VisionLabs
MACHINES CAN SEE

VISIONLABS LUNA PLATFORM 5

Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации ПО

ООО «ВижнЛабс»

123458, г. Москва, ул. Твардовского д. 8, стр. 1

+7 (499) 399 3361

info@visionlabs.ru

www.visionlabs.ru

Оглавление

Глоссарий

Термин	Определение
LUNA PLATFORM	Система распознавания лиц.
Параметры лиц	Эмоции, параметры рта, положение головы и т. д.
Базовые атрибуты	Возраст, пол, расовая принадлежность.
Биометрический образец	Изображения, на которых присутствует лицо и которые соответствуют определенным стандартам.
Лица	Изменяемые объекты, содержащие информацию об одном лице.
Дескриптор	Набор уникальных свойств, получаемых из биометрического образца. Дескриптор требует гораздо меньше памяти по сравнению с образцом и используется при сравнении лиц.
Перекрытие	Состояние объекта (глаз, рта), при котором он скрыт другим объектом.
Характерные точки	Особые точки на лице, соответствующие положению определенного объекта на лице. Их определяет LUNA PLATFORM.
Очередь сообщений	Сервис, получающий и хранящий сообщения в соответствующей очереди, пока их не получит другой сервис.
Ключ маршрутизации	Имя очереди в сервисе очереди сообщений.
Бакет	Хранилище объектов.
LUNA PLATFORM Enterprise	ПО VisionLabs LUNA PLATFORM Enterprise включено в Единый реестр российских программ для электронных вычислительных машин и баз данных Запись в реестре №5878 от 20.09.2019 Доступны версии 3 и 4 LPE.

Сокращенное название	Расшифровка
NN	Нейросеть
LP	LUNA PLATFORM
DB	База данных
MQ	Очередь сообщений
UI	Пользовательский интерфейс
SW	Программное обеспечение
GC	Сбор мусора
API	LUNA PLATFORM API
Faces	LUNA PLATFORM Faces
Image Store	LUNA PLATFORM Image Store
Matcher	LUNA PLATFORM Matcher
Events	LUNA PLATFORM Events
Sender	LUNA PLATFORM Sender
Handlers	LUNA PLATFORM Handlers
Python Matcher	LUNA PLATFORM Python Matcher
Python Matcher Proxy	LUNA PLATFORM Python Matcher Proxy
Backport 3	LUNA PLATFORM Backport 3
Backport 4	LUNA PLATFORM Backport 4
Admin	LUNA PLATFORM Admin
Configurator	LUNA PLATFORM Configurator
Tasks	LUNA PLATFORM Tasks
Licenses	LUNA PLATFORM Licenses
User Interface 3	LUNA PLATFORM User Interface 3
User Interface 4	LUNA PLATFORM User Interface 4
LPE	LUNA PLATFORM Enterprise

Введение

LUNA PLATFORM 5 (Далее – LP) это система распознавания лиц.

Данный документ описывает:

- возможности системы и задачи, выполняемые сервисами
- создаваемые объекты
- общие процессы
- архитектуру и взаимодействие сервисов
- схемы обработки запросов
- структуру баз данных
- конфигурации сервисов
- возвращаемые ошибки

Обзор

LP предназначена для решения следующих задач:

- Обработка и анализ изображений:
  - распознавание лиц на фотографиях;
  - оценка базовых атрибутов (возраст, пол, расовая принадлежность) и свойств лиц (эмоции, направление взгляда, параметры глаз и рта);
- Поиск схожих лиц в базе данных;
- Хранение получаемых атрибутов лиц в базах данных;
- Создание *списков* для поиска;
- Сбор статистики;
- Гибкое управление запросами для соответствия требованиям обработки пользовательских данных.

Доступны следующие дополнительные возможности:

- Конфигурация всех сервисов LP с помощью одного сервиса конфигурации;
- Проверка наличия живого человека в кадре;
- Управление пользовательским аккаунтом посредством сервиса Admin;
- Уведомления о событиях посредством сервиса Sender.

Аппаратные и программные требования

Минимальные аппаратные требования

Следующие минимальные системные требования необходимы для установки программного пакета LUNA PLATFORM:

- CPU Intel, минимум 4 физических ядра с тактовой частотой 2.0 GHz или выше. Требуется поддержка набора инструкций AVX2 для CPU;
- RAM DDR3 (рекомендуется DDR4), 8 Гб или выше.
- Свободное место на диске - минимум 80 Гб.

> Рекомендуется использование SSD для баз данных и хранилищ Image Store.

- Доступ к Интернету (для контейнеров и дополнительных загрузок ПО).

Примечание Приведенная выше конфигурация обеспечит минимальную мощность для работы ПО, но для использования системы в продуктивном контуре этого недостаточно. Требования для использования системы в продуктивном контуре рассчитываются в зависимости от предполагаемой нагрузки.

GPU

Для ускорения GPU необходим NVIDIA GPU. Поддерживаются следующие архитектуры:

- Pascal или более новые.

Требуется минимум 6Гб оперативной или выделенной видеопамяти. Рекомендуется 8 Гб VRAM или более.

> CUDA версии 11.2 уже установлена в Docker контейнере в сервисе Handlers. Рекомендуемые драйверы NVIDIA - r450, r455.

Для корректной работы приложения аппаратное обеспечение должно отвечать следующим минимальным требованиям:

- CPU с частотой 2 ГГц и выше;
- 4 Гб оперативной памяти и выше;
- 400 Мб свободного места на жестком диске.

Программные требования

Для запуска LUNA PLATFORM может использоваться RedOS (РЕД ОС) версии 7.3 и выше, CentOS версии 7.8 и выше.

Общие процессы LP

Короткое описание сервисов LP

Все сервисы LP можно разделить на основные и дополнительные. Запустить и использовать LP без основных сервисов нельзя, тогда как дополнительные сервисы не обязательны для запуска LP, но предоставляют больше возможностей. У большинства сервисов имеется собственная база данных или файл для хранения данных.

> Базы данных, указанные в таблице 1, не входят в комплект поставки и требуют отдельной установки. Обратите внимание, что все данные, хранящиеся в базах данных, привязываются к конкретному пользователю (его аккаунту). Более подробное описание см. в разделе “Аккаунты”.

Таблица 1. Основные сервисы

Сервис	Описание	База данных
API	Основной интерфейс доступа для работы с LP. Получает запросы, распределяет задачи между другими сервисами LP.	
Handlers	Распознает лица на изображениях, получает параметры лиц и создает биометрические образцы. Получает из образцов дескрипторы. Получает базовые атрибуты изображений. Создает и хранит обработчики	PostgreSQL/ Oracle
Python Matcher	Выполняет операции сравнения дескрипторов	
Faces	Создает лица, списки и атрибуты. Сохраняет эти объекты в базе данных. Позволяет другим сервисам получать требуемые данные из базы данных	PostgreSQL/ Oracle, Redis
Image Store	Хранит: биометрические образцы, отчеты о долгом выполнении задач, создаваемые кластеры и дополнительные метаданные	Local storage/ Amazon S3
Events	Хранит события в базе данных. Этот сервис можно отключить, но рекомендуется его использовать, в случае если вы планируете сохранять события	Vertica/ PostgreSQL
Licenses	Проверяет данные лицензии и возвращает информацию о них.	

Admin	Позволяет выполнять общие административные процедуры	PostgreSQL/ Oracle
Configurator	Хранит конфигурации всех сервисов в одном месте	PostgreSQL/ Oracle
Tasks	Выполняет длительные задачи, такие как сбор мусора, получение дескрипторов с помощью новой версии нейросети, кластеризация	PostgreSQL/ Oracle

Таблица 2. Дополнительные сервисы

Сервис	**Описание**	**База данных**
Sender	Отправляет уведомления о создаваемых событиях через веб-сокет	Redis
Backport 3	Данный сервис используется для обработки запросов LUNA PLATFORM 3 Enterprise с использованием LUNA PLATFORM 5.	PostgreSQL/ Oracle
Backport 4	Данный сервис используется для обработки запросов LUNA PLATFORM 4 Enterprise с использованием LUNA PLATFORM 5.	
User Interface 3	Пользовательский интерфейс, используемый для визуального представления возможностей, предоставляемых сервисом Backport 3. Он не включает в себя весь функционал LPE 3.	
User Interface 4	Пользовательский интерфейс, используемый для визуального представления возможностей, предоставляемых сервисом Backport 4. Он не включает в себя весь функционал LPE 4.	

Существует несколько сторонних сервисов, обычно используемых с LP.

> Эти сервисы не описаны в данном документе. См. соответствующую документацию, предоставляемую поставщиками сервисов.

Таблица 3. Сторонние сервисы

Сервис	Описание	Поддерживаемые сервисы
Balancer	Осуществляет балансировку запросов между сервисами LP, в случае если запущено несколько аналогичных сервисов. К примеру, можно балансировать запросы между сервисами API или между двумя контурами LP при масштабировании.	NGINX
Monitoring system	Используется для сбора статистики о запросах и работе системы	Supervisor
Monitoring database	База данных, используемая для хранения собранной информации по мониторингу системы	Influx
Log rotation service	Все сервисы LP пишут логи, размер которых может очень сильно расти. Сервис распределения логов позволяет удалять старые файлы логов и освобождать место на диске	Logrotate, etc.

Следующие разделы дают общее представление о работе LP, ее сервисах и создаваемых типах данных. Описания запросов, структур баз данных и другие технические детали в данном разделе не приводятся.

LP состоит из сервисов. Сообщение между ними происходит посредством HTTP запросов: сервис получает запрос и возвращает ответ.

Информацию об архитектуре LUNA PLATFORM 5 можно найти в разделе “Взаимодействие LP сервисов”.

Аккаунты

Все данные, хранящиеся в базах данных LP, принадлежат конкретным аккаунтам. Аккаунт представлен полем “Luna-Account-Id” (ID аккаунта), передаваемым с каждым запросом, вносящим изменения в базу данных.

> “Luna-Account-Id” должен быть уникальным для каждого пользователя.

“Luna-Account-Id” имеет формат UUID. Вы можете создать “Luna-Account-Id” с помощью любого доступного инструмента или задать его вручную в соответствии с требуемым форматом.

Обработка исходного изображения и создание образцов

LP предназначена для работы с фотографиями (изображения или отдельные кадры из видео). LP получает фото посредством HTTP-запроса в сервис API.

> Фото должно быть в одном из стандартных форматов (JPG, PNG, TIFF и т. д.) или перекодировано в BASE64.

Порядок обработки изображения представлен на рисунке 1.

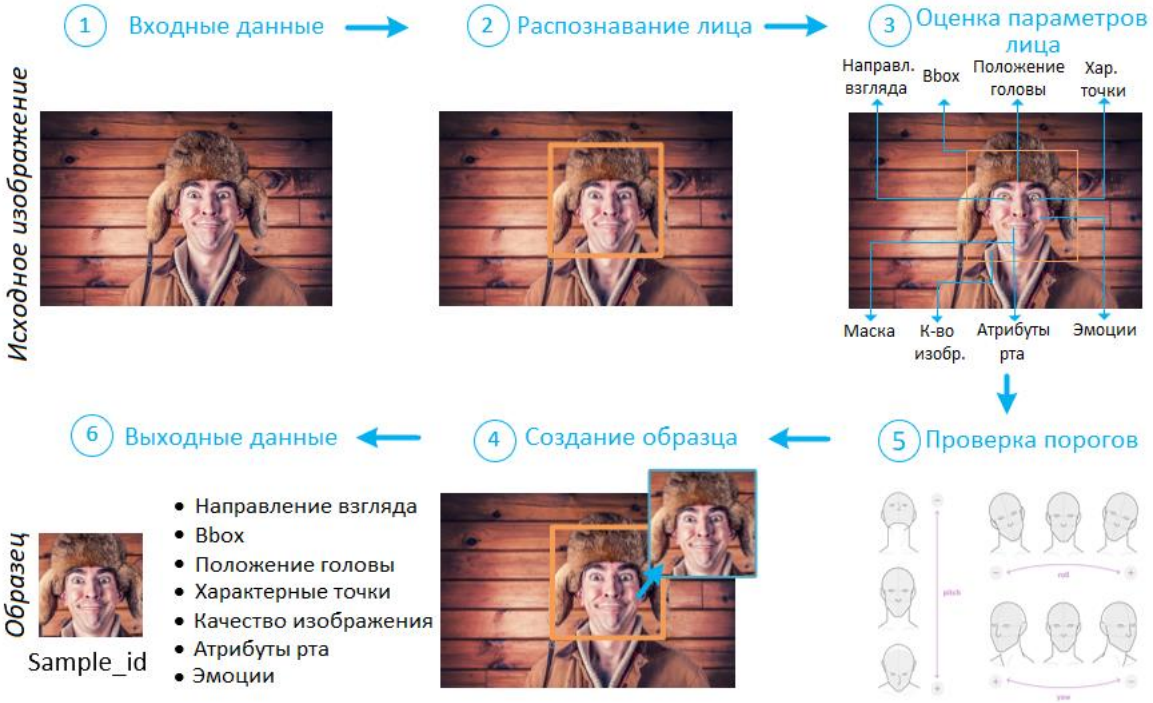

Рисунок 1. Порядок обработки изображения

Основные шаги при обработке изображений приведены ниже:

- Распознавание лица (см. п. 2, рис. 1). Можно сконфигурировать сервис Handlers для:
  - обработки изображений, на которых присутствует несколько лиц;
  - обработки изображений, на которых присутствует одно лицо;
  - поиска лица с наилучшим качеством изображения.
- Создание образца (см. п. 4, рис. 1). Созданному образцу присваивается “sample_id”. Образец соответствует особому формату и обрабатывается сервисами LP.

> Эти шаги выполняются сервисом Handlers.

Следует использовать “sample_id” для дальнейшей обработки изображения.

Сервис Handlers может производить оценку параметров лица для каждого найденного лица по запросу (см. п. 3, рис. 1). Можно оценить следующие параметры:

- *Направление взгляда* (углы поворота для обоих глаз);
- *Ограничивающий прямоугольник (bbox)* (высота, ширина, координаты “X” и “Y”). Ограничивающий прямоугольник - область с определенными координатами, в которой найдено лицо;
- *Положение головы* (наклон вперед, в сторону, поворот);
- *Пять или шестьдесят восемь характерных точек*. Характерные точки — это особые точки на лице, по которым проводится оценка параметров лица. Количество характерных точек зависит от того, какие параметры лица подлежат оценке;
- *Качество изображения* (свет, тень, насыщенность, размытость, освещенность и зеркальность);
- *Атрибуты рта* (перекрытие, наличие улыбки);
- *Оценка наличия маски* (медицинская или тканевая маска на лице, маска отсутствует, рот перекрыт);
- *Вероятностная оценка эмоций* (гнев, отвращение, страх, счастье, нейтральное выражение, грусть, удивление).

При необходимости можно настроить фильтрацию по порогам угла рыскания, крена и тангажа (см. п. 5, рис. 1)

> Данные параметры не хранятся в базе данных. Их можно получить только в ответе сервиса.

Извлечение дескрипторов и создание атрибутов

Биометрические образцы используются для извлечения дескрипторов. Дескрипторы — это наборы параметров, считываемых с лиц на изображениях. Дескрипторы используются для сравнения лиц.

> Нельзя сравнить два лица при отсутствии извлеченных дескрипторов. К примеру, при необходимости сравнить лицо из базы данных с входящим изображением лица, необходимо извлечь дескриптор для данного изображения.

Помимо дескрипторов можно получить базовые атрибуты лица: возраст, пол, расовую принадлежность.

> Операции извлечения дескрипторов и атрибутов выполняются сервисом Handlers.

И дескриптор, и базовые атрибуты для одного и того же лица называются *атрибутами*.

Можно извлекать дескрипторы и базовые атрибуты с использованием нескольких образцов одного и того же лица одновременно. Таким образом можно получить агрегированный дескриптор и агрегированные базовые атрибуты.

Точность сравнения и оценки базовых атрибутов посредством агрегированного дескриптора выше.

> Агрегированный дескриптор можно получить из изображений, но не из уже созданных дескрипторов.

Ресурс “/extract” используется для оценки базовых атрибутов и извлечения дескрипторов.

Хранить атрибуты можно во внешней базе данных вне LP и использовать их в LP только тогда, когда требуется.

Все созданные атрибуты изначально являются временными и хранятся в базе данных в течение ограниченного периода времени (по умолчанию 300 секунд). Временные атрибуты удаляются после истечения указанного периода времени. Таким образом не требуется удалять эти данные вручную.

> Следует создать лицо с использованием существующего атрибута, чтобы данные по атрибуту хранились в базе данных постоянно.

Сохранение внешних образцов

Вы можете загрузить биометрический образец и использовать его для дальнейшей обработки.

Лица и списки

Объект *лицо* используется для хранения информации, принадлежащей одному лицу. Поля объекта *лицо* можно обновлять.

Можно создать новое *лицо*, указав ID существующего атрибута или используя дескриптор в чистом виде и данные базовых атрибутов.

Лица обычно используются для хранения информации об известных людях. К примеру, можно создать список работников и использовать его для операций сравнения. Поле External ID используется для того, чтобы связать между собой несколько лиц.

События и обработчики

Обработчик — это набор правил (политик) для обработки изображений.

События — это отчеты с информацией, получаемой после обработки изображений с помощью обработчика.

> В отличие от лиц события нельзя изменять после создания.

Обработчики

С помощью обработчика можно:

- Указывать параметры распознавания лиц и оценочные параметры.
- Указывать, необходимо ли выполнять детекцию тела человека.

- Активизировать базовые атрибуты и извлечение дескрипторов.
- Выполнять сравнение дескрипторов для получения результатов сравнения и использования этих результатов в качестве фильтров для дальнейших политик.
- Конфигурировать автоматическое создание лиц с помощью фильтров. Фильтры можно задавать в соответствии с оценочными базовыми атрибутами и результатами сравнения дескрипторов.
- Конфигурировать автоматическое прикрепление созданных лиц к спискам с помощью фильтров. Фильтры можно задавать в соответствии с указанными базовыми атрибутами и результатами сравнения дескрипторов.
- Выбирать объекты для сохранения после обработки изображений.
- Конфигурировать автоматическое добавление тега для события с помощью фильтров. Фильтры можно задавать в соответствии с оценочными базовыми атрибутами и результатами сравнения дескрипторов.

Использование обработчиков имеет следующие особенности:

- **Два запроса для обработки базовых изображений**

  Следует создать обработчик с помощью запроса “create handler” и затем указать его для создания событий в запросе “generate events”. Таким образом требуется только два запроса для обработки группы изображений.

- **Все правила обработки в одном месте**

  Можно легко редактировать существующий обработчик или создать новый под новую задачу. Таким образом нет необходимости создавать и настраивать несколько различных запросов для выполнения базовых операций с изображениями.

Динамические обработчики

Обработчики могут быть статическими и динамическими.

Если обработчик статический, можно задавать его параметры во время его создания, а затем задать ID созданного обработчика в процессе создания события.

Если обработчик динамический, можно менять его параметры во время запроса создания событий. Динамические обработчики позволяют вам разделять технические параметры (пороговые значения и параметры качества, скрытые от внешних пользователей) и бизнес-логику.

Пример использования динамического обработчика:

Например, необходимо отделить пороговые значения для положения головы от других параметров обработчика.

Можно сохранить эти значения в вашу внешнюю базу данных и включить логику автоматической подстановки этих данных при создании событий (например, в вашем внешнем приложении).

Внешний пользователь отправляет запросы на создание событий и указывает требуемые списки и другие параметры. Пользователь не знает о пороговых значениях и не может их изменять.

События

События обладают следующими особенностями:

- Информация о времени, месте и источнике

Каждое событие может включать в себя следующую информацию:

  - *Время возникновения события*. Время, когда событие произошло в видеопотоке.
  - *Место возникновения события*. Можно напрямую указать город, район, улицу, номер дома. Также можно указать географические координаты места, где произошло событие.
  - *Источник возникновения события*. Можно настроить камеру в данном поле так, чтобы различать события, получаемые из разных источников.
- Теги

Можно задать теги таким образом, чтобы несколько тегов относились к одной категории. Их можно задать вручную или автоматически (с помощью фильтров).

- Статистика

Можно осуществлять сбор статистики по следующим событиям:

  - Группировать события по частоте или временным интервалам;
  - Фильтровать события в соответствии со значениями их параметров;
  - Считать количество созданных событий в соответствии с фильтрами;
  - Определять минимальное, максимальное и среднее значения параметров для существующих событий;
  - Группировать существующие события в соответствии с заданными значениями параметров.

Если вам необходимо собирать статистику после создания события, следует сохранять создаваемые события в базе данных.

- Уведомления через веб-сокеты

Вы можете получать уведомления о событиях с помощью веб-сокетов. Уведомления отправляются в соответствии с заданными фильтрами. Например, при распознавании работника можно отправить уведомление в приложение, управляющее турникетом, и он откроется.

- **Сравнение дескрипторов по событиям**

События можно использовать в качестве эталонов и кандидатов для операций сравнения.

> Дескриптор можно привязать к событию, хранимому в базе данных событий.

> При создании событий можно включить агрегацию всех атрибутов из изображений в запросе. Убедитесь в том, что на каждом изображении только одно лицо и что лицо принадлежит одному и тому же человеку.

Сравнение дескрипторов

При наличии дескрипторов LP позволяет искать похожие лица в базе данных посредством сравнения данного дескриптора с дескрипторами, хранящимися в базе данных.

Сервис Matcher позволяет вам сравнивать дескрипторы и получать степень схожести, значение которой лежит в интервале от 0 до 1. Высокая степень схожести означает, что два дескриптора принадлежат одному и тому же человеку.

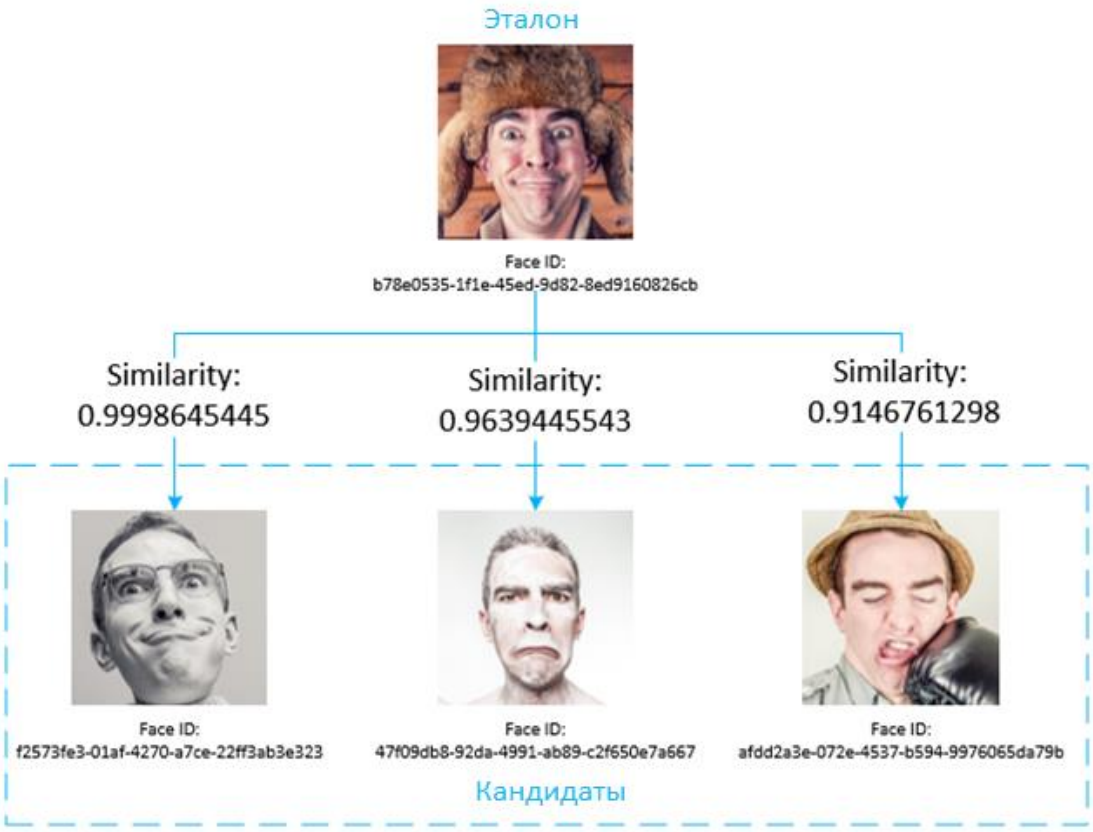

Рисунок 2. Пример сравнения

Исходные изображения сравниваемых объектов представлены в виде *эталонов* (объектов, подлежащих сравнению) и *кандидатов* (набора объектов, с которыми производится сравнение). Каждый эталон сопоставляется с каждым из заданных кандидатов.

Можно выбрать следующие объекты в качестве эталонов и кандидатов.

Эталоны (указываются с помощью массива из ID):

- Атрибуты

- *Лица*
- Внешние ID *лиц*
- События (лицо или тело)
- Дескрипторы

Кандидаты (указываются с помощью фильтров):

- Лица
- События (лицо или тело)
- Атрибуты

Эталоны указываются с помощью ID соответствующих объектов. Если задается несуществующий эталон (например, несуществующий ID указан в поле “event_id” или “face_id”), возвращается соответствующая ошибка.

Кандидаты указываются с помощью фильтров. Результаты сравнения возвращаются для тех кандидатов, которые соответствуют заданным фильтрам. Если таковых не обнаружено (например, несуществующий ID указан в поле “event_ids” или “face_ids”), не будет ни результата сравнения, ни ошибки. Поле результата будет пустым.

> Сравнение не может быть выполнено при отсутствии дескриптора для любого из сравниваемых лиц.

Для фильтрации событий можно:

- Указать камеру (поле “source”) и период (поля “create_time_gte” и “create_time_lt”);
- Указать теги (поле “tags”) для событий в качестве фильтров и выполнить сравнение для событий с помощью одних этих тегов;
- Указать географическую зону и выполнить сравнение с помощью событий, созданных только в этой зоне. Можно указать конкретную локацию (город, район, улица, дом) или область, заданную географическими координатами.

Для фильтрации *лиц* можно:

- Указать список внешних ID лиц для выполнения сравнения по ним;
- Указать список ID и выполнять сравнение по нему.

Запрос raw matching

Можно задавать в качестве эталонов и кандидатов дескрипторы с помощью запроса “raw matching”.

Все данные по дескрипторам предоставляются посредством запроса, таким образом вы можете использовать этот запрос при необходимости производить сравнение дескрипторов, не хранимых в базах данных LUNA PLATFORM.

Верификация

Можно использовать ресурс “/verifiers” для создания специального обработчика для верификации. Он включает несколько политик для обработки входящих изображений.

В этом обработчике можно задать “verification_threshold”.

Созданный верификатор следует использовать при отправке запросов в:

- ресурс “/verifiers/{verifier_id}/verifications”. Можно задать ID объектов, по которым должна производиться верификация.
- ресурс “/verifiers/{verifier_id}/raw”. Можно задать дескрипторы в чистом виде в качестве эталонов и кандидатов для сравнения. Т. к. обрабатываются дескрипторы в чистом виде, “verification_threshold” — это основной параметр, используемый из указанного верификатора.

Ответ включает в себя поле “status”. Оно показывает, успешно ли прошла верификация для каждой пары сравниваемых объектов. Она успешна, если схожесть двух объектов превышает значение, заданное в “verification_threshold”.

> Этот запрос используется для сравнения одного заданного объекта со входящим объектом. Для задач идентификации используйте другие запросы сравнения дескрипторов.

Отправка уведомлений через Sender

Можно отправлять уведомления о созданных событиях сторонним приложениям через веб-сокеты. Например, можно сконфигурировать LP для отправки уведомлений на мобильный телефон о прибытии VIP-гостей.

Когда LP создает новое событие, оно добавляется в специальную базу данных. Затем *событие* можно отправить в сервис Sender, если он активен.

Стороннее приложение должно быть подключено к сервису Sender через веб-сокет. Фильтр интересующих событий необходимо устанавливать для каждого приложения. Таким образом, клиент будет получать уведомления только об интересующих событиях.

Уведомление о новом событии отправляется сервисом Sender.

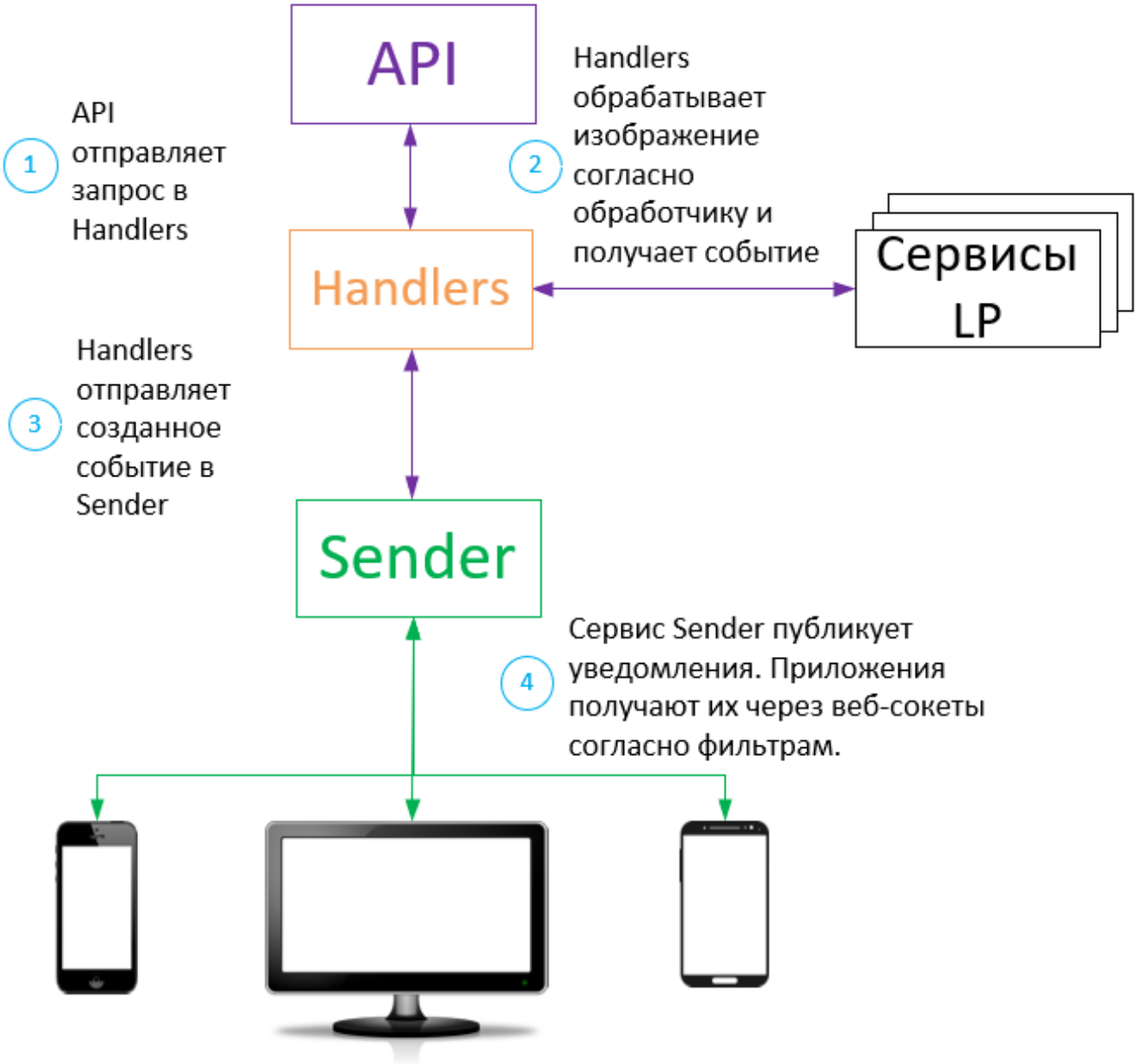

Рисунок 3. Работа сервиса Sender

Backports

В LP 5 сервисы Backport (ретроподдержка) — это механизм, позволяющий принимать запросы в формате предыдущих версий LP, но обрабатывать их в новой версии.

Ретроподдержка в LPE версии 3 и LPE версии 4 реализуется посредством сервисов Backport 3 и Backport 4 соответственно.

Благодаря этим сервисам нет необходимости писать новую интеграцию для LP 5 с нуля, когда пользователь обновляет предыдущую версию. Ретроподдержка позволяет вам отправлять запросы, аналогичные запросам из LUNA PLATFORM 3 и LUNA PLATFORM 4, и получать ответ в требуемом формате.

Случай использования:

К примеру, у вас есть интерфейсное приложение, отправляющее запросы в LUNA LPE версии 3.

При использовании сервиса Backport 3 запросы в LPE версии 3 принимаются сервисом, форматируются и отправляются в LUNA PLATFORM 5 в формате, соответствующем спецификации API. LP 5 обрабатывает этот запрос и отправляет ответ сервису Backport 3, который приводит все входящие результаты к формату LPE версии 3 и отправляет ответ.

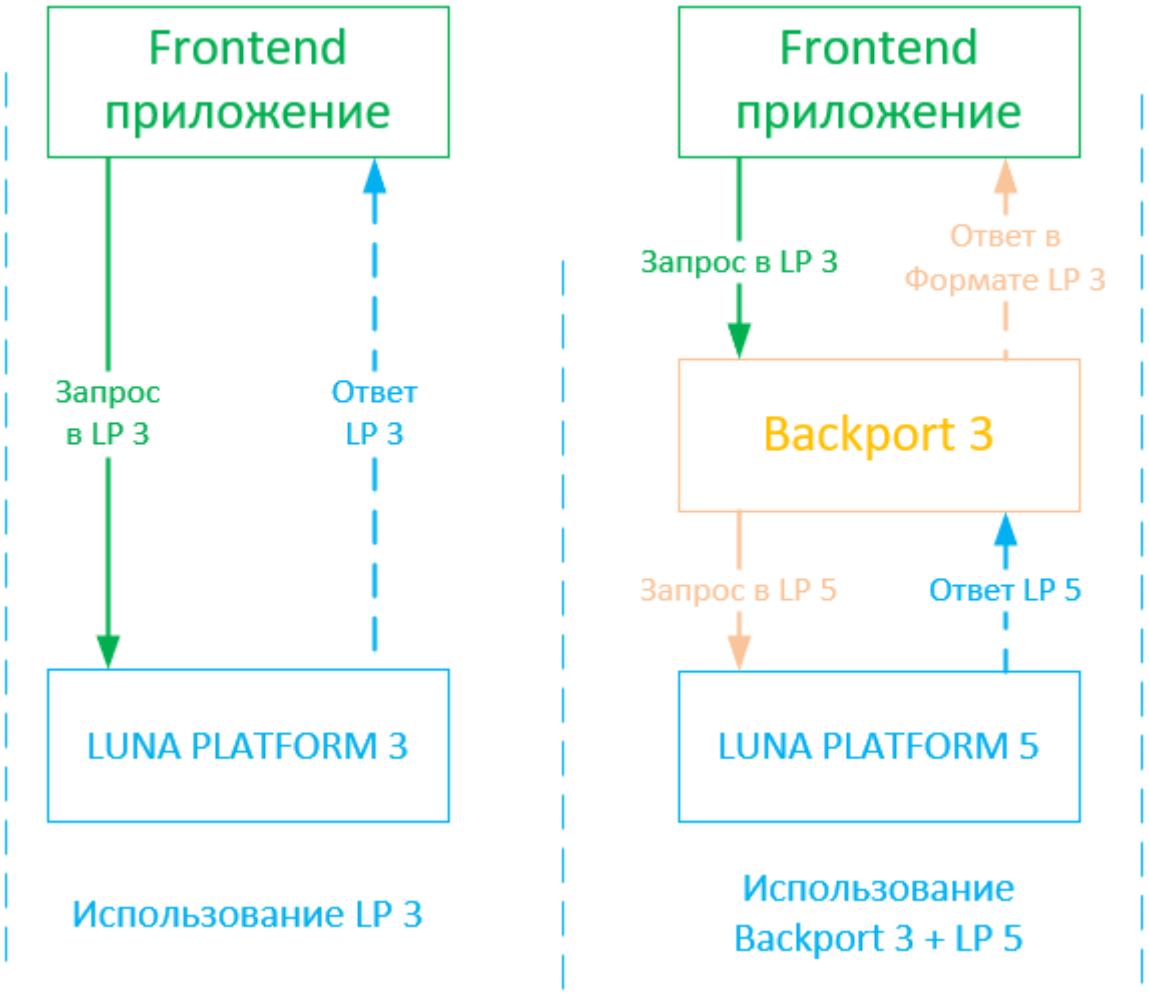

Рисунок 4. Работа с LPE версии 3 и Backport 3 и LP 5

Ограничения при работе с сервисами Backport

Так как таблицы сохраненных данных и сами данные различны в разных версиях LP, существуют особенности и ограничения при выполнении запросов.

Сервисы Backport и API в LP 5 можно использовать одновременно с соблюдением следующих ограничений:

- При использовании сервисов Backport рекомендуется не использовать один и тот же аккаунт для запросов в сервисы Backport и API в LUNA PLATFORM 5.
- Следует выполнять запросы администратора, которые не используют ID аккаунта, только в LUNA PLATFORM 5.

Дополнительные сервисы LP

Задачи

Запросы Tasks предоставляют дополнительные возможности для обработки больших объемов данных.

Чем больше размер массива обрабатываемых данных, тем больше времени занимает выполнение задачи. Когда задача создана, вы получаете ID задачи в качестве результата запроса.

Задача кластеризации

Задача кластеризации дает возможность группировать события и лица, принадлежащие одному и тому же человеку, в кластеры.

Можно задать фильтры для выбора обрабатываемых объектов.

При помощи задачи кластеризации можно, например, получать все ID событий, принадлежащих одному и тому же человеку и произошедших в течение указанного периода времени.

Задача создания отчета

Задача создания отчета позволяет вам получать отчет в формате CSV с расширенной информацией об объектах, объединенных в кластеры.

Можно выбрать столбцы, добавляемые в отчет. Можно получать в ответе изображения, соответствующие каждому из ID в каждом кластере.

Задача экспорта

Задача экспорта позволяет вам получать данные по событиям и/или лицам и экспортировать их из LP в CSV-файл.

Строки файла представляют собой запрашиваемые объекты и соответствующие образцы (при запросе).

Задача перекрестного сравнения

Перекрестное сравнение означает, что большое количество эталонов можно сравнить с большим количеством кандидатов. Таким образом, каждый эталон сравнивается с каждым кандидатом.

Как эталоны, так и кандидаты задаются с помощью фильтров для *лиц* и *событий*.

Задача привязки

Задача привязки позволяет вам:

- осуществлять привязку существующих лиц к спискам;
- создавать лица из существующих событий и привязывать их к спискам.

Лица с привязкой выбираются в соответствии с заданными фильтрами.

Описание сервиса Liveness

Алгоритм liveness позволяет LUNA PLATFORM обнаруживать атаки на биометрическое предъявление. Такая атака означает ситуацию, когда злоумышленник пытается использовать видео или фото другого человека для того, чтобы обойти систему распознавания и получить доступ к личным данным этого человека.

Атаки на биометрическое предъявление бывают следующих основных видов:

- Атака с использованием распечатанного фото. Используются одно или несколько изображений другого человека.
- Атака с воспроизведением видео. Используется видеоряд с другим человеком.
- Атака с распечатанной маской. Злоумышленник закрывает свое лицо изображением лица другого человека, вырезанным из фото.

Результаты проверки Liveness

Алгоритм liveness использует одно изображение для обработки и возвращает следующие данные:

- Liveness probability (вероятность живости) [0..1]. Здесь 1 означает реального человека, 0 - подмена. Этот параметр показывает вероятность того, что на изображении присутствует живой человек, то есть это не атака на биометрическое предъявление.
- Image quality (качество изображения) [0..1]. Здесь 1 означает хорошее качество, 0 - плохое. Данный параметр оценивает характеристики изображения, лица, внешних условий в целом. Это значение можно увеличить в соответствии с вашими условиями съемки.

Liveness

Liveness является частью сервиса Handlers.

Фильтрование по живости доступно в следующих сценариях:

- при выполнении операций сравнения;
- при выполнении задач;
- при отправке данных через веб-сокет.

Можно также задать параметр оценки живости при создании и сохранении событий вручную в ресурсе “handlers/{handler_id}/events/raw”.

Для многократно загружаемых изображений можно агрегировать результаты по живости для получения более точных данных.

Требования Liveness

Оценка живости не поддерживается для биометрических образцов (нормализованных изображений), т. к. они не отвечают требованиям к входящим изображениям.

Этот механизм оценки поддерживает изображения, полученные с помощью мобильных устройств и веб-камер (компьютера или ноутбука).

Минимальные требования к разрешению изображений:

- Мобильные устройства - 720 × 960 px
- Веб-камера (компьютера или ноутбука) - 1280 x 720 px

На изображении должно быть только одно лицо. Если на изображении два или более лиц, возвращается ошибка.

Минимальный размер распознаваемого лица - 200 пикселей.

Углы поворота вперед, в сторону и наклона не должны превышать 25 градусов в каждом направлении.

Минимальный отступ между лицом и границей изображения - 10 пикселей.

Взаимодействие сервисов LP

На картинках ниже представлены отдельные базы данных, заданные для каждого сервиса. Они приведены для наглядности.

Не нужно запускать отдельный экземпляр БД для каждого сервиса. Можно запустить один экземпляр (например, PostgreSQL) и использовать его для хранения данных каждого из сервисов LP. У каждого сервиса будет своя таблица в этой базе данных.

Основные сервисы

Сервис API обеспечивает RESTful интерфейс для выполнения детекции, извлечения и сравнения дескрипторов.

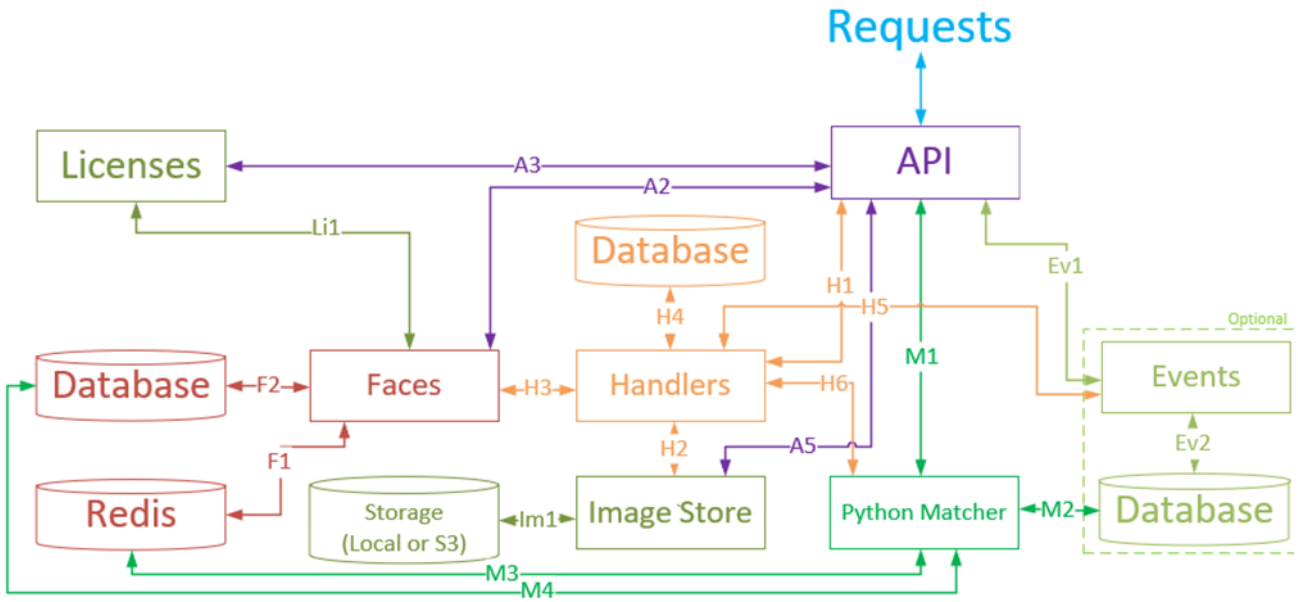

Рисунок 5. Взаимодействие основных сервисов

Запросы отправляются в LP через HTTP. Обычная ситуация - когда внешний сервис отправляет запросы в LP, получает результаты и обрабатывает их в соответствии с заданными в запросе параметрами.

API получает запросы, обрабатывает их и передает их в другие сервисы:

- Запросы на детекцию лица, оценку параметров лица и создание образцов *(Samples > Detect faces)* отправляются в сервис Handler **(H1)**;
- Запросы на создание временных атрибутов *(Attributes > Extract attributes)* отправляются в сервис Handler **(H1)**;
- Запросы на сравнение дескрипторов *(Matcher > Matching)* могут отправляться в сервис Python Matcher **(M1)**.

API отправляет запросы в сервис Faces для создания новых лиц, а также создания списков и их изменения *(раздел Faces)* **(A2)**.

Сервис API получает информацию о текущих лицензионных условиях из сервиса Licenses **(A3)** с помощью промежуточного ПО.

Сервис API отправляет биометрические образцы от внешних сервисов в сервис Image Store (Samples > Save sample) **(A5)**.

Handlers обрабатывает запросы на детекцию лица и создание атрибутов и оценивает параметры лиц.

Сервис Handlers обрабатывает запрос на детекцию из API **(H1)** и отправляет полученные биометрические образцы в сервис Image Store **(H2)**.

Сервис Handlers обрабатывает запрос на создание атрибутов из сервиса API **(H1)**, запрашивает существующие биометрические образцы из Image Store **(H2)** и отправляет созданные временные атрибуты в сервис Faces, который хранит их в базе данных Redis **(F1)**.

Сервис Handlers создает новые обработчики *(Handlers > Create handler)* и хранит их в базе данных Handlers **(H4)**.

Сервис API получает запрос на создание нового события *(Events > Generate events)*, он отправляется в сервис Handlers **(H1)** и обрабатывается в соответствии с тем обработчиком, чей ID указан в запросе.

Image Store получает биометрические образцы из сервиса Handlers и хранит их в SSD или в базе данных S3 и предоставляет к ним доступ **(Im1)**.

Licenses. Сервис Licenses получает информацию о количестве созданных лиц с атрибутами из базы данных Faces (Li1).

Faces отвечает за взаимодействие с базой данных Faces, которая хранит: *лица*, дескрипторы *лиц* и *списки* **(F2)**. Он дает доступ к сохраненным данным для сервисов API, Python Matcher и Tasks.

Faces также хранит созданные временные атрибуты в базе данных Redis **(F1)**.

Python Matcher может получать запросы на сравнение напрямую из сервиса API **(M1)**. Python Matcher отправляет запросы на сравнение в базу данных Faces **(M4)** или Events **(M2)**. Эти базы данных сравнивают дескрипторы и отправляют результаты в сервис API.

Python Matcher также может получать временные атрибуты, требуемые для сравнения, из базы данных Redis **(M3)**.

Events хранит и предоставляет информацию о событиях *(раздел Events)*. После создания события сервис Events получает созданное событие из сервиса Handlers **(H5)** и хранит его в базе данных Events **(Ev2)**.

> Использование сервиса Events можно активировать/дезактивировать в файле конфигурации сервиса API.

Административные и дополнительные сервисы

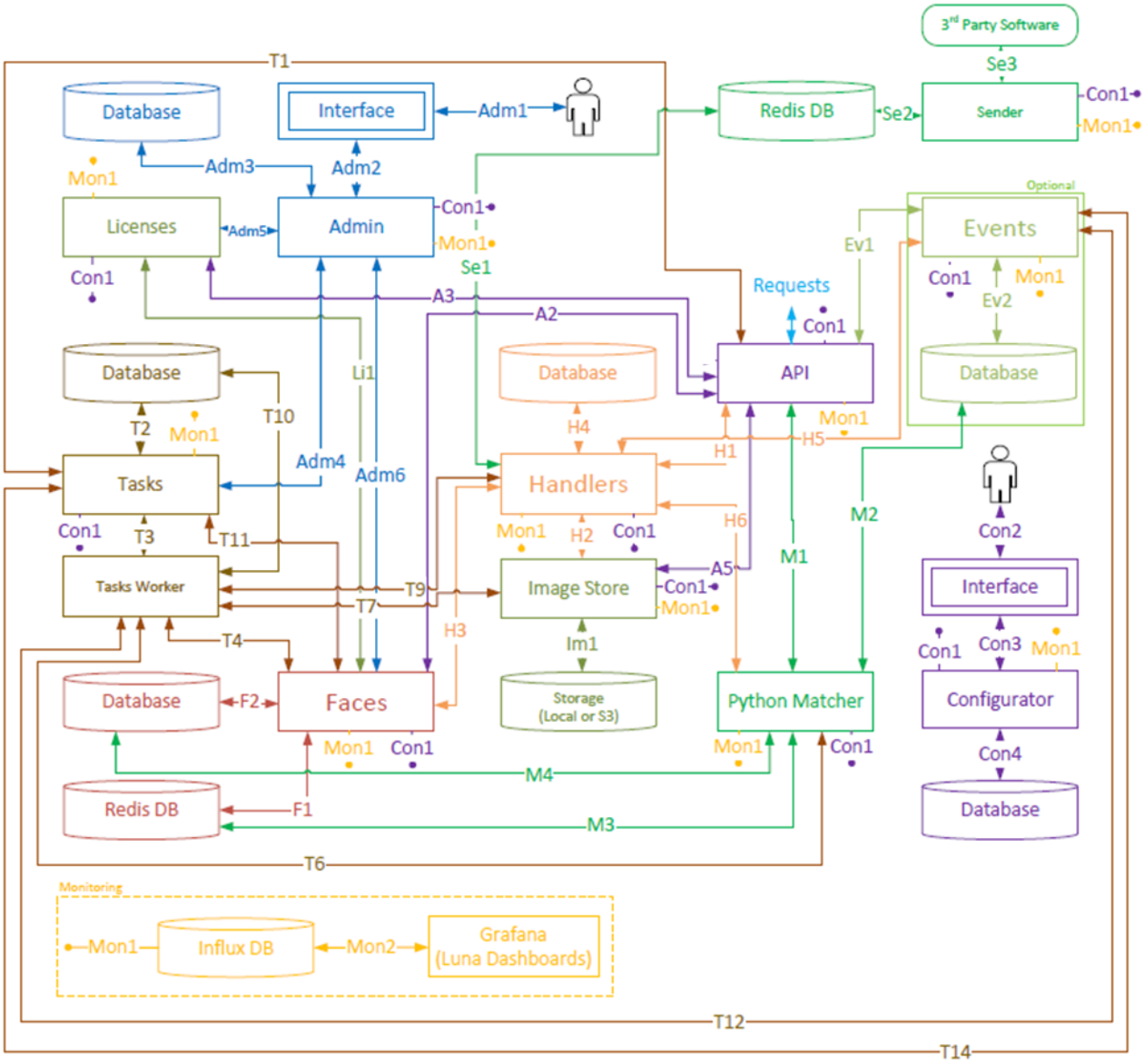

Рисунок 6. Взаимодействие дополнительных сервисов

Admin. Пользователь может управлять аккаунтами и выполнять другие действия через пользовательский интерфейс сервиса Admin **(Adm1)**. Соответствующие запросы отправляются в сервис Admin **(Adm2)**.

Вся информация о пользовательских аккаунтах хранится в базе данных Admin **(Adm3)**.

Сервис Admin отправляет запросы в сервис Tasks **(Adm4)**. Эти задачи выполняются для баз данных LP в целом, не для отдельного ID аккаунта.

Сервис Admin получает информацию о текущих лицензионных условиях из сервиса Licenses **(Adm5)** с помощью промежуточного ПО.

Сервис Admin проверяет, сколько на данный момент созданных лиц с привязанными атрибутами присутствует в базе данных Faces **(Adm6)** и подсчитывает процент заполненности базы данных.

Sender. Все созданные события принимаются базой данных Redis DB из сервиса Handlers (Se1). Внешние сторонние приложения подписываются на получение событий в соответствии с заданными фильтрами (Se3). Сервис Sender проверяет канал Redis, получает требуемую информацию и отправляет ее внешнему ПО **(Se2)**.

> Сервис Sender активируется/дезактивируется в конфигурационном файле сервиса API.

Configurator. Сервисы LP получают конфигурационные параметры из сервиса Configurator (Con1). Пользователь может управлять конфигурациями в сервисе Configurator посредством графического пользовательского интерфейса (Con2). Изменения отправляются в Configurator (Con3). Все изменения в конфигурационных файлах сохраняются в базе данных Configurator (Con4).

> Configurator используется по умолчанию для каждого сервиса Luna.

Tasks. Сервис Tasks получает запросы из сервиса API **(T1)**. Затем Tasks создает и хранит задачу в базе данных Tasks **(T2)**. Затем Tasks отправляет подзадачи своим “работникам” **(T3)**.

Сервис Tasks получает данные из сервиса Faces для задач кластеризации, привязки и дополнительных задач извлечения **(T11)**.

Сервис Tasks получает данные из сервиса Events для задач кластеризации и привязки **(T13, T12)**.

> Сервис Tasks активируется/дезактивируется в конфигурационном файле сервиса API.

Взаимодействие “работников” Tasks с другими сервисами зависит от типа задачи.

“Работники” Tasks получают данные из сервиса Faces для каждой обрабатываемой задачи **(T4)**.

“Работники” Tasks отправляют запрос в Python Matcher **(T6)** для задач кластеризации, перекрестного сравнения и создания ROC-кривых.

“Работники” Tasks хранят все отчеты в хранилище Image Store **(T7)**. “Работники” Tasks также хранят и получают кластеры из Image Store.

“Работники” Tasks отправляют запрос для выполнения дополнительного извлечения в сервис Handler **(T9)**.