

АКЦИОНЕРНОЕ ОБЩЕСТВО "ТЕРРА ТЕХ"

УТВЕРЖДАЮ
Генеральный директор
АО "Терра Тех"

М.Н. Болтачев
« ____ » _____ 2022 г.

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА
НЕЙРОСЕТЕВОЙ ОБРАБОТКИ И ИНТЕРПРЕТАЦИИ
КОСМИЧЕСКИХ СНИМКОВ «PIXEL.AI»**

ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

5177746378431.05.11.01.001 ПД-ЛУ

ЛИСТ УТВЕРЖДЕНИЯ

СОГЛАСОВАНО

Заместитель генерального
директора по производству

А. В. Беленов

2022

АКЦИОНЕРНОЕ ОБЩЕСТВО "ТЕРРА ТЕХ"

УТВЕРЖДЕН
5177746378431.05.11.01.001 ПД-ЛУ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА
НЕЙРОСЕТЕВОЙ ОБРАБОТКИ И ИНТЕРПРЕТАЦИИ
КОСМИЧЕСКИХ СНИМКОВ «PIXEL.AI»**

ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

5177746378431.05.11.01.001 ПД

Листов 39

2022

2
5177746378431.05.11.01.001 ПД
АННОТАЦИЯ

Настоящий документ содержит общее описание автоматизированной системы нейросетевой обработки и интерпретации космических снимков «Pixel.AI» 5177746378431.05.11.01.001 (далее – Система).

Документ содержит сведения о назначении автоматизированной системы, описание её структуры, перечисление технических и программных средств, необходимых для её эксплуатации, описание подсистем, сведения о входных и выходных данных.

Документ предназначен для специалистов технической поддержки и пользователей Системы.

Документ разработан в соответствии с требованиями следующих документов:

- ГОСТ Р 2.106 Единая система конструкторской документации. Текстовые документы;
- ГОСТ Р 2.105 Единая система конструкторской документации Общие требования к текстовым документам;
- ГОСТ 34.201-2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем;
- ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;
- ГОСТ Р 59795-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов».

СОДЕРЖАНИЕ

1.	Назначение системы.....	5
1.1.	Полное наименование системы, обозначение.....	5
1.2.	Вид деятельности, для автоматизации которой предназначена система.....	5
1.3.	Перечень объектов автоматизации.....	5
1.4.	Перечень функций.....	5
2.	Описание системы.....	7
2.1.	Структура системы и назначение составных частей	7
2.1.1.	Общие сведения.....	7
2.1.2.	Структура системы	7
2.1.3.	Назначение составных частей системы	8
2.2.	Сведения о системе и её составных частях, необходимые для обеспечения эксплуатации.....	9
2.2.1.	Общие сведения.....	9
2.2.2.	Требования к программно-аппаратному обеспечению.....	10
2.2.3.	Защита информации от несанкционированного доступа	11
2.3.	Описание функционирования системы и её составных частей	11
2.3.1.	Общие сведения.....	11
2.3.2.	Подсистема авторизации.....	12
2.3.3.	Подсистема загрузки данных.....	12
2.3.4.	Подсистема подбора данных ДЗЗ.....	14
2.3.5.	Подсистема обработки.....	16
2.3.6.	Подсистема аналитики.....	18
2.3.7.	Подсистема выгрузки данных	19
2.3.8.	Подсистема информационного взаимодействия	20
2.3.9.	Интегрированный интерфейс	20
2.3.10.	Подсистема осуществления оплаты	22
3.	Описание взаимосвязи системы с другими системами.....	23

3.1. Перечень АС, с которыми взаимодействует система	23
3.2. Описание взаимосвязей между системами	23
3.3. Описание информации взаимодействия	23
3.3.1. Входные данные	23
3.3.2. Выходные данные.....	23
4. Описание составных частей (подсистем)	24
4.1. Структура модулей и назначение их составных частей.....	24
4.2. Сведения о подсистемах и их составных частях, необходимые для обеспечения их функционирования.....	24
4.3. Описание функционирования модулей и их составных частей.....	24
Приложение 1 Описание запросов к API подсистем.....	25
Приложение 2 Описание структуры таблиц базы данных.....	34
Перечень терминов	37
Перечень сокращений.....	38

1. НАЗНАЧЕНИЕ СИСТЕМЫ

1.1. Полное наименование системы, обозначение

Полное наименование системы: «Автоматизированная система нейросетевой обработки и интерпретации космических снимков «Pixel.AI».

Краткое наименование системы: АС «Pixel.AI», Система.

Обозначение системы: 5177746378431.05.11.01.001

1.2. Вид деятельности, для автоматизации которой предназначена система

АС «Pixel.AI» предназначена для автоматического формирования аналитических данных о состоянии территории посредством последовательного пиксельного анализа и интерпретации космических снимков с различных группировок дистанционного зондирования Земли с использованием нейросетевых технологий.

Аналитические данные могут предоставляться в виде готовых продуктов или встраиваться в проектное решение потребителя любой отраслевой направленности.

1.3. Перечень объектов автоматизации

Объектом автоматизации является процесс формирования аналитических материалов на основе последовательного пиксельного анализа и интерпретации данных дистанционного зондирования Земли (далее – ДЗЗ, спутниковые снимки, снимки) и последующая интеграция в виде программного решения с ГИС-системами и проектными решениями Заказчика.

1.4. Перечень функций

Система обеспечивает выполнение следующих функций:

- а) идентификация и аутентификация пользователей Системы;
- б) автоматический поиск и загрузка спутниковых снимков из внешних и локальных источников;
- в) автоматический подбор данных ДЗЗ согласно с выбранным Пользователем алгоритмом;

- г) автоматическая обработка спутниковых снимков в части улучшения яркостных характеристик;
- д) автоматический пиксельный анализ спутниковых снимков (попиксельная классификация и сегментация снимка);
- е) автоматическая верификация данных об объектах и территориях на основе данных пиксельного анализа и их обогащения кадастровой и другой необходимой информацией;
- ж) предоставление доступа к аналитическим данным как в файловом виде, так и с использованием интегрированного интерфейса, в том числе и предназначенного для взаимодействия с геоинформационными системами;
- з) расчет и проведение оплаты, формирование платежных документов, учёт выполненных платёжных операций с привязкой информации к Пользователю.

2. ОПИСАНИЕ СИСТЕМЫ

2.1. Структура системы и назначение составных частей

2.1.1. Общие сведения

Система «Pixel.AI» обеспечивает автоматический подбор спутниковых снимков с различных группировок дистанционного зондирования Земли, дальнейшее автоматическое формирование аналитических данных о состоянии территории посредством их последовательного пиксельного анализа и интерпретации.

Схема основных процессов Системы отображена на рис. 1.

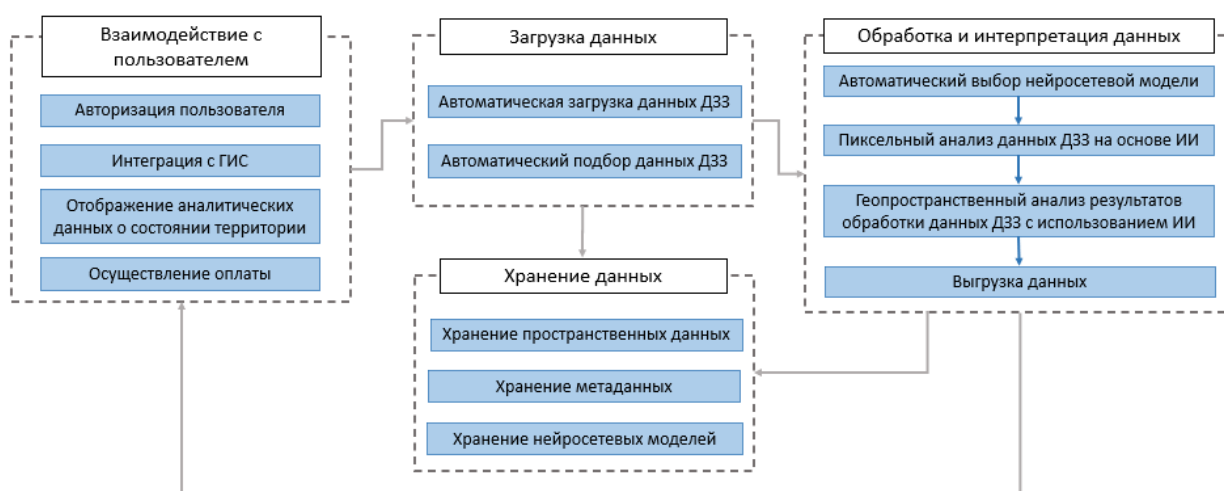


Рисунок 1 – Основные процессы Системы

2.1.2. Структура системы

Система включает следующие составные части – подсистемы:

- а) подсистема авторизации;
- б) подсистема загрузки данных;
- в) подсистема подбора данных ДЗЗ;
- г) подсистема обработки;
- д) подсистема аналитики;
- е) подсистема выгрузки данных;
- ж) подсистема информационного взаимодействия;

з) подсистема осуществления оплаты.

Архитектура Системы представлена на рис. 2.

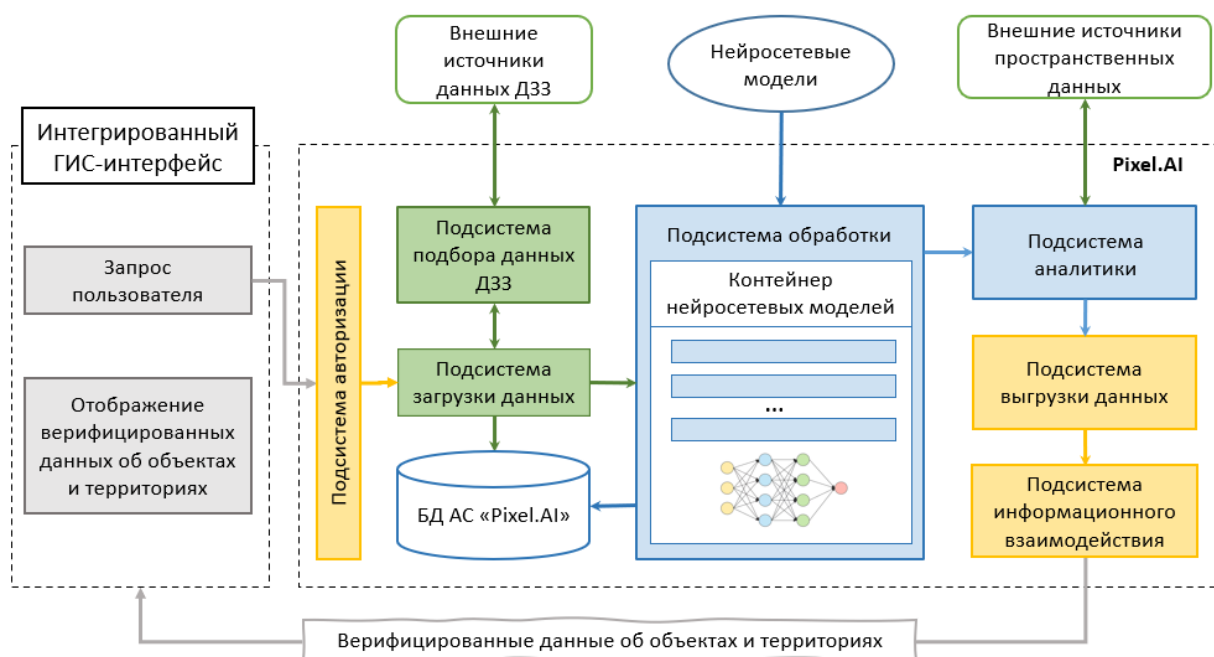


Рисунок 2 – Архитектура Системы

Взаимодействие между компонентами системы осуществляется с использованием интерфейсов прикладного программирования Representational state transfer application programming interface (REST API). Система реализована по модульной схеме.

2.1.3. Назначение составных частей системы

Подсистема авторизации обеспечивает идентификацию и аутентификацию пользователей системы, проверку уровня их полномочий (пользователь, администратор) и отображение доступных Пользователю данных.

Подсистема загрузки данных обеспечивает загрузку сведений об объектах, данных ДЗЗ на территорию объектов.

Подсистема подбора данных ДЗЗ обеспечивает автоматический поиск снимков территории объектов как во внешних источниках, так и в локальном каталоге. Подсистема оперирует данными, получаемыми через API подсистемы загрузки данных.

Подсистема обработки обеспечивает обработку спутниковых снимков, полученных от подсистем загрузки и подбора данных, с автоматическим выбором нейросетевой модели, согласно с поставленной пользователем задачей.

Подсистема аналитики выполняет верификацию объектов в соответствии с исходной задачей, оперируя данными из подсистемы обработки. Подсистема выполняет как ряд функций пространственного анализа, такие как подсчет площадей, определение пересечений контуров объектов, так и такие операции, как склеивание векторных или растровых данных и обогащение данных кадастровой и другой необходимой информацией.

Подсистема выгрузки данных обеспечивает выгрузку аналитических данных об объектах в файловом виде, а также отображает результаты обработки и анализа в интегрированном интерфейсе (при его использовании).

Подсистема информационного взаимодействия предназначена для обеспечения интеграции Системы и других (внешних) информационных систем для отображения результатов обработки, интерпретации и анализа данных ДЗЗ и прочей пространственной информации, передачи данных по API, а также для обеспечения доступа пользователей к информации о состоянии территории.

Подсистема осуществления оплаты обеспечивает расчёт оплаты, формирование платежных документов, производит операцию оплаты и учёт выполненных платёжных операций с привязкой информации к пользователю.

2.2. Сведения о системе и её составных частях, необходимые для обеспечения эксплуатации

2.2.1. Общие сведения

Система базируется на модульной архитектуре построения информационных систем. Серверная часть реализована на языках программирования Python и Go, клиентская – на связке HTML5/JavaScript.

Работа Системы выполняется в инфраструктуре Docker – программного обеспечения (далее – ПО) для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

Каждая подсистема Системы запускается в отдельной изолированной среде (контейнере). Для одновременного управления несколькими контейнерами используется технология Docker Compose, позволяющей управлять сложными приложениями.

Взаимодействие между подсистемами выполняется с использованием интерфейсов прикладного программирования (REST API) и при помощи очереди сообщений, реализованной с использованием программного брокера сообщений RabbitMQ. Описание запросов к API различных подсистем АС Pixel.AI приведено в приложении 1.

В качестве СУБД используется PostgreSQL. Каждая подсистема АС Pixel.AI обеспечивает накопление данных в выделенном наборе таблиц. Обобщённая структура хранения данных, которые используются в работе Системы, представлена на рис. 3. Допускается использование разных схем данных для каждой подсистемы. Описание структуры таблиц базы данных приведено в приложении 2.

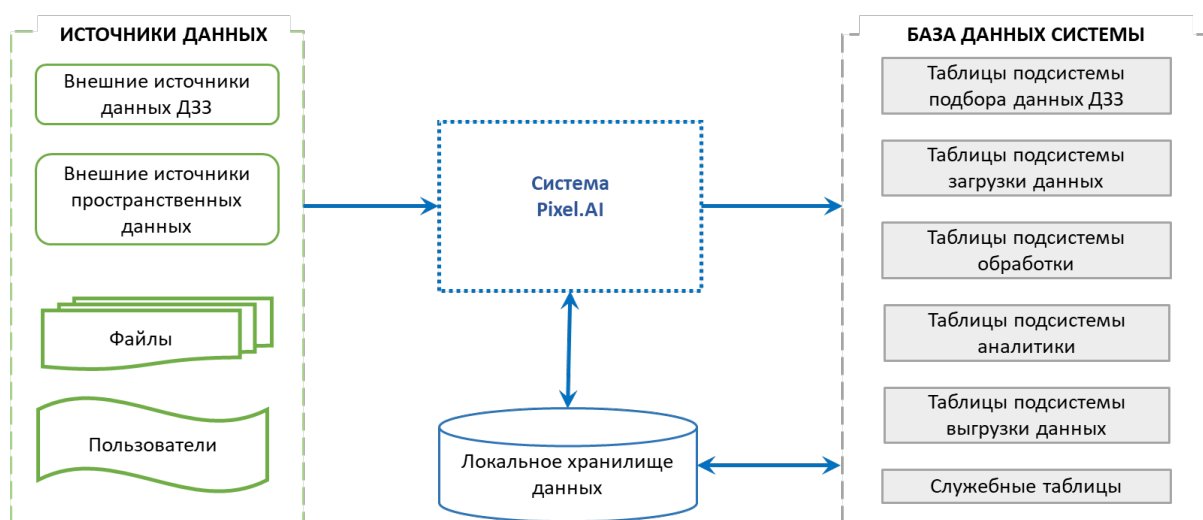


Рисунок 3 – Структура хранения данных

2.2.2. Требования к программно-аппаратному обеспечению

Параметры аппаратного обеспечения серверов должны соответствовать или превосходить по характеристикам следующие требования:

- процессор 2 × Intel Gold 6240 (18x2.6 ГГц HT);

- память 256 ГБ – 8×32 ГБ DDR4 ECC Reg;
- диск 2×1600 ГБ SSD SATA Enterprise, 6×10 ТБ HDD SATA Enterprise;
- материнская плата X11DPi-NT;
- корпус 2U, 1PSU;
- сеть 100Гбит/с.

При работе пользователей с Системой должны использоваться следующие веб-браузеры:

- Microsoft Edge (версия не ранее 80);
- Mozilla Firefox (версия не ранее 73);
- Google Chrome (версия не ранее 80);
- Apple Safari (версия не ранее 14);
- Opera (версия не ранее 67);
- Apple iOS Safari (версия не ранее 14.0);
- Google Chrome Android (версия не ранее 80.0).

2.2.3. Защита информации от несанкционированного доступа

Для обеспечения возможности безопасной аутентификации и повышения скорости работы канала связи между пользователем и сервером системы используется Wildcard SSL сертификат (SSL сертификат домена 2-го уровня и всех его поддоменов 3-го уровня) для домена terratech.ru.

Сертификат установлен на сервере Системы.

При установке формируется запрет на использование небезопасных протоколов и настраивается шифрование по протоколам TLS 1.0/1.1/1.2. Для всех запросов, не использующих шифрование (HTTP), конфигурируются перенаправления, использующие шифрование (HTTPS).

2.3. Описание функционирования системы и её составных частей

2.3.1. Общие сведения

Система может использоваться в следующих вариантах:

а) автономно – в виде установленного на компьютер или ноутбук ПО. Пользователь использует для работы интегрированный интерфейс Системы;

б) интегрированно – в составе сторонних программных комплексов и информационных систем. Система встраивается в работу сторонних систем и устанавливается без интерфейса для обеспечения автоматической работы со спутниковыми снимками. Взаимодействие со сторонними системами (вход и выход данных) выполняется без участия пользователя – через API.

При работе Системы формируется запрос, содержащий данные о выборе объекта или территории, критерии подбора данных ДЗЗ и тип задачи для формирования аналитических данных. Далее согласно этому запросу на вход подаются исходные данные и запускается пиксельный анализ и аналитика пространственных данных.

Описание работы составных частей Системы приведено ниже.

2.3.2. Подсистема авторизации

Подсистема авторизации обеспечивает идентификацию и аутентификацию пользователей системы.

Авторизованные пользователи уровня специалист или руководитель имеют возможность просмотра и получения результатов работы Системы, а также формирования запросов на создание новых наборов аналитических данных. Авторизованные пользователи уровня администратор имеют доступ к изменению настроек Системы, подключению внешних нейросетевых моделей и источников данных, управлению составом и описанием наборов аналитических данных.

2.3.3. Подсистема загрузки данных

Подсистема осуществляет хранение загруженных данных на дисковом пространстве сервера и обеспечивает формирование и передачу запроса посредством API в подсистему подбора снимков. Подсистема загрузки данных позволяет проводить поиск по хранилищу данных с учетом следующих

параметров: источник, из которого был загружен снимок, область интереса, дата, максимальная облачность. Для этого при загрузке в хранилище снимки проходят автоматическую индексацию. Снимки в хранилище разделены уникальной комбинацией следующих параметров: источник, спутник, миссия, уровень обработки, формат и состав файлов каталога.

Архитектура подсистемы представлена на рис. 4.

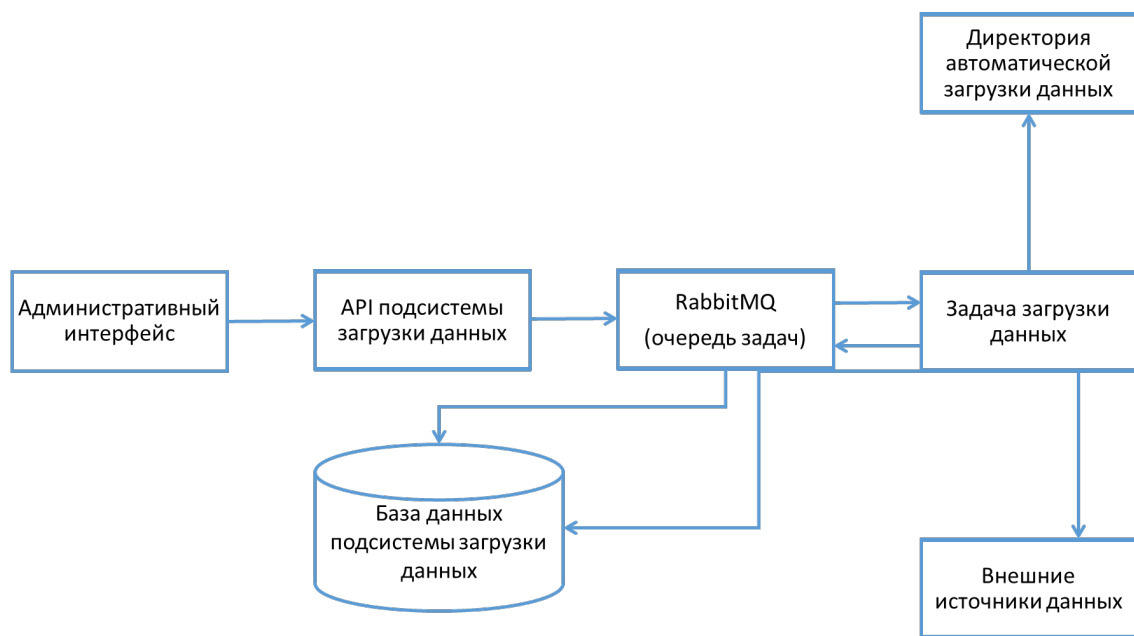


Рисунок 4 – Архитектура подсистемы загрузки данных

Подсистема поддерживает загрузку данных ДЗЗ следующих космических аппаратов (далее – КА) с соответствующими уровнями обработки:

- Landsat 8 Collection 1 Level 1;
- Landsat 8 Collection 2 Level 1;
- Landsat 8 Collection 2 Level 2;
- Sentinel-2 MSIL2A
- Kanopus MS L2 (16 bit, RBGN);
- Kanopus PMS L2 (16 bit, RBGN).

Загрузка снимков, полученных в результате работы вышеуказанных космических аппаратов, осуществляется из двух источников: Copernicus Open Access Hub и USGS EarthExplorer. Ввиду технических ограничений указанных

сервисов возможна одновременная загрузка только четырех снимков, поэтому в подсистеме была реализована очередь загрузки.

Снимки из источника Copernicus Open Access Hub, которые были сделаны ранее, чем за два месяца от даты поиска или загрузки, могут находиться в архиве, в результате чего им будет необходима активация, занимающая до 24 ч. Для этих целей в подсистеме реализован алгоритм отложенного скачивания (загрузки) архивных снимков, который выполняет загрузку снимка из Copernicus Open Access Hub в локальное хранилище после его активации.

Подсистема создает и хранит превью снимков и их RGB варианты.

Для каждой задачи формируется каталог, содержащий векторные данные и превью растровых данных. Векторные данные доступны для скачивания пользователю Системы через интегрированный интерфейс или через API.

2.3.4. Подсистема подбора данных ДЗЗ

В базовом состоянии системы поддерживается работа со следующими спутниковыми данными с зарубежных и отечественных КА:

- Landsat 8 Collection 1 Level 1;
- Landsat 8 Collection 2 Level 1;
- Landsat 8 Collection 2 Level 2;
- Sentinel-2 MSI L2A;
- Kanopus MS L2 (16 bit, RBGN);
- Kanopus PMS L2 (16 bit, RBGN).

Также реализована возможность подключения новых источников данных ДЗЗ.

Архитектура подсистемы подбора данных ДЗЗ представлена на рис. 5.

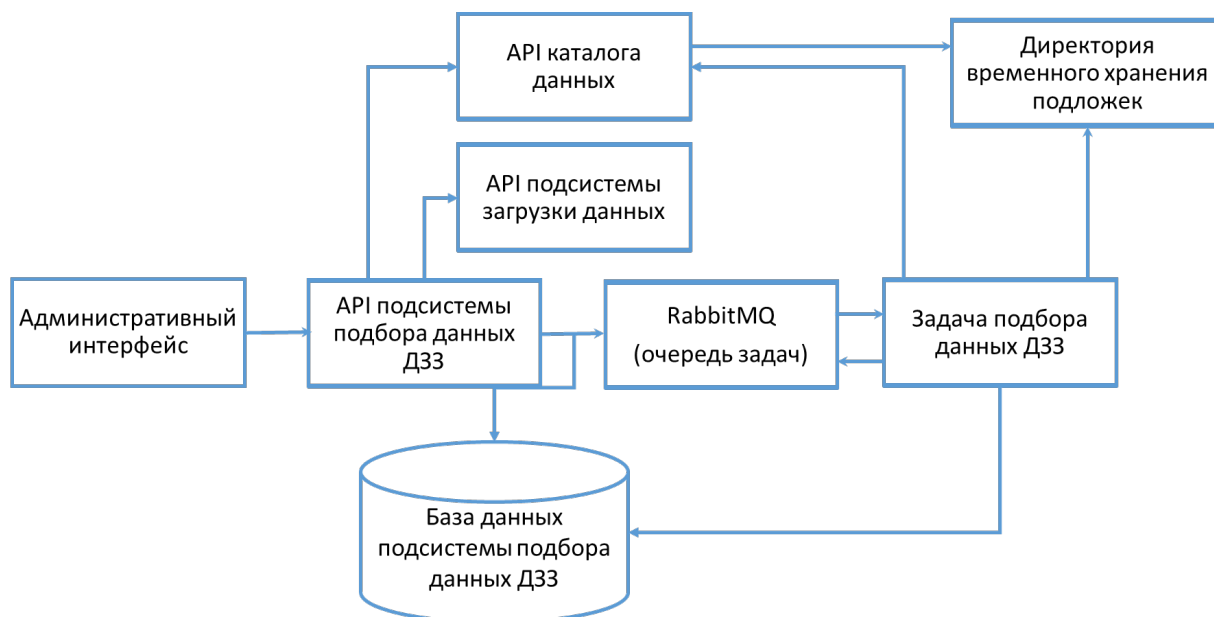


Рисунок 5 – Архитектура подсистемы подбора данных ДЗЗ

Поиск подсистемой подбора данных ДЗЗ осуществляется как в режиме стандартном, с представлением пользователю выборки снимков заданной территории за определенный временной промежуток, так и с использованием алгоритмов, настроенных на задачи формирования полного покрытия или нескольких временных срезов съемки территории.

В подсистеме реализованы следующие алгоритмы поиска:

а) простой поиск. Выполняется поиск снимков в заданном временном промежутке. В параметрах запроса от подсистемы загрузки данных выбирается список источников, диапазон дат и максимально допустимая облачность;

б) полное покрытие. Выполняется поиск минимальной комбинации снимков с наименьшей облачностью за заданный диапазон дат, полностью покрывающей область интереса. В параметрах запроса от подсистемы загрузки данных предусмотрен выбор одного из двух вариантов (сообщение об ошибке или выдача максимального количества доступных снимков для выделенной области) выдачи результатов, если не представляется возможным найти снимки, полностью покрывающие область интереса;

в) поиск срезов. Выполняется поиск, разбивающий переданный в параметрах запроса временной промежуток на заданное количество срезов. В

каждом из срезов алгоритм пытается найти полное покрытие снимками области интереса. Если найти полное покрытие в каждом из срезов не удаётся, алгоритм уменьшает количество срезов до тех пор, пока не будут найдены срезы с полным покрытием.

2.3.5. Подсистема обработки

Подсистема обработки выполняет обработку спутниковых снимков после их подбора и загрузки в локальное хранилище, далее формируется задача на обработку снимка, которая передаётся в подсистему обработки с использованием API.

Архитектура подсистемы обработки представлена на рис. 6.

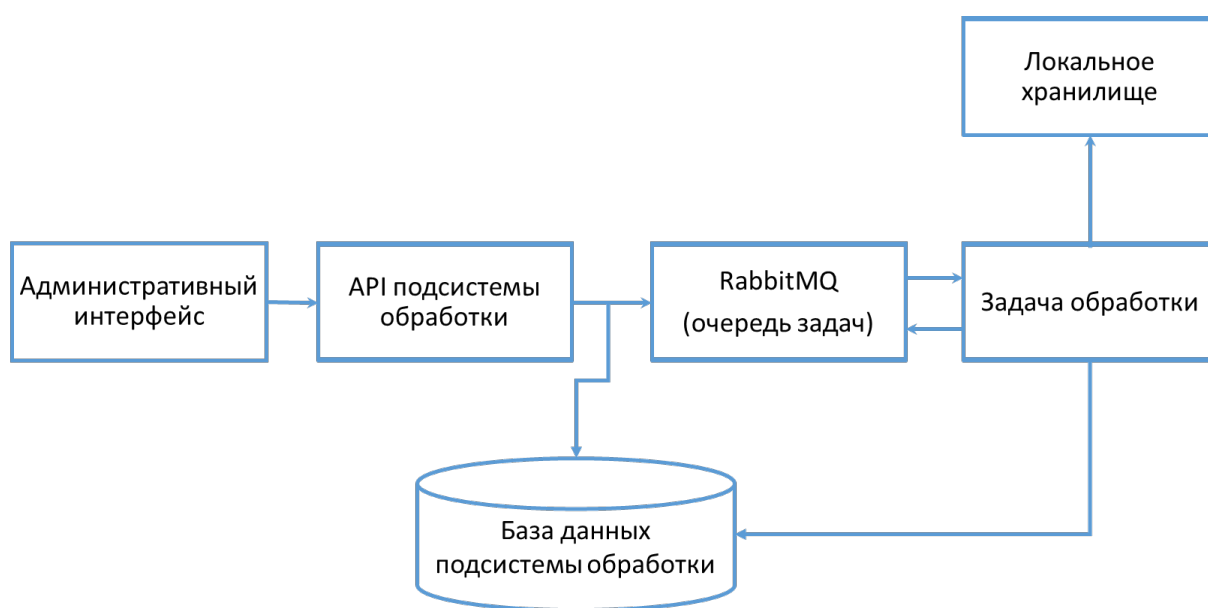


Рисунок 6 – Архитектура подсистемы обработки

Содержащийся в подсистеме контейнер нейросетевых моделей может быть пополнен при дальнейшей эксплуатации Системы. Подсистема обработки автоматически выполняет обработку снимков в части улучшения их яркостных характеристик и попиксельную классификацию и сегментацию снимка. Данные действия выполняются алгоритмами на основе свёрточных нейронных сетей. Результат работы подсистемы обработки в виде снимка с улучшенными характеристиками и векторных данных, содержащие контуры объектов и

прочую информацию, полученную в ходе пиксельного анализа, направляется в подсистему аналитики.

Подсистема обработки получает сведения о задаче по обработке снимка и ссылки на необходимый набор исходных данных. Далее выполняется автоматический подбор алгоритмов решения поставленной задачи.

Для обработки снимков предусмотрены следующие операции:

- обработка снимка в части улучшения яркостных характеристик (если это предусмотрено задачей);
- автоматический последовательный пиксельный анализ снимка (попиксельная классификация и сегментация снимка) для задач выявления объектов, извлечения границ объектов, определения местонахождения объектов на местности, сравнения объектов и других задач.

Для выполнения вышеперечисленных операций Система использует ранее загруженные нейросетевые модели. Кроме этого, Система обеспечивает возможность подключения новых нейросетевых моделей для расширения возможностей обработки данных. Для этого предусмотрено наличие контейнера нейросетевых моделей.

Все нейросетевые модели, загружаемые в Систему, проходят процесс подготовки. На первом этапе работы составляется список кодов сегментации, определяющих классы сегментации, также выделяется дополнительный несегментируемый класс (фон). Коды сегментации присваиваются как классам поверхности, объектам на ней, так и облакам, теням, дымке и разнообразным дефектам изображения.

Далее производится подбор космических снимков, которые разделяются на обучающую, валидационную и тестовую выборки. Снимки, относящиеся к обучающей выборке, проходят процедуру разметки, в результате которой формируются растровые маски, совпадающие по ширине и высоте со снимками. Каждый пиксел в снимке принимает значение одного из кодов сегментации.

На следующем этапе исследуется решение задачи сегментации нейросетевыми алгоритмами. Архитектура нейронной сети выбирается из известных архитектур, а также проводится модификация сверточных слоев. Выполняется выбор функции потерь, таких, как кроссэнтропия, Jaccard loss, Dice loss, Focal loss, и др., в т. ч. их линейные комбинации. Применяются аугментации при обучении, такие, как поворот изображения, зеркальное отражение, изменение яркости в каналах, масштабирование изображения. Выполняется выбор алгоритма оптимизации весов нейронной сети, выбор величины скорости обучения и политики ее изменения, выбор политики досрочного завершения обучения. Определяется размер патча входного/выходного изображения, выбор размера батча, подбор прочих гиперпараметров нейронной сети.

После выбора и программирования архитектуры проводится обучение нейронной сети на обучающей выборке с контролем величины функции потерь и метрики на валидационной выборке. В процессе выбираются те веса, которые обеспечивают максимальное значение функции метрики на валидационной выборке. Обучение производится на языке программирования Python с использованием фреймворков машинного обучения, таких, как TensorFlow, Keras, PyTorch. После обучения проводится контроль результатов на тестовой выборке.

После разработки, оптимизации и обучения сеть встраивается в ПО Системы, производится настройка и отладка работы.

Учитывая технические особенности работы Системы и серверных мощностей, принимается, что подсистема обработки одновременно выполняет до четырёх задач на обработку снимков. Для последовательного выполнения поступающих задач в подсистеме обработки реализована система очередей.

После обработки осуществляется автоматическое сохранение результатов обработки в локальном хранилище и их передача в виде ссылок в подсистему аналитики с использованием API.

2.3.6. Подсистема аналитики

Подсистема аналитики обеспечивает выполнение следующих операций:

- подсчет площадей;
- сбор\поиск\анализ информации о объекте;
- верификация наличия объекта на снимке;
- верификация состояния объекта;
- определение пересечений;
- определение расположения объекта на территории;
- анализ территории, на которой расположен объект (категория земель, состояние земель и пр.);
- определение соотношения используемой и неиспользуемой части территории.

Верификация наличия объекта – это процесс подтверждения с использованием мультиспектральных, синтезированных среднего и высокого разрешения космических снимков наличия указанных в задаче строений и земельных участков в рамках обозначенной территории (очертаний кадастрового участка, пользовательского контура, адреса). Типы объектов: отдельно стоящие здания жилой и коммерческой недвижимости, земельные участки различного назначения.

Подсистема аналитики работает с результатами, полученными от подсистемы обработки, в том числе проводит склеивание векторных или растровых данных и обогащение данных кадастровой и другой необходимой информацией.

Верифицированные данные об объектах анализа передаются в подсистему выгрузки данных.

2.3.7. Подсистема выгрузки данных

Подсистема выгрузки данных обеспечивает выгрузку верифицированных данных об объектах в файловом виде с использованием API. Подсистема также передаёт результаты обработки и анализа данных для отображения в интегрированный интерфейс (при его использовании).

2.3.8. Подсистема информационного взаимодействия

Подсистема информационного взаимодействия обеспечивает интеграцию Системы с геоинформационными системами для отображения результатов обработки и интерпретации космических снимков, а также с другими (сторонними) системами в части обмена данными. Интеграция со сторонними программными комплексами и геоинформационными системами проходит посредством API. Система встраивается в работу сторонних систем и устанавливается без интегрированного интерфейса для обеспечения автоматической работы со спутниковыми снимками и получения данных их автоматизированного анализа.

2.3.9. Интегрированный интерфейс

Интегрированный интерфейс обеспечивает графическое представление верифицированных данных об объектах и другой сопроводительной информации (метаданные по объектам, текстовая информация, дополняющие материалы) (рис. 7).

Интегрированный интерфейс работает в виде одностраничного веб-приложения (Single Page Application, SPA).

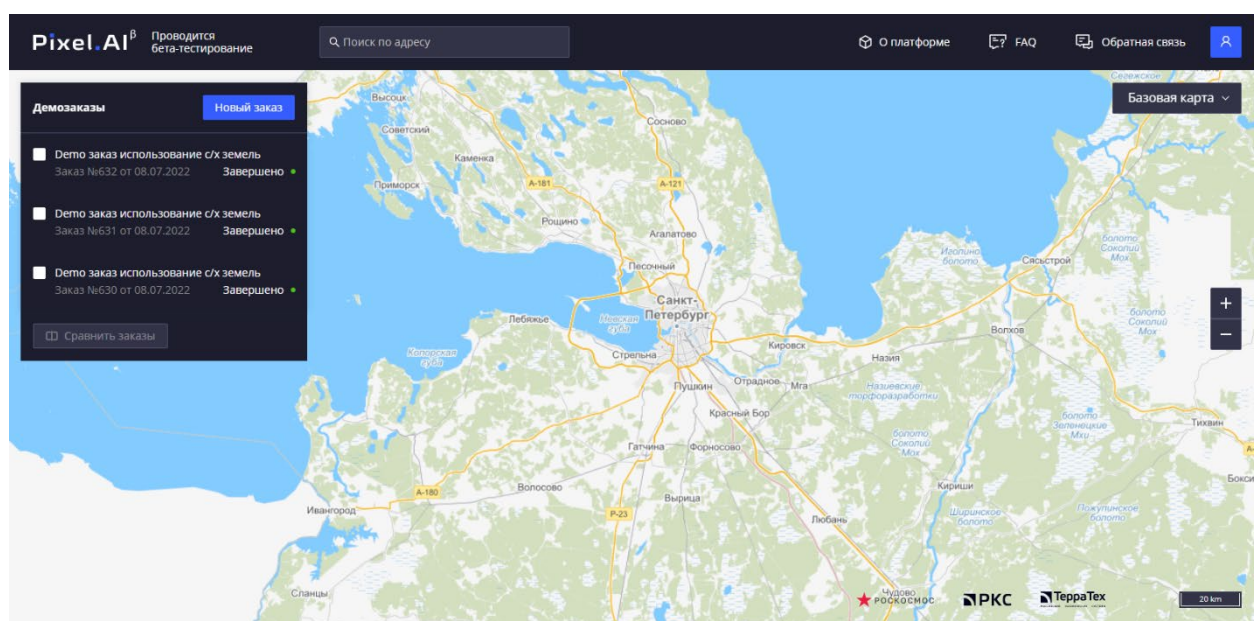


Рисунок 7 – Интегрированный интерфейс Системы

Интегрированный интерфейс является надстройкой Системы и может использоваться опционально, в зависимости от задачи.

Интегрированный интерфейс не перегружен графическими элементами и обеспечивает быстрое отображение экранных форм. Навигационные элементы выполнены в удобной для пользователя форме. Ввод/вывод данных, приём управляющих команд и отображение результатов их исполнения выполняются в интерактивном режиме. Интегрированный интерфейс соответствует современным эргономическим требованиям и обеспечивает удобный доступ к основным функциям и операциям Системы.

Интегрированный интерфейс рассчитан на преимущественное использование манипулятора типа «мышь», т. е. управление в интерфейсе выполняется с помощью набора экранных меню, кнопок, значков и др. элементов управления.

Все надписи экранных форм, а также сообщения, выдаваемые пользователю (кроме системных сообщений), выполнены на русском языке.

Для облегчения навигации по разделам интерфейса и считывания информации предусмотрено дополнительное цветовое и/или шрифтовое дифференцирование.

Интегрированный интерфейс обеспечивает доступ пользователей ко всем функциям Системы и содержит следующие функциональные компоненты (рис. 8):

- карта (экран работы с картографическими данными и данными ДЗЗ с привязкой к местности);
- панель инструментов;
- панель слоёв.

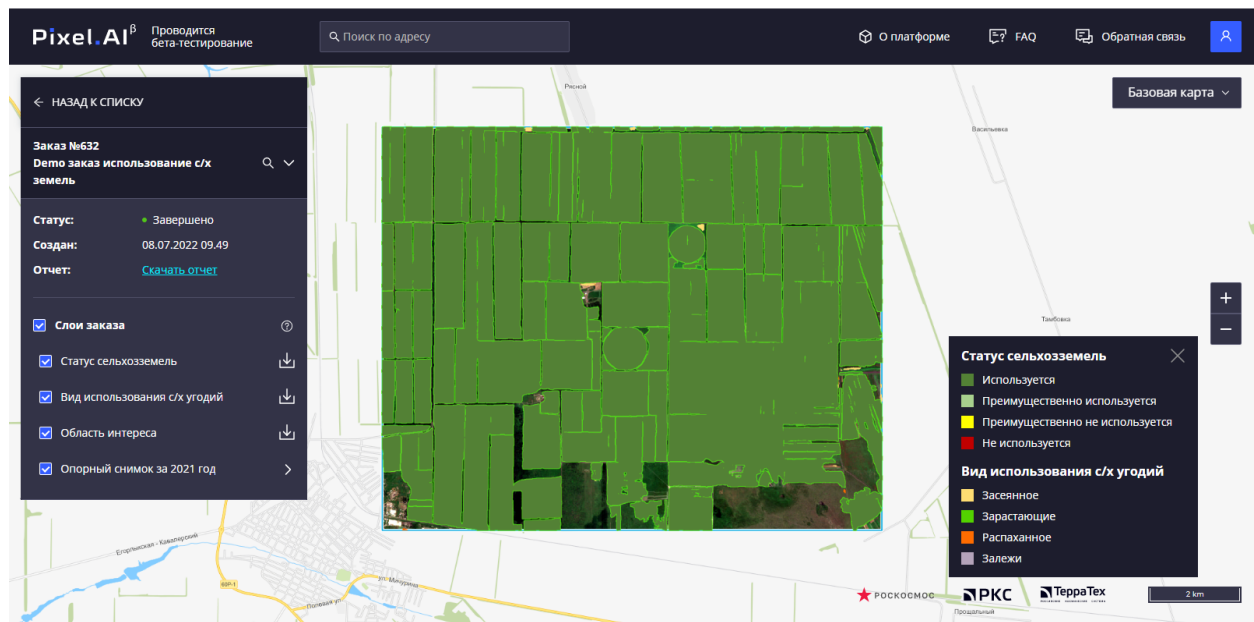


Рисунок 8 – Пример отображения стилизованных картографических данных

Для отображения на мобильных устройствах (планшеты, смартфоны) реализована адаптивная версия интегрированного интерфейса. Для мобильных устройств возможны ограничения по функциональности (программно-аппаратные ограничения платформ).

2.3.10. Подсистема осуществления оплаты

Подсистема осуществления оплаты обеспечивает расчёт оплаты, формирование платежных документов, операцию оплаты, учёт выполненных платёжных операций с привязкой информации к пользователю.

3. ОПИСАНИЕ ВЗАИМОСВЯЗИ СИСТЕМЫ С ДРУГИМИ СИСТЕМАМИ

3.1. Перечень АС, с которыми взаимодействует система

Система является частью группы сервисов АО «Терра Тех» по работе с данными ДЗЗ, в которую также входит облачное решение TerraCloud по продаже данных ДЗЗ. Предусмотрена единая система авторизации.

3.2. Описание взаимосвязей между системами

Интеграция с внешними системами и передача данных осуществляется посредством API.

В API Системы реализованы функции передачи исходных данных, визуализация данных, изменения моделей, разграничения доступа к данным и т. д.

Описание API и параметров его работы приведено в документации.

API предоставляется Заказчику по запросу.

3.3. Описание информации взаимодействия

3.3.1. Входные данные

Входными данными для Системы являются следующие сведения:

- контур объекта;
- координаты объекта;
- предмет верификации.

3.3.2. Выходные данные

Выходными данными для Системы являются верифицированные с помощью попиксельного анализа снимков данные об объектах и сопутствующие материалы.

4. ОПИСАНИЕ СОСТАВНЫХ ЧАСТЕЙ (ПОДСИСТЕМ)

4.1. Структура модулей и назначение их составных частей

Сведения о структуре модулей и назначении их составных частей приведены в 2.1 настоящего документа.

4.2. Сведения о подсистемах и их составных частях, необходимые для обеспечения их функционирования

Сведения, необходимые для обеспечения функционирования модулей и их составных частей, приведены в 2.2 настоящего документа.

4.3. Описание функционирования модулей и их составных частей

Сведения о функционировании модулей и их составных частей приведены в 2.3 настоящего документа.

ОПИСАНИЕ ЗАПРОСОВ К API ПОДСИСТЕМ

Таблица 1 – API запрос на создание новой задачи

Тип запроса	POST
URL	/api/requests/new/
Запрос	application/json
	<pre>{ // Тип и параметры задачи "order_type": null, "order_type_params": {}, // Пользовательские файлы с их интерпретацией "files": [{ "file_id": 1, "interpretation": "aoi" }, { "file_id": 2, "interpretation": "fields_contour" }], // Опциональное название задачи "title": "Осень 2021" }</pre>
Ответ	application/json 200 OK
	<pre>{ "aoi": null, "created_at": "2022-03-03T18:43:45.029147", "files": [{ "container_id": null, "created_at": "2022-03-15T17:16:35.871157", "file_id": 1, "interpretation": "aoi", "request_id": "57396f490f5948f6948eddd5d390abf3"</pre>

	<pre> }, { "container_id": null, "created_at": "2022-03-15T17:16:35.871262", "file_id": 2, "interpretation": "fields_contour", "request_id": "57396f490f5948f6948eddd5d390abf3" }], "order_id": null, "order_type": "agro_usages_analysis", "order_type_params": { "n_slices_per_veg_year": 1, "searching_limit": 1, "year_from": "2021", "year_to": "2021" }, "performer": null, "project": null, "request_id": "f9a0cf0e-db02-46a8-ba68-d41b959c69d2", // Статусы: new, canceled, payments_pending, paid, accepting, accept_error, done "status": "new", // Способы оплат: 0 - бесплатно, 1 - автоматическая оплата, 2 - ручная оплата "payment_mode": 0, "title": "", "updated_at": null, "user_id": 130 } </pre>
Ошибки	400 - ошибки в параметрах запроса, отрицательный баланс бесплатных 401 - некорректная сессия. 500 - невозможность получить данные о пользователе системы и прочие ошибки.

Таблица 2 – API запрос к подсистеме загрузки данных для просмотра списка задач

Тип запроса	GET
URL	/api/tasks/[?all]
Ответ	application/json 200 OK
	<pre>[{ // Обозначение внешней системы "adapter_name": "copernicus", // Дата создания задачи "created_at": "2021-11-16T15:34:32.334226", // Идентификатор снимка во внешней системе "identifier": "ac63f549-9b42-4c83-8524-b690497b406f", // Тип источника "source_type": "Sentinel2_2A", // Статус задачи (new, processing, deferred, done, error) "status": "done", // Внутренний идентификатор задачи "task_id": 19, // Название снимка "title": "S2A_MSIL2A_20200827T075611_N0214_R035_T39UUV_ 20200827T105108", // Дата обновления задачи "updated_at": "2021-11-16T15:54:20.668120" }]</pre>
Ошибки	401 - некорректная сессия 403 - запрос не от администратора 500 - прочие ошибки.

Таблица 3 – API запрос к подсистеме обработки для создания новой задачи

Тип запроса	POST
URL	/api/tasks/
Запрос	application/json

```

{
  // Информация о растрах, которые будут обработаны
  (JSON-поле в минимальном варианте загруженного
  растра)
  "rasters": [
    {
      "selection": {
        "source_type": "Sentinel2_2A",
        "title":
        "S2A_MSIL2A_20210829T074611_N0301_R135_T39UUU_
        20210829T104722",
        "index_path":
        "Sentinel2_2A/S2A_MSIL2A_20210829T074611_N0301_R1
        35_T39UUU_20210829T104722/data"
      }
    }
  ],
  // Карта с растровыми контейнерами
  "map_id": 2,
  // Выбранные классы для сохранения
  "processing_classes": [
    {
      "class_name": "",
      // Признак необходимости сохранять векторные
      результаты обработки, по-умолчанию - true
      "save_vector": true,
      // Код слоя для сохранения векторного результата.
      "vector_code": "layer_water",
      // Признак необходимости сохранить растровую
      маску, по-умолчанию - true
      "save_raster": true,
      // Код слоя для сохранения растрового результата.
      "raster_code": "",
      // Признак необходимости сохранить растровую
      маску в директорию исходного растра, по-умолчанию -
      false
      "save_mask": false,
      // Имя TIF файла без расширения для сохранения
      маски, по умолчанию совпадает с именем класса.

```

	<pre> "mask_code": "water" }], // Алгоритм обработки "processing_type": "landcover", // Id контейнера, по которому будет отсекается входной // контейнер для ускорения обработки "crop_container": 100500, // Информация о применяемых масках валидности, // опционально "validity": { // Перечень масок валидности "valid": ["footprint"], // Перечень масок инвалидности "invalid": ["clouds"], // Применять маски до или после сегментации "before": false } } </pre>
Ответ	application/json 200 OK
	<pre> // Созданные задачи обработки [{ // Дата создания "created_at": "2021-12-09T09:18:08.767244", // Группа - задачи созданные в одном запросе "group_id": "c7257284f2664137a561853e94f91795", // Пользователь, создавший задачу "initiator": "admin", // Идентификатор карты "map_id": 2, // Выбранные классы для сохранения "processing_classes": [</pre>

```

{
  "class_name": "",
  // Успешность завершения импорта векторного
результата (null если не выполнялся)
  "vector_success": null,
  // Успешность сохранения растровой маски (null
если не выполнялся)
  "raster_success": null,
  // Успешность сохранения растровой маски для
исходного растра (null если не выполнялся)
  "mask_success": null,
  "proc_cls_id": 11,
  "vector_code": "layer_fields",
  "raster_code": "raster_fields",
  "mask_code": "fields",
  // Успешность завершения обработки (null если
не выполнялся)
  "success": null,
  "task_id": 1
}
],
// Алгоритм обработки
"processing_type": "landcover",
// Проект (БД)
"project": "dbname",
// Обрабатываемый растр
"raster": {
  "selection": {
    "source_type": "Sentinel2_2A",
    "title":
"S2A_MSIL2A_20210829T074611_N0301_R135_T39UUU_
20210829T104722",
    "index_path":
"Sentinel2_2A/S2A_MSIL2A_20210829T074611_N0301_R1
35_T39UUU_20210829T104722/data"
  }
},
// Id контейнера, по которому будет отсекается
входной контейнер для ускорения обработки

```

	<pre> "crop_container": 100500, "validity": { "valid": ["footprint"], "invalid": ["clouds"], "before": false }, // Статус обработки (new, processing, error, done) "status": "new", // Идентификатор задачи "task_id": 1, // Дата обновления "updated_at": null }, { "created_at": "2021-12-09T09:18:08.974101", "group_id": "c7257284f2664137a561853e94f91795", "initiator": "admin", "map_id": 2, "processing_classes": [{ "class_name": "fields", "vector_success": null, "raster_success": null, "mask_success": null, "proc_cls_id": 12, "vector_code": "layer_fields", "raster_code": "raster_fields", "mask_code": "fields", "success": null, "task_id": 2 }], "processing_type": "landcover", "project": "dbname", "raster": { </pre>
--	---

	<pre> "selection": { "source_type": "Sentinel2_2A", "title": "S2A_MSIL2A_20210829T074611_N0301_R135_T39UUU_ 20210829T104722", "index_path": "Sentinel2_2A/S2A_MSIL2A_20210829T074611_N0301_R1 35_T39UUU_20210829T104722/data" } }, "status": "new", "task_id": 2, "updated_at": null }] </pre>
Ошибки	400 - ошибки в параметрах запроса 401 - некорректная сессия 403 - запрос не от администратора 500 - прочие ошибки.

Таблица 4 – API запрос к подсистеме обработки для просмотра списка задач

Тип запроса	GET
URL	/api/tasks/[?map_id=<map_id>]
Ответ	application/json 200 OK
	Аналогичен ответу запроса на создание задачи
Ошибки	401 - некорректная сессия 403 - запрос не от администратора 500 - прочие ошибки

Таблица 5 – API запрос к подсистеме обработки для просмотра конкретной задачи

Тип запроса	GET
-------------	-----

URL	/api/tasks/<task_id>/
Ответ	application/json 200 OK
	Аналогичен элементу ответу запроса на создание задачи
Ошибки	401 - некорректная сессия 403 - запрос не от администратора 500 - прочие ошибки - задача не найдена

ПРИЛОЖЕНИЕ 2

ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦ БАЗЫ ДАННЫХ

Таблица 1 – Описание таблицы подсистемы подбора данных ДЗЗ в БД Системы

Атрибут	Тип	Описание
identifier	varchar	Идентификатор
title	varchar	Наименование
source_type	varchar	Тип источника
status	external_modules.tt_sat_ s_task_status	Статус
download_params	json	Параметры загрузки
external_task_id	integer	Идентификатор задачи
target_map_id	integer	Ссылка на данные
initiator:	varchar	Инициатор
project	varchar	Проекция
created_at	timestamp	Дата и время создания задачи
updated_at	timestamp	Дата и время последнего обновления задачи
adapter_name	varchar	Адаптированное наименование
container_id	integer	Контейнер
selection	json	Выборка
crop_container	integer	Результирующий контейнер

Таблица 2 – Описание таблицы заявок подсистемы загрузки данных в БД Системы

Атрибут	Тип	Описание
adapter_name	varchar	Адаптированное наименование
source_type	varchar	Тип источника
identifier	varchar	Идентификатор
status	external_modules.tt_sat_ d_task_status	Статус
created_at	timestamp	Дата и время создания задачи

updated_at	timestamp	Дата и время последнего обновления задачи
title	varchar	Описание

Таблица 3 – Описание таблиц классов подсистемы обработки в БД Системы

Атрибут	Тип	Описание
class_name	varchar	Имя класса
vector_code	varchar	Векторные данные
success	boolean	Итог
vector_success	boolean	Итог по вектору
task_id	integer	Идентификатор задачи
raster_code	varchar	Растровые данные
raster_success	boolean	Итог по растру
mask_code	varchar	Маска
mask_success	boolean	Итог по маске

Таблица 4 – Описание таблицы задач подсистемы обработки в БД Системы

Атрибут	Тип	Описание
group_id	varchar	Идентификатор
status	external_modules.tt_sat_processing_task_status	Статус
processing_type	varchar	Тип обработки
map_id	integer	Идентификатор карты
initiator	varchar	Инициатор
project	varchar	Проекция
created_at	timestamp	Дата и время создания задачи
updated_at	timestamp	Дата и время последнего обновления задачи
crop_container	integer	Результирующий контейнер
raster	json	Растр
validity	json	Возможность обработки

Таблица 5 – Описание таблицы хранилища в БД Системы

Атрибут	Тип	Описание
source_type	varchar	Тип источника
sat_datetime	timestamp	Дата
data_path	varchar	Расположение данных
preview_path	varchar	Расположение превью снимков
cloudness	geometry	Облака
footprint	geometry	Тени
meta	json	Метаданные
acquired_at	timestamp	Дата поступления

ПЕРЕЧЕНЬ ТЕРМИНОВ

В документе используются следующие термины:

Система	– облачное платформенное решение «Pixel.AI»
Администратор	– роль пользователя, относящегося к персоналу Системы. Данная роль наделена полными правами доступа к административному интерфейсу Системы
Аутентификация пользователя	– процедура проверки подлинности пользователя, проверяющая соответствие имени введенной учетной записи (логина) и введенного пароля
Информационная система	– организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы. Информационные системы предназначены для хранения, обработки, поиска, распространения, передачи и предоставления информации
Пользователь	– физические и юридические лица, заинтересованные в процессе верификации информации об объектах посредством использования данных ДЗЗ (спутниковых снимков).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В документе используются следующие сокращения:

ГОСТ	–	государственный стандарт
ДЗЗ	–	дистанционное зондирование Земли
КА	–	космический аппарат
ПО	–	программное обеспечение
СУБД	–	система управления базами данных
API	–	Application Programming Interface, программный интерфейс приложения
SPA	–	Single Page Application, одностраничное веб-приложение

[illegible]