Инструкция по доступу к экземпляру программного обеспечения «Content Monitoring»

1. ВВЕДЕНИЕ

Программа «Content Monitoring» представляет собой веб-сервис для поиска в сети Интернет страниц, на которых без разрешения используются фотографии правообладателя. Пользователь загружает свои фотографии в кабинет, после чего система сразу и далее каждый час ищет их копии через поисковые сервисы Google и Яндекс. Найденные страницы пользователь отмечает как разрешённое использование или как нарушение и ведёт по нарушениям заметки; о новых находках сервис сообщает в Telegram. В кабинете может работать команда сотрудников, доступ к функциям определяется тарифом, оплачиваемым с баланса кабинета. Для администратора платформы предусмотрена административная панель для управления пользователями, тарифами, поисковыми сервисами и мониторинга работы сервера.

2. ДОСТУП К ЭКЗЕМПЛЯРУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Правообладатель готов продемонстрировать экземпляр программного обеспечения, для чего просим связаться с нами:

телефон: +74997045052

почта: info@content-monitoring.ru

2. После успешной авторизации открывается доступ к командной оболочке (CLI) операционной системы сервера. Через неё доступны каталог приложения `/opt/logo`, конфигурационные файлы и Docker-контейнеры, составляющие единый программный комплекс.

3. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В основе продукта лежат 2 основные составные части:

1) Клиентская часть ПО — это интерфейс для взаимодействия с пользователем. Он включает в себя веб-страницы, CSS-стили, JavaScript-скрипты, а также библиотеки и фреймворки, используемые для создания пользовательского интерфейса: публичный сайт, кабинет пользователя и административную панель.

2) Серверная часть ПО, которая в свою очередь подразделяется на две части:

— API-сервер бизнес-логики;

— Система хранения данных.

Серверная часть продукта отвечает за логику работы бизнес-процессов и взаимодействие с реляционной базой данных. Для разработки использовался язык Python 3.12 и фреймворк FastAPI; приложение разделено на функциональные модули:

- модуль аутентификации (регистрация, вход, двухфакторная аутентификация, сессии);
- модуль пользователей и администрирования;
- модуль папок и фотографий;
- модуль поиска (Google и Яндекс, очередь и почасовой мониторинг);
- модуль находок, пометок и заметок;
- модуль команды кабинета;
- модуль уведомлений Telegram;
- модуль финансов и тарифов;
- модуль мониторинга (журналы, метрики сервера, блокировка IP).

Серверное приложение взаимодействует с СУБД PostgreSQL 16. Файлы фотографий и резервные копии базы данных хранятся в приватном S3-хранилище.

Взаимодействие между этими частями осуществляется на основе клиент-серверной архитектуры (рисунок 1).

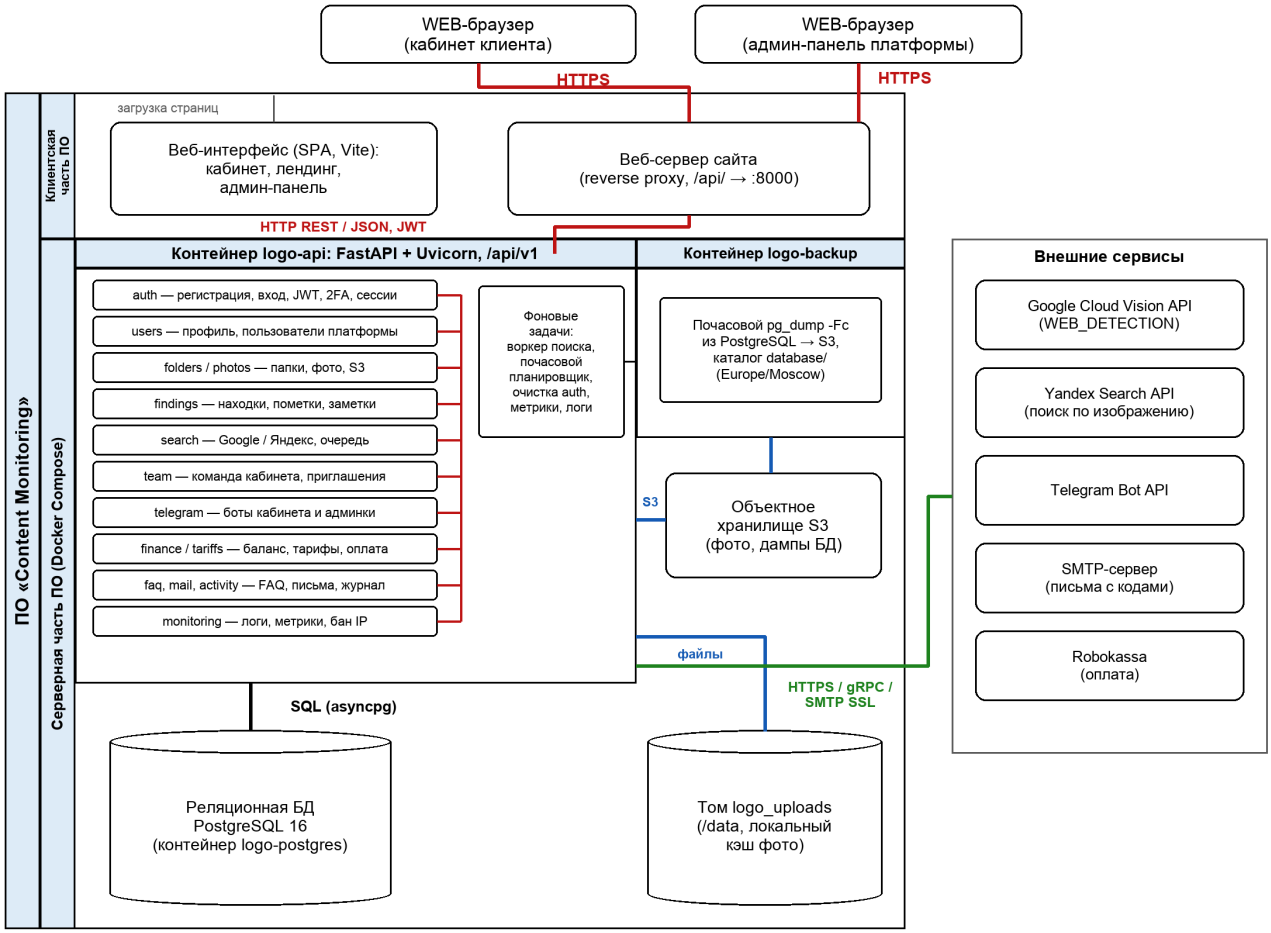


Рисунок 1 — Архитектура программного обеспечения

Взаимодействие между клиентской частью и API-сервером происходит по протоколу HTTP(S), данные передаются в формате JSON. API-сервер взаимодействует с базой данных PostgreSQL, с S3-хранилищем — по протоколу S3 (HTTPS), с внешними сервисами Google Cloud Vision, Yandex Search API, Telegram и почтовым сервером — по HTTPS и SMTP.

Путь /opt/logo – корневой каталог приложения, в котором размещены файлы приложения.

По пути /opt/logo хранятся инструменты, которые отвечают за сборку инфраструктуры в Docker-контейнеры:

- Файл docker-compose.yml отвечает за сборку и запуск всех контейнеров приложения (база данных, API, резервное копирование). Файл конфигурации находится по пути /opt/logo/backend/.env;
- Файл Dockerfile — сборка образа приложения;
- По пути /opt/logo/backend находится исходный код серверной части, миграции базы данных и конфигурация;
- В каталоге /opt/logo/workers/backup хранится сервис почасового резервного копирования базы данных.

4. ВНУТРЕННИЕ МОДУЛИ СЕРВЕРНОЙ ЧАСТИ

Аутентификация и безопасность: регистрация, авторизация, восстановление доступа, двухфакторная аутентификация, пользовательские сессии.

Пользователи и администрирование: управление кабинетами, сотрудниками, ролями, ограничениями и тарифами.

Папки и фотографии: загрузка файлов и изображений по ссылке, хранение метаданных, организация фотографий в папках, разрешённые адреса.

Поиск: постановка фотографий в очередь, запуск первичных и периодических проверок, обращения к Google и Яндексу по API.

Находки: учёт интернет-страниц, фильтрация, статусы «Не проверено», «В работе», «Разрешено», ведение заметок.

Уведомления: информирование о новых результатах поиска через Telegram.

Финансы и тарифы: баланс кабинета, история начислений и списаний, пополнение баланса через подключённого платёжного провайдера, применение тарифных ограничений.

Мониторинг платформы: показатели сервера, журналы приложений и HTTP-запросов, контроль подозрительной активности и IP-блокировки.

5. ИНТЕГРАЦИЯ С ПОИСКОВЫМИ СЕРВИСАМИ ЧЕРЕЗ API

5.1. Два внешних поисковых канала

Content Monitoring обращается по HTTPS к двум внешним API: Google Cloud Vision API (Web Detection) и Image Search API Яндекса (Yandex Cloud Web Search API, метод ImageSearch.SearchByImage). Сервисы возвращают результаты поиска и не входят в исходный код веб-сервиса.

В административной панели каждый поисковый канал включается отдельно. При недоступности Google поиск через Яндекс продолжается, если его API доступен и канал включён. Результаты провайдеров могут различаться.

5.2. Обработка поисковых заданий и находок

После загрузки фотографии система сохраняет файл в приватном хранилище, а сведения о фотографии — в базе данных.

Фотография включается в очередь первичного поиска сразу после загрузки; после этого автоматически запускается ежечасная проверка.

Серверная подсистема обращается к включённым и доступным API Google и Яндекса и получает результаты поиска, предоставляемые каждым из этих сервисов.

Находки учитываются отдельно для каждой фотографии. Один URL страницы не создаёт повторной находки, даже если адрес найден обоими поисковыми сервисами.

Новая находка получает статус «Не проверено», если для неё не задано разрешённое использование. Адреса из списка разрешённых автоматически отмечаются как разрешённые.

Пользователь проверяет находку, меняет статус и добавляет заметки. Информация о находке и выполненных действиях сохраняется в системе.

Сопоставление изображений и обнаружение страниц выполняются с участием внешних поисковых API. Программа при этом реализует собственный процесс управления заданиями, учёта результатов, устранения повторов по URL и пользовательской классификации. Документ не приписывает системе собственные алгоритмы компьютерного зрения, не подтверждённые техническими материалами.

6. ХРАНЕНИЕ ДАННЫХ И БЕЗОПАСНОСТЬ

PostgreSQL 16 хранит структурированные сведения о пользователях, фотографиях, поисковых заданиях, найденных страницах, пометках и заметках. Исходные файлы фотографий и резервные копии базы данных размещаются в приватном S3-совместимом хранилище на российской инфраструктуре. Доступ к хранилищу осуществляет серверная часть по S3 API через HTTPS.

Доступ пользователей к функциям определяется ролью (владелец кабинета, сотрудник, супер-администратор). Предусмотрены механизмы двухфакторной аутентификации и управления сессиями. Работа с пользовательским интерфейсом и API осуществляется по HTTPS.

Отсутствие связи с отдельным внешним сервисом поиска не препятствует работе остальных компонентов кабинета. Доступность поиска через конкретный сервис зависит от работоспособности его API и действующих условий использования.

7. РАЗВЁРТЫВАНИЕ И ДОСТУП К ЭКЗЕМПЛЯРУ

Content Monitoring доступен пользователям как веб-сервис по адресу https://content-monitoring.ru. Для работы необходим браузер и доступ в Интернет, установка клиентского приложения не требуется. Серверные компоненты развёрнуты в Docker-контейнерах; физическое размещение файлов фотографий обеспечивает российское S3-совместимое хранилище.

Корневой каталог приложения на сервере — /opt/logo. В нём размещены элементы конфигурации и развёртывания, в том числе следующие:

/opt/logo/backend — исходный код серверной части, миграции БД и конфигурация;
/opt/logo/backend/.env — файл конфигурационных параметров;
docker-compose.yml — конфигурация контейнеров приложения, базы данных и резервного копирования;
Dockerfile — инструкция сборки контейнерного образа;
/opt/logo/workers/backup — сервис резервного копирования базы данных.
Для технической экспертизы правообладатель может предоставить согласованный доступ к проверочному экземпляру и серверной среде, включая просмотр исходного кода и конфигурации. Адрес сервера, порт и персональные учётные данные передаются отдельно по защищённому каналу после согласования условий доступа. Пароли и токены внешних сервисов в настоящий документ не включаются.

8. СТОРОННИЕ БИБЛИОТЕКИ, ИНФРАСТРУКТУРА И ВНЕШНИЕ API

Версии и лицензии библиотек приведены по исходному техническому перечню компонентов. Их соответствие фактической сборке проверяется по составу развёрнутого приложения. Лицензии библиотек не следует смешивать с условиями использования внешних API, которые определяются отдельными правилами провайдеров.

Название стороннего ПО	Название лицензии	Ссылка на лицензию	Лицензия платная?
Python 3.12	PSF License 2.0	https://docs.python.org/3/license.html	Нет
PostgreSQL 16	PostgreSQL License (аналог MIT/BSD)	https://www.postgresql.org/about/licence/	Нет
Docker Engine, Docker Compose	Apache-2.0	https://www.apache.org/licenses/LICENSE-2.0	Нет

Название стороннего ПО	Название лицензии	Ссылка на лицензию	Лицензия платная?
FastAPI 0.141.1	MIT	https://github.com/fastapi/fastapi/blob/master/LICENSE	Нет
Uvicorn 0.52.4	BSD-3-Clause	https://github.com/encode/uvicorn/blob/master/LICENSE.md	Нет
SQLAlchemy 2.0.52	MIT	https://github.com/sqlalchemy/sqlalchemy/blob/main/LICENSE	Нет
Alembic 1.19.1	MIT	https://github.com/sqlalchemy/alembic/blob/main/LICENSE	Нет
Pydantic 2.13.5	MIT	https://github.com/pydantic/pydantic/blob/main/LICENSE	Нет
Pillow 12.3.0	MIT-CMU (HPND)	https://github.com/python-pillow/Pillow/blob/main/LICENSE	Нет
boto3 1.43.96	Apache-2.0	https://github.com/boto/boto3/blob/develop/LICENSE	Нет
google-cloud-vision 3.15.0	Apache-2.0	https://github.com/googleapis/google-cloud-python/blob/main/LICENSE	Нет

8.1. Внешние сервисы

Google Cloud Vision API (Web Detection) — внешний API, позволяющий получать сведения о совпадающих изображениях и интернет-страницах, на которых они обнаружены.

Image Search API Яндекса (Yandex Cloud Web Search API, метод ImageSearch.SearchByImage) — внешний API обратного поиска по изображению; канал работает независимо от Google.

Российский провайдер S3-совместимого хранилища — инфраструктурный сервис для изображений и резервных копий; конкретный провайдер и регион размещения подтверждаются эксплуатационными документами.

Telegram API и SMTP — внешние каналы уведомлений и электронных сообщений; они не используются для поиска фотографий.

Платёжная интеграция — взаимодействие веб-сервиса с подключённым внешним платёжным провайдером для пополнения баланса кабинета и получения сведений о результате оплаты. Выбор провайдера определяется текущей конфигурацией системы.