность встроенного модуля datetime. Он предоставляет инструменты для работы с датами и временем, включая анализ строк, обработку часовых поясов и выполнение арифметики с датами.
geopy	Библиотека Python для работы с геоданными. Она предоставляет собой интерфейс для выполнения задач, связанных с геокодированием (преобразование адресов в координаты) и геолокацией (определение координат по адресу или IP-адресу).
fastapi-utils	Веб-фреймворк для создания API на Python 3.8+.

Название	Описание
regex	Шаблоны, по которым можно проверять валидность данных, искать совпадения в тексте или заменять определённые фрагменты.
diffusers	Библиотека для современных предварительно обученных диффузионных моделей для создания изображений, аудио и даже трёхмерных структур молекул.
transformers	Библиотека с открытым исходным кодом, предоставляющая доступ к трансформерным моделям для обработки естественного языка (NLP). Библиотека поддерживает более 100 тысяч предобученных моделей, разработанных на архитектурах BERT, GPT, T5, RoBERTa, DistilBERT и многих других
scipy	Библиотека с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов. Она расширяет возможности NumPy, добавляя функции для работы с матрицами, статистическими данными, интегралами и оптимизацией.
gdal	Библиотека-транслятор с открытым исходным кодом и лицензией MIT для работы с растровыми и векторными форматами геопространственных данных.
Rtree	библиотека, которая реализует пространственный индекс R-Tree (R-Tree). Это обёртка библиотеки libspatialindex для Python, которая предоставляет функции для пространственной индексации.
torch	Пакет Python, предоставляющий две функции высокого уровня: - Вычисление тензоров (как в NumPy) с мощным ускорением на GPU. - Глубокие нейронные сети, построенные на основе системы автоградиента.
torchvision	Библиотека для задач компьютерного зрения, встроенная в экосистему PyTorch. Она предоставляет инструменты для работы с наборами данных, предобученными моделями и преобразованиями изображений.
torchaudio	Библиотека для работы с аудиоданными в экосистеме PyTorch. Она расширяет возможности PyTorch, предоставляя инструменты для обработки аудиосигналов.
opencv-python-headless	Пакет для библиотеки OpenCV (Open Source Computer Vision Library) в Python, предназначенный для серверных (безголовых) сред (например, Docker, облачные среды). Это «облегчённая» версия OpenCV, которая не содержит компонентов графического интерфейса пользователя (GUI).

### 2.3.4 Backend

Название	Описание
Newtonsoft.Json	Библиотека для работы с данными JSON в приложениях на языке C#. Она позволяет преобразовывать строки JSON в объекты C# и обратно, а также выполнять другие операции с JSON-данными.
Dapper	Фреймворк для объектно-реляционного сопоставления (ORM), который упрощает доступ к данным в приложениях .NET.
Microsoft.EntityFrameworkCore	Основной пакет Entity Framework Core (EF Core) — это объектно-ориентированная, легковесная и расширяемая технология для доступа к данным. Она позволяет работать с базами данных, абстрагируясь от самой базы и её таблиц, и работать с данными независимо от типа хранилища.
Npgsql.EntityFrameworkCore.PostgreSQL	Пакет, который является провайдером для работы с базой данных PostgreSQL в Entity Framework Core (EF Core). Он построен на основе Npgsql и позволяет взаимодействовать с PostgreSQL через объекты .NET, используя синтаксис LINQ для запросов.
Npgsql.EntityFrameworkCore.PostgreSQL.NetTopologySuite	Пакет NuGet, который поддерживает работу с пространственными данными (географическими и геометрическими) в Entity Framework Core для PostgreSQL с помощью библиотеки NetTopologySuite (NTS). Плагин для провайдера Npgsql EF Core, который интегрирует NTS с расширением PostGIS PostgreSQL, сопоставляя типы NTS с типами PostGIS и переводя пространственные операции в их эквиваленты SQL.
Microsoft.Extensions.Caching.Memory	Пакет NuGet, который предоставляет интерфейс для кэширования данных в памяти в приложениях на C#. Он входит в библиотеку Microsoft.Extensions.Caching, которая предназначена для реализации кэширования в приложениях .NET.
AutoMapper	Библиотека для автоматического сопоставления (маппинга) объектов в языке C#. Она упрощает преобразование данных между разными классами или объектами, заменяя повторяющийся код.
Dapper.SqlBuilder	Компонент библиотеки Dapper (микро-ORM для .NET). Он предназначен для динамического создания SQL-запросов. Класс SqlBuilder не является частью .NET Framework, но включён в библиотеку Dapper.
Elastic.Apm.NetCoreAll	Пакет для агента Elastic APM (Application Performance Monitoring) для приложений .NET Core. Пакет содержит расширение для ASP.NET Core, которое позволяет включить агент и активировать автоматический инструментарий с одной строкой кода.
Elastic.Clients.Elasticsearch	Клиент для Elasticsearch версии 8. Он обеспечивает доступ ко всем новым функциям Elasticsearch 8 и вносит в библиотеку ряд модернизаций.

<b>Название</b>	<b>Описание</b>
RabbitMQ.Client	Пакет для подключения к брокеру сообщений RabbitMQ.
Microsoft.AspNetCore.Authentication.JwtBearer	Компонент middleware для приложений ASP.NET Core, который поддерживает аутентификацию с помощью JSON Web Token (JWT). Он позволяет проверять токены, выпущенные сервером аутентификации, и обеспечивать безопасный доступ к ресурсам приложения.
Microsoft.AspNetCore.OpenApi	Пакет NuGet, который обеспечивает встроенную поддержку генерации документов OpenAPI в приложениях ASP.NET Core.
Microsoft.EntityFrameworkCore.Design	Пакет, необходимый для работы на этапе проектирования с инструментами командной строки или консоли диспетчера пакетов и является зависимостью от dotnet-ef и Microsoft.EntityFrameworkCore.Tools. Используется для управления миграциями и создания DbContext и типов сущностей путем обратного проектирования схемы базы данных.
Serilog	Библиотека для логирования в .NET-приложениях с открытым исходным кодом. Она позволяет разработчикам создавать структурированные логи, что упрощает их анализ и поиск информации.
Serilog.AspNetCore	Пакет для интеграции библиотеки Serilog с платформой ASP.NET Core. Он заменяет стандартный поставщик ведения журнала ASP.NET Core и позволяет записывать события в различные места назначения (sink).
Serilog.Enrichers.Environment	Пакет для библиотеки Serilog, который обогащает события логов информацией из среды выполнения. Это один из обогатителей (Enrichers) в Serilog, который добавляет дополнительные свойства к событиям логов.
Serilog.Extensions.Hosting	NuGet-пакет, который поддерживает логирование Serilog для Microsoft.Extensions.Hosting. Он предназначен для приложений, не использующих общий хост (например, консольных приложений).
Serilog.Sinks.Elasticsearch	Пакет для библиотеки логирования Serilog, который позволяет отправлять события в Elasticsearch - распределённую поисковую систему NoSQL.
Swashbuckle.AspNetCore	Инструмент для создания документации API, построенных на ASP.NET Core.
Swashbuckle.AspNetCore.Annotations	Пакет NuGet, который расширяет документацию API, сгенерированную с помощью Swagger (OpenAPI) в ASP.NET Core.
Swashbuckle.AspNetCore.Newtonsoft	Пакет NuGet, который поддерживает поведение сериализатора Newtonsoft.Json в генераторе Swagger (OpenAPI) для API, построенных с ASP.NET Core. По умолчанию Swashbuckle учитывает поведение сериализатора System.Text.Json (STJ), который поставляется с ASP.NET Core. Если используется сериализатор Newtonsoft, пакет позволяет настроить Swashbuckle, чтобы он учитывал его поведение вместо STJ.

Название	Описание
FluentValidation	Библиотека для .NET, которая позволяет описывать правила валидации запросов.
FluentValidation.AspNetCore	Пакет, который интегрирует библиотеку FluentValidation с ASP.NET Core. Он позволяет использовать FluentValidation для проверки входящих моделей в веб-приложениях ASP.NET Core.
Microsoft.AspNetCore.Mvc.NewtonsoftJson	Пакет NuGet, предназначенный для интеграции библиотеки Newtonsoft.Json в приложения ASP.NET Core MVC. Он позволяет использовать сериализацию и десериализацию JSON с помощью Newtonsoft.Json вместо встроенной библиотеки System.Text.Json.
Swashbuckle.AspNetCore.Filters	Пакет NuGet, который добавляет фильтры для улучшения документации веб-API с помощью Swagger (OpenAPI) в ASP.NET Core.
Microsoft.AspNetCore.Identity.EntityFrameworkCore	Пакет NuGet, который интегрирует систему ASP.NET Core Identity с Entity Framework Core. Он предоставляет реализацию для хранения данных, связанных с идентификацией, в базе данных с помощью EF Core.

Информация о расположении компонентов, номере используемой версии и типе лицензионного соглашения представлена в документе «Перечень используемых сторонних компонентов», ЦВАУ.01185-01 92.

## 2.4 Средства и инструменты, используемые при разработке ПО

В процессе разработки программного обеспечения ООО «ЦРТ» (далее – Правообладатель) использует программные средства и инструменты, перечисленные ниже.

### 2.4.1 Инструмент для хранения и управления репозиториями *GitLab*, билд-сервер для обеспечения непрерывной интеграции

Для хранения исходного текста и объектного кода Правообладателем используется система Gitlab (<https://about.gitlab.com/>).

Адрес расположения ПО (локально):

<https://it-gitlab.speechpro.com/geometry/>

Адрес расположения серверов:

Репищева ул., 20, лит. А, Санкт-Петербург, 197375 (Linx data center)

#### ***2.4.2 Технические средства Guardant DL***

Для активации, выпуска, управления и распространения лицензионными ключами программного обеспечения Правообладатель использует технические средства Guardant DL. Данные технические средства контролируются российскими организациями, что подтверждается письмами производителя ПО.

### 3 ИНФРАСТРУКТУРА ПОД РАЗМЕЩЕНИЕ ПО

Программный комплекс «Дальплан» размещается на сервере со следующей конфигурацией:

Наименование показателя	Значение
Процессор	32-поточный x86_64-совместимый
Оперативная память (RAM):	–
тип памяти	DDR5
объем памяти, Гб	128
Видеоускоритель:	–
объем памяти, Гб	24
тип памяти	GDDR6
разрядность шины памяти, бит	384
производительность, TFlops	80
Системный диск:	–
емкость, Тб	2
шина	pci-e 4
поддержка	nvme
скорость запись/чтение, Мб/с	7000/5000
ресурс записи, Тб	1200
Емкость дисков, Тб	8
Шина:	–
тип	SATA 6Gbps
скорость запись/чтение, Мб/с	560/530
ресурс записи, Тб	2800
Емкость дисков, Тб	8
Порты ввода/вывода	USB версии 3.0 или выше, HDMI версии 2.1a или выше, Ethernet 1Gbps или выше (RJ45)
Операционная система	Ubuntu 24.04, Debian 12.10

## 4 КОМПОНЕНТНЫЙ СОСТАВ

Программный комплекс «Дальплан» представляет собой набор микросервисов, взаимодействующих по HTTPS/REST API. Серверная часть отвечает за логику и пайплайны работы сервисов.

В качестве интерфейса пользователя используется тонкий web-клиент, при помощи которого можно работать со списком заданий, загружать снимки местности и просматривать результаты обработки.

Структурная схема программного комплекса «Дальплан» представлена на рисунке 1.

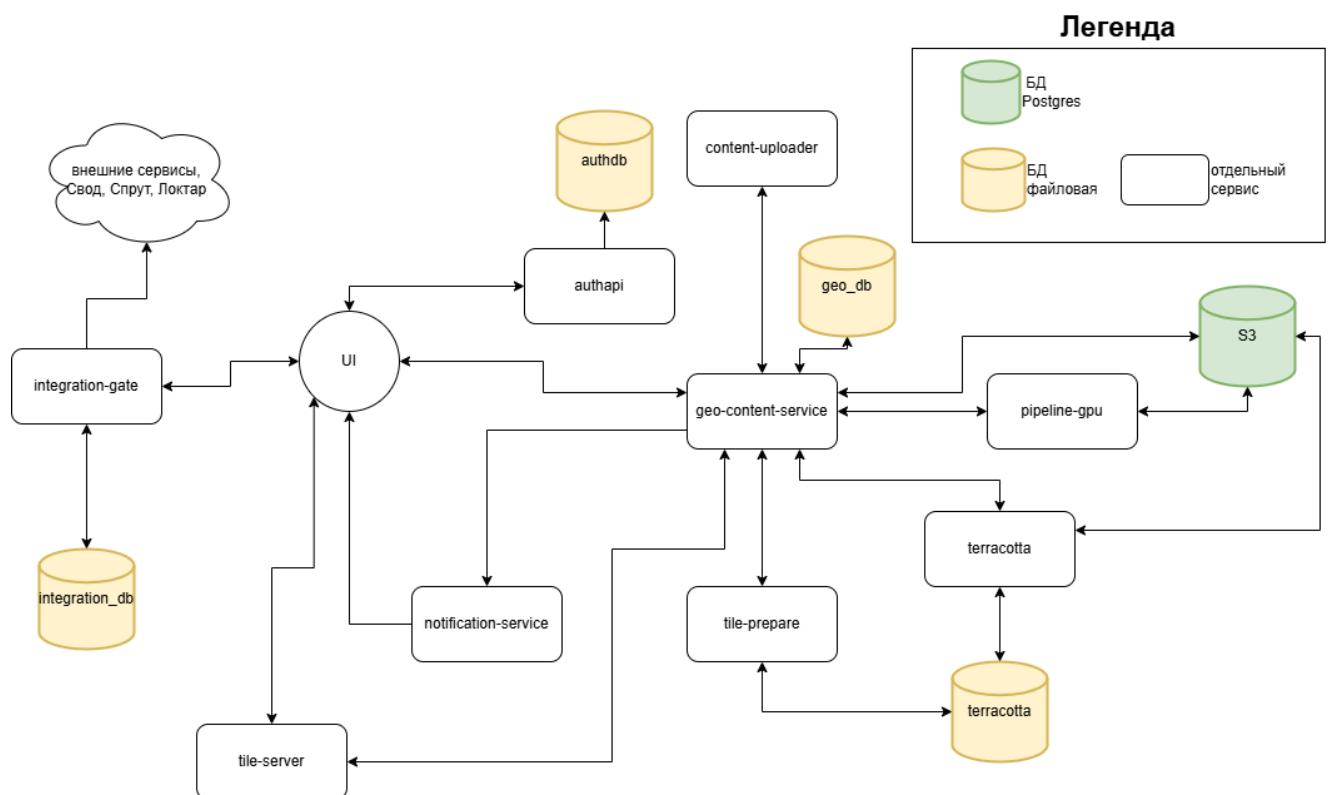


Рисунок 1 – Структурная схема программного комплекса «Дальплан»

### 4.1 Назначение компонентов

Назначение компонентов программного комплекса «Дальплан»:

Компонент	Тип	Описание
<b>authapi</b>	.NET	Сервис авторизации и аутентификации.
<b>users-service DB</b>	Postgres	База данных, которая отвечает за хранение данных о пользователях.

<b>Компонент</b>	<b>Тип</b>	<b>Описание</b>
<b>tile_prepare</b>	Python	<p>Сервис, который загружает снимки в Terracotta и извлекает метаданные из снимков GeoTIFF.</p> <p>Также изменяет геопривязку снимков на карте (georeferencing).</p>
<b>terracotta</b>	Python	Tiling-сервис, который дает возможность читать снимки "тайлами" и отображать GeoTIFF на карте в браузере.
<b>terracotta DB</b>	Postgres	База данных, которая отвечает за хранение снимков в GeoTIFF для отображения на карте.
<b>geo-content-service</b>	.NET	<p>Сервис обрабатывает входные данные, оркестрирует их обработку и предоставляет основное API для клиентского приложения</p> <p>Также через этот сервис происходит менеджмент взаимодействия с подложками карт (создание, редактирование, удаление, чтение подложек).</p>
<b>geo_db DB</b>	Postgres	<p>База данных, которая отвечает за хранение информации о:</p> <p>Instance-ах сервиса pipeline-gpu и моделях детекции, которые там используются;</p> <p>картах высот (<i>хранятся в отдельных таблицах, но ничего не мешает вынести их в отдельную БД</i>);</p> <p>хранит информацию об обработанных файлах и их связь с результирующими KML-файлами.</p>
<b>S3/ minio</b>	файловая БД, S3	Файловая база данных, которая отвечает за хранение снимков, загруженных пользователем, результатов их обработки, в том числе в формате KML.
<b>pipeline-gpu</b>	Python	<p>Сервис (AI Instance), который работает напрямую с видеокартой (GPU) на том железе, на котором он расположен.</p> <p>Осуществляет обработку каждого снимка последовательно.</p> <p>Через этот сервис также выставляются модели детекции, с помощью которых на GPU осуществляется обработка.</p>
<b>tile-server</b>	Node.js	<p>Сервис подложек карты, который держит открытые подключения к базам подложек, отрабатывает запросы в эти БД и отдает результаты.</p> <p>Для каждой подложки есть своя SQLite база в файловой системе, например, "Подложка N.mbtiles".</p>
<b>Подложка_N.mbtiles</b>	SQLite	<p>SQLite база в файловой системе, которая отвечает за хранение подложки карты.</p> <p>Подложка может быть в растровой базе (расширение .mbtiles) или векторной (расширение .pbf).</p>
<b>notification-service</b>	.NET	Пересыпает сообщения об изменении данных в сторону клиентского приложения

Компонент	Тип	Описание
<b>integration-gate</b>	.NET	Отправка результатов детекций во внешние сервисы
<b>content-uploader</b>	Python	Автоматическая загрузка снимков

## 4.2 Frontend-сервисы

Frontend-сервисы отвечают за пользовательский интерфейс программного комплекса «Дальплан».

Компонент	Описание
<b>users-service</b>	через <b>api-gateway</b> для получения информации о пользователе и дальнейшем ее использовании сервисом <b>tasks-service</b> . В самом <b>tasks-service</b> хранится только id пользователя.
<b>tasks-service</b>	через <b>api-gateway</b> для менеджмента заданий и получения результатов обработки задания.
<b>notification-service</b>	для получения уведомлений о статусе обработки пользователем.
<b>tile-server</b>	для загрузки и отображения подложек "тайлами" в формате .png или .jpeg. через <b>geo-content-service</b> для получения, добавления, обновления и удаления списка подложек карты.
<b>terracotta</b>	через <b>geo-content-service</b> для отображения обработанных результатов заданий на карте.
<b>tile_prepare</b>	через <b>geo-content-service</b> для изменения геопривязки снимка на карте и оптимизация загружаемых фото для представления в пользовательском интерфейсе

## 4.3 Сервис authapi

Задачи сервиса:

- занимается авторизацией и аутентификацией пользователей.

## 4.4 Сервис Integration-gate

Сервис хранит сведения о внешних системах, с которыми произведена интеграция, осуществляет отправку результатов детекций по запросу с фронта

## 4.5 Сервис tile\_prepare

Задачи сервиса:

– через RabbitMQ для каждого снимка (по id) получает запрос на обработку снимка от сервиса **geo-content-service**.

– скачивает снимки GeoTIFF из БД "S3/minio" и извлекает из них метаданные.

*Если из метаданных GeoTIFF нашли дату создания снимка, то отдааем ее в сервис **geo-content-service** через очередь "FileInfoUpdateEvent".*

– поканально загружает снимки GeoTIFF (раскладывает картинку на RGB-каналы) в сервис **terracotta** для последующего отображения на UI.

– через RabbitMQ (очередь "TileStatusEvent") информирует о статусе обработки снимка (статусе размещения снимка в сервисе **terracotta**). Этую очередь слушают сервисы **geo-content-service** и **geo-detection-tasks-processor**.

– по итогу вычисления масштаба снимка через RabbitMQ (очередь "ModelSelectionEvent") отправляет запрос на обработку снимка нейросетью в сервис **geo-content-service**.

– изменяет геопривязку снимков GeoTIFF на карте (georeferencing).

## 4.6 Сервис **terracotta**

Задачи сервиса:

– получает снимки GeoTIFF, разложенные на RGB-каналы, из сервиса **tile\_prepare** для последующего отображения на UI.

– дает возможность читать снимки "тайлами" и отображать GeoTIFF на карте в браузере.

## 4.7 Сервис **geo-content-service**

Задачи сервиса:

- по Rest-api принимает снимки, KML и CSV, сохраняет их в S3, запускает обработку

– через RabbitMQ (очередь "NewFileUploadEvent") для каждого снимка (по id) отправляет запрос на обработку в сервис **tile\_prepare**.

– через RabbitMQ (очередь "KmlStatusEvent") отправляет для каждого файла (по id) уведомления по о статусе обработки (KmlStatus) в сервис **geo-detection-tasks-processor**.

KmlStatus - статус процесса получения результирующего KML.

– через RabbitMQ (очередь "TileStatusEvent") получает информацию о статусе обработки снимка (TileStatus) сервисом **tile\_prepare** и записывает в свою БД.

TileStatus - статус размещения снимка в сервисе **terracotta**.

– если это возможно, то через RabbitMQ (очередь "FileInfoUpdateEvent") получает информацию о дате и времени создания снимка от сервиса **tile\_prepare**, записывает в свою БД в таблицу "fileinfo" и приписывает этот create\_date в очередь "KmlStatusEvent", когда публикует сообщение.

– стартует обработку снимка на **pipeline-gpu**, при этом передает конкретный id (имя файла в формате GUID).

– обрабатывает каждый файл отдельно.

– осуществляет периодический опрос (Polling /getStatus(id)) **pipeline-gpu** о статусе обработки снимка.

– передает запросы с UI на изменение гепривязки снимков в сервис **tile\_prepare**.

– валидирует KML и CSV для типов задания "GeoImportKml" и "GeoImportCsv".

– преобразовывает CSV в KML и сохраняет в БД "S3/minio".

– скачивает из БД "S3/minio" результаты обработки задания в формате KML.

– отдает результаты обработки задания в сервис **tasks-service** по методу Kml/download/{id} в формате KML.

– отдает расчет высоты объектов задания, если это возможно, по методу Heights/calculate в сервис **tasks-service**.

Сервис менеджмента файлов KML/CSV и снимков и результатов их обработки, то есть гео-контента.

#### 4.8 Сервис pipeline-gru

Задачи сервиса:

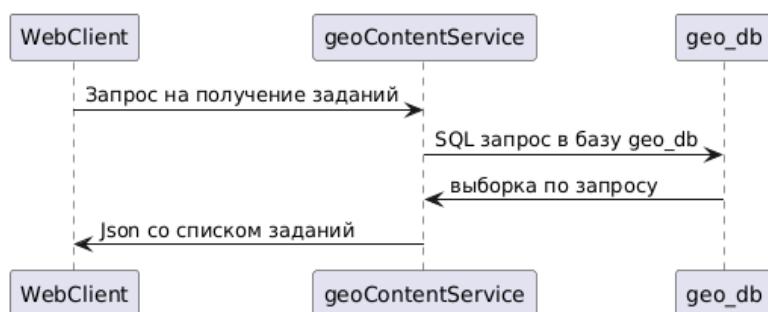
- скачивает снимки в формате GeoTIFF.
- занимается обработкой снимков (включая определение класса объекта).
- складывает результаты работы в БД "S3/minio" в виде файлов формата KML.

#### 4.9 Сервис notification-service

Сервис получает уведомления по шине RabbitMQ от других сервисов, трансформирует их в пользовательские уведомления и доставляет их до фронтенда по SignalR.

#### 4.10 Схема взаимодействия сервисов

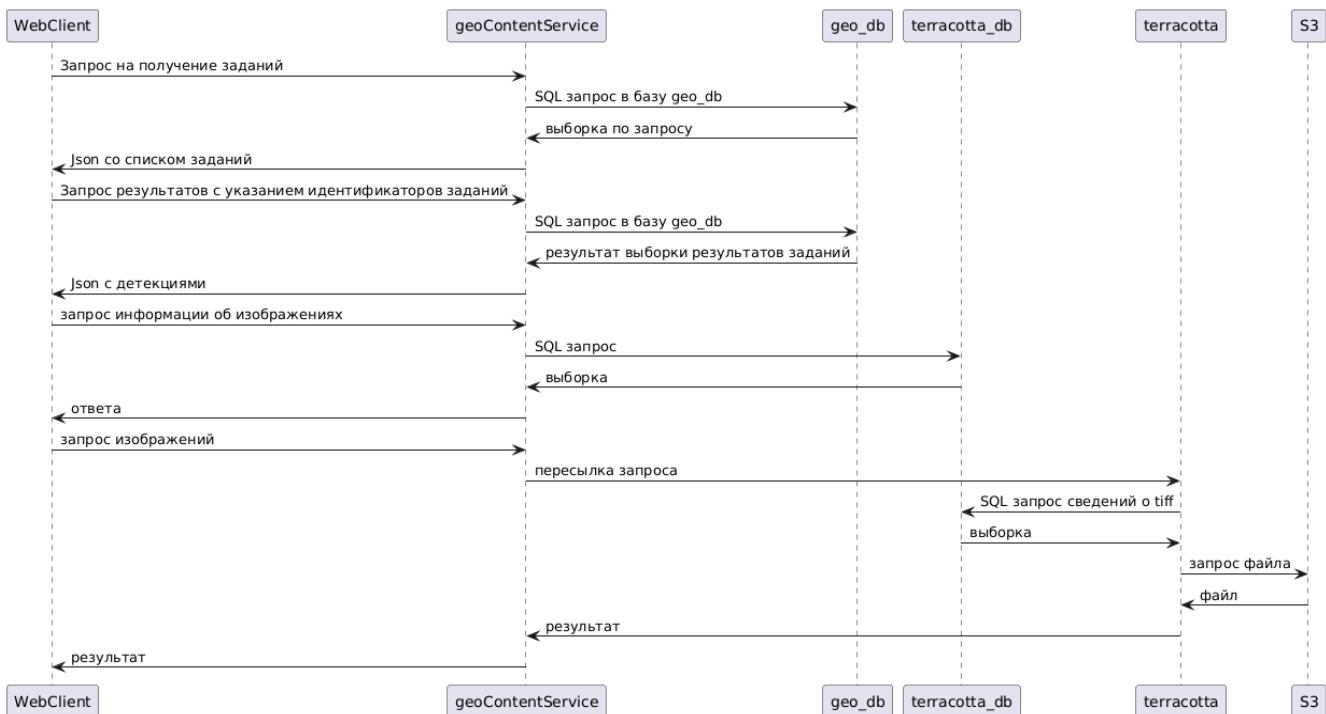
##### 4.10.1 Получение списка заданий



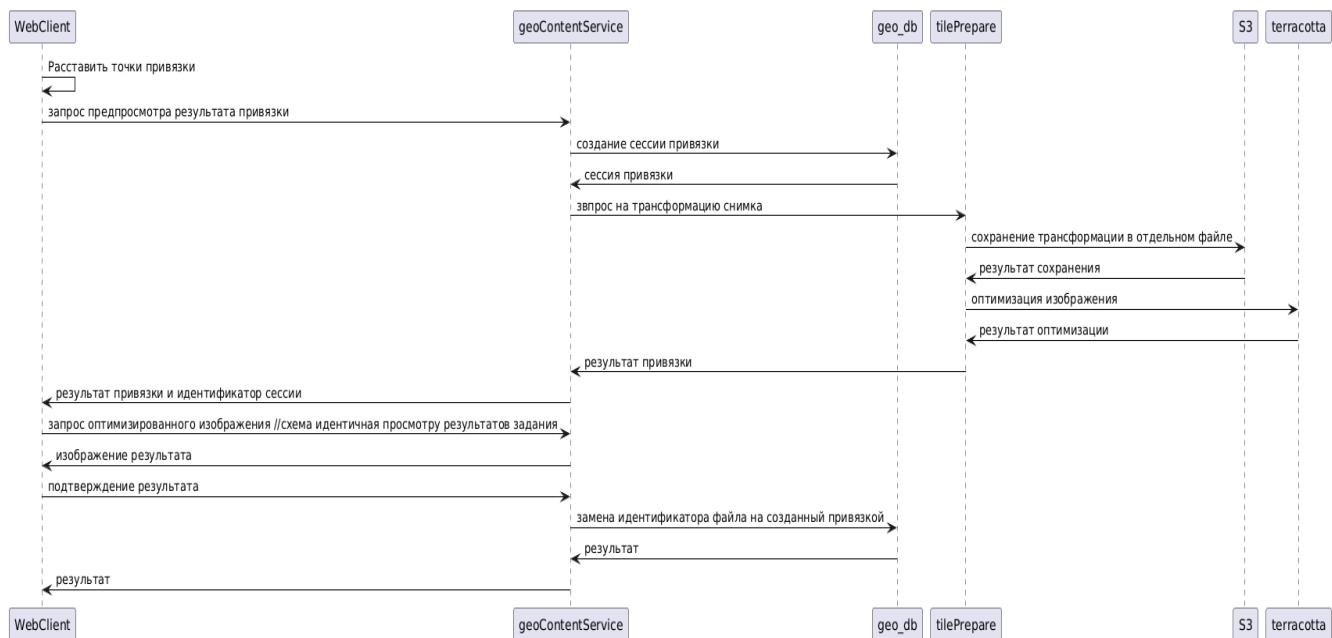
#### 4.10.2 Создание задания



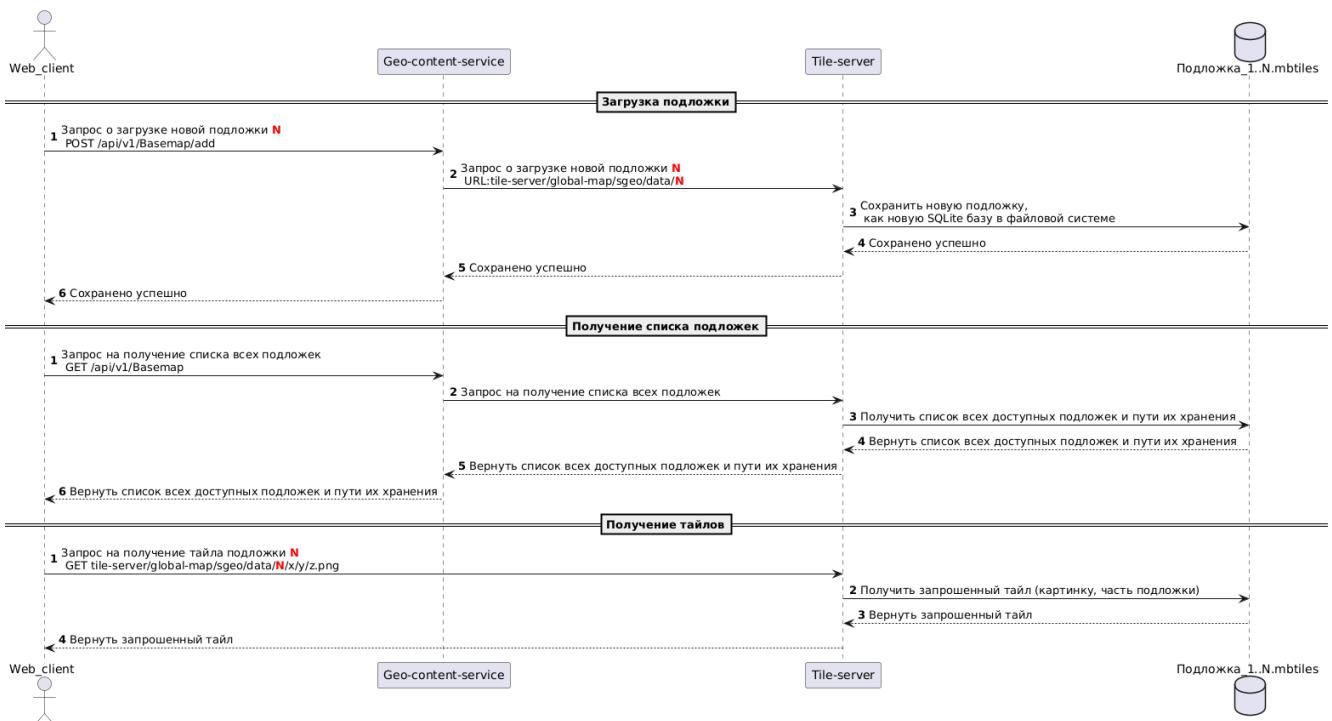
#### 4.10.3 Получение и отображение результатов обработки



#### 4.10.4 Изменение геопривязки снимка на карте



#### 4.10.5 Взаимодействие с подложками карт



## 5 ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

Сокращение	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
БПЛА	Беспилотный летательный аппарат
ПО	Программное обеспечение
ОС	Операционная система

## 6 ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

**Вектор** – графический элемент, наносимый на снимок системой или пользователем для обозначения объекта; вектор может быть различного типа:

точка,  
прямая,  
ломаная линия,  
квадрат,  
прямоугольник,  
полигон.

**Вероятность** – метрика, характеризующая уверенность системы в принятии решения о принадлежности объекта к одному из известных ей классов.

**Дерево объектов** – форма представления результатов детекции в UI системы.

**Детекция** – процедура автоматического поиска и классификации системой объектов на снимке.

**Задание** (таска/ джоба/ Jobs) – Задания на обработку и распознавание объектов на снимках, на внесение в систему KML-файлов с уже готовой информацией об объектах.

**Картографические данные** – любые данные с географической привязкой.

**Картографическая подложка** – конкретная реализация картографического слоя; примеры подложек:

Open street map,  
Google map.

**Картографический слой** – базовый слой визуализации на котором размещаются картографические данные.

**Класс** – группа объекта одного типа или вида, детектируемых системой; класс объекта и класс вектора тождественны.

**Модель** – модель машинного обучения, решающая задачи детекции и классификации объектов на снимке.

**Область интереса** – территория на снимке или карте, выделяемая пользователем для анализа распределения детектируемых объектов по классам.

**Объект** – условный знак, который отображается на карте; может как принадлежать реальному миру, так и быть виртуальной сущностью.

**Полигон** – графический элемент, принятый в системе по умолчанию для обозначения детектируемых объектов.

**Слой** – набор картографических данных, объединенных по типу источника и/или функциональному назначению; примеры слоев:

картографический слой,  
слой высот,  
слой векторов.

**Снимок** – фотографическое изображение на котором система производит поиск объектов.

**GeoTIFF** – Формат изображений с привязкой к местности (.tif, .tiff).

**GeoJson** – Формат для хранения географических структур данных, основанный на языке JavaScript Object Notation (JSON).

**GPU** (graphics processing unit) – Графический процессор. Это специализированный процессор, который работает с графикой, обрабатывает данные, связанные с отображением изображений, видео и 3D-графики.

**NATS** – Брокер сообщений.

**RabbitMQ** – Брокер сообщений.

**UI** (фронт) – Web-клиент.

## *Лист регистрации изменений*