Участники (Room 2):

1. Барсуков P3415
2. Конкин P3410
3. Мальков P3410
4. Кобелев P3412
5. Шорников P3411
6. Ляшенко P3409
7. Аскаров P3410
8. Таранов P3419
9. Эйдельман P3414
10. Кремпольская P3420

Название системы: Облачное файловое хранилище (CloudDrive)

Шаг 1: определяем нашу систему

Предоставлять пользователям возможность загружать, хранить, скачивать, управлять и делиться файлами через веб-интерфейс или API.

Шаг 2: определяем акторов

Актор	Роль
Пользователь (End User)	Загружает, скачивает, просматривает, удаляет файлы. Может создавать папки, делиться файлами.
Администратор (Admin)	Управляет учетными записями, квотами, мониторингом, резервным копированием.
Внешний сервис авторизации (OAuth Provider)	Например, Google, GitHub — для входа пользователей.
Внешний сервис оплаты (Payment Gateway)	Если есть платные тарифы (Stripe, PayPal).
Система мониторинга / логирования (Monitoring System)	Собирает метрики и логи (например, Prometheus + Grafana).

Пользователь	Роль	Уровень доверия	Почему?
Конечный пользователь (End User)	Загружает, скачивает, делится файлами	Частичное доверие	Мы доверяем ему вводить свои данные, но не доверяем хранить чужие файлы без контроля. Нужна аутентификация и авторизация.
Администратор системы	Управляет пользователями, квотами, мониторингом	Высокое доверие	Имеет доступ к конфигурации, логам, данным. Должен быть строго ограничен (MFA, RBAC).
Внешние сервисы (OAuth, Payment)	Обеспечивают вход/платежи	Низкое доверие	Мы не контролируем их. Взаимодействие через API с токенами и подписями.

Шаг 3: определяем внешние системы

Внешняя система	Назначение
OAuth-провайдер (Google, Microsoft, etc.)	Аутентификация и авторизация пользователей.
Шлюз платежей (Stripe, YooKassa)	Обработка подписок и оплат.
Облачное хранилище объектов (AWS S3, Google Cloud Storage, MinIO)	Хранение самих файлов.
База данных (PostgreSQL, MySQL)	Хранение метаданных: пользователи, файлы, папки, права доступа.
Сервис отправки email (SendGrid, SMTP)	Уведомления о действиях (например, приглашения к файлам).
CDN (Cloudflare, Akamai)	Ускорение доставки файлов конечным пользователям.

Внешняя система	Назначение	Протокол / Механизм аутентификации	Безопасность
OAuth-провайдер (Google, GitHub)	Аутентификация пользователей	OAuth 2.0 / OpenID Connect	Токены JWT, HTTPS, PKCE для SPA
Payment Gateway (Stripe, YooKassa)	Обработка платежей	API Key + Signature / OAuth	Секретные ключи в переменных окружения, HTTPS, IP-whitelist
Object Storage (S3, MinIO)	Хранение файлов	AWS IAM / Access Key + Secret	Подписанные URL (presigned URLs), временные токены STS
Email Service (SendGrid)	Отправка уведомлений	API Key	Ограниченные права, HTTPS
CDN (Cloudflare)	Доставка контента	Token / Origin Pull	Защита от DDoS, WAF
Monitoring System (Prometheus)	Сбор метрик	Basic Auth / TLS Certificates	Изолированный сетевой доступ, TLS

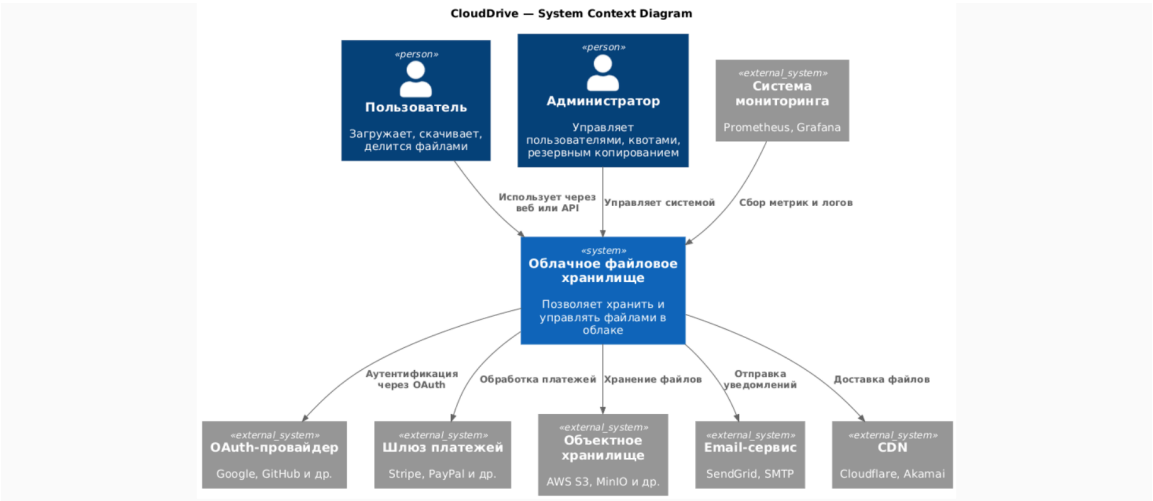
Все внешние интеграции должны использовать HTTPS, токены/ключи с минимальными привилегиями и временные учетные данные, где возможно

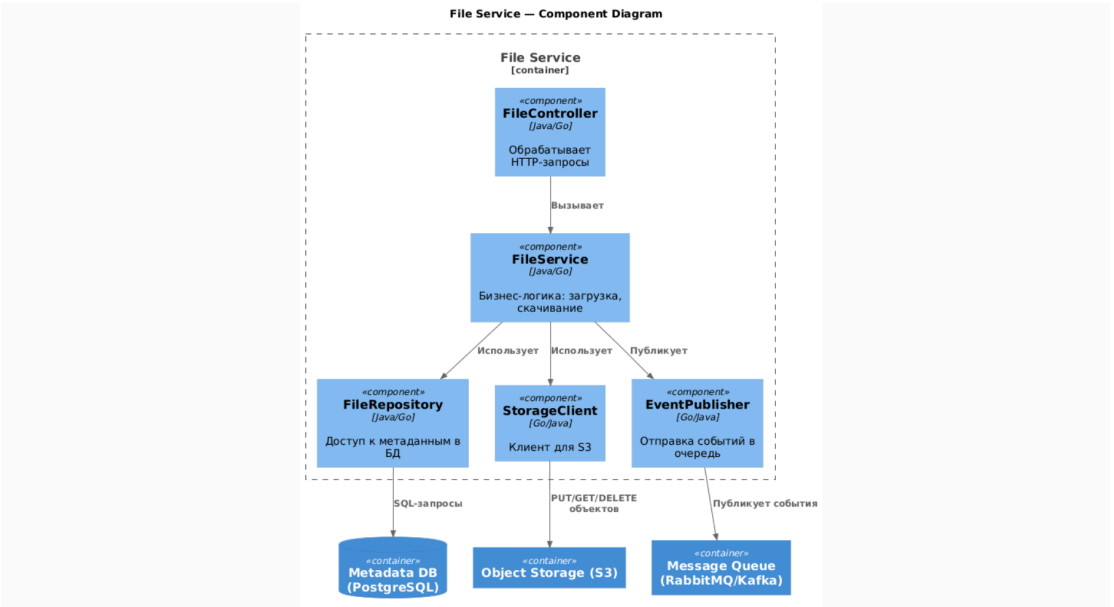
Контейнер → Контейнер	Протокол	Формат данных	Почему?
WebApp → API Gateway	HTTP/HTTPS	JSON (REST) или GraphQL	Стандартный веб-интерфейс, простой для фронтенда
API Gateway → Auth Service	gRPC / HTTP	JSON / Protobuf	Быстрая и надёжная внутренняя связь
API Gateway → File Service / User Service	HTTP/HTTPS	JSON	Гибкость, поддержка REST
File Service → Metadata DB	TCP/IP (PostgreSQL)	SQL	Надёжное хранение структурированных данных
File Service → Message Queue	AMQP / Kafka Protocol	JSON	Асинхронная обработка задач (загрузка, резервное копирование)

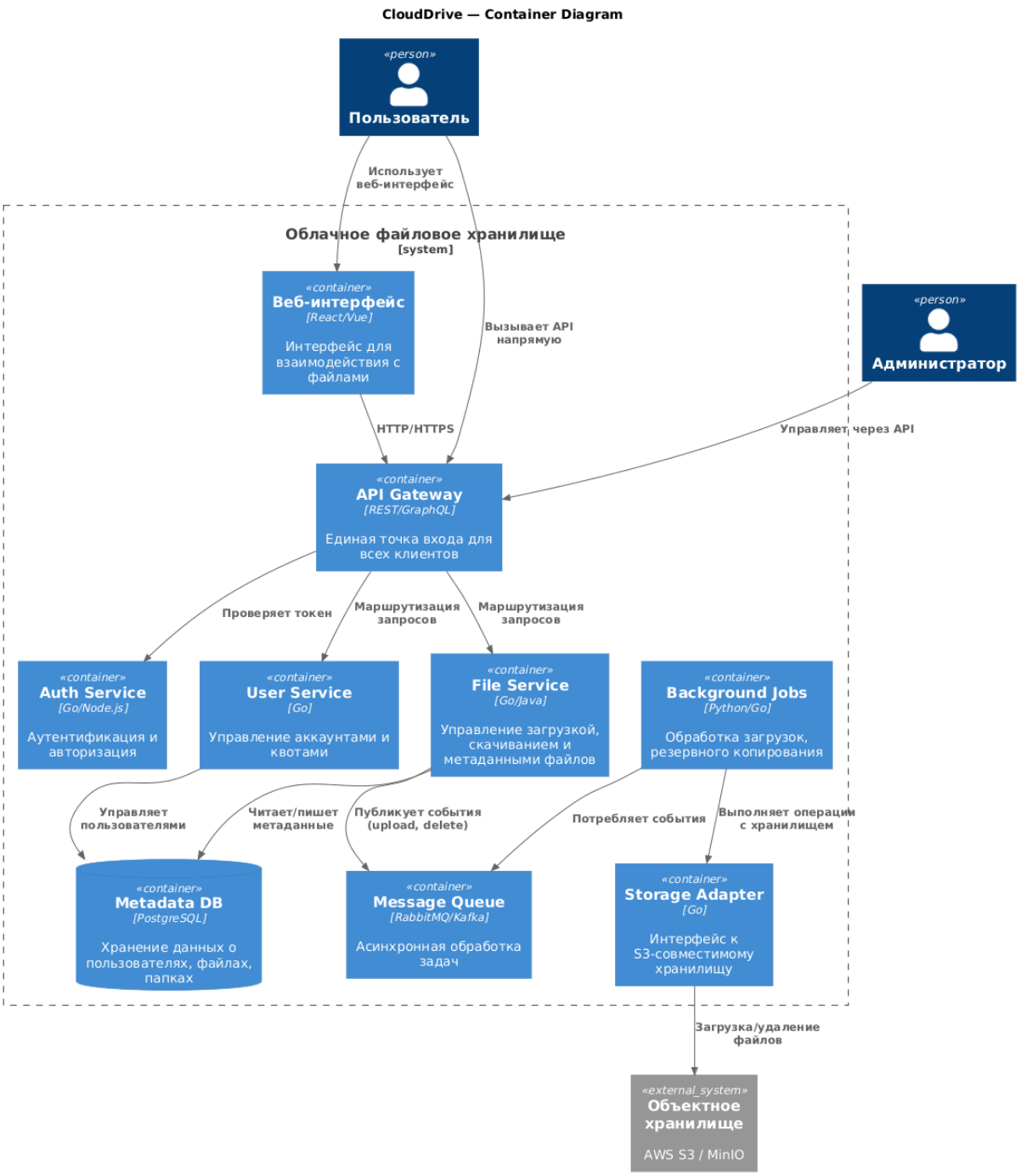
Background Jobs → Storage Adapter	HTTP/HTTPS	S3 API	Совместимость с облачными хранилищами
Storage Adapter → Object Storage	S3 API	Binary / JSON	Стандартный протокол для объектного хранилища

Уровень	Что валидируется	Как
Frontend (WebApp)	Размер файла, тип MIME, расширение	JavaScript:file.type,file.name.endsWith('.jpg'), ограничение по размеру (например, 50 МБ)
API Gateway	Заголовки, токен авторизации, формат запроса	Проверка JWT, Content-Type, Content-Length, CORS
File Service (Controller/Service Layer)	Расширение, MIME-тип, сигнатура файла, антивирусный скан	Проверка черезmagic number(первые байты), вызов антивируса (ClamAV), блокировка.exe,.bat,.sh

Шаг 4: определяем взаимодействия






CloudDrive — Container Diagram
«person»
Пользователь
Использует веб-интерфейс
Облачное файловое хранилище
[system]
«container»
Веб-интерфейс
[React/Vue]
Интерфейс для взаимодействия с файлами
Вызывает API напрямую
«person»
Администратор
HTTP/HTTPS
Управляет через API
«container»
API Gateway
[REST/GraphQL]
Единая точка входа для всех клиентов
Проверяет токен
Маршрутизация запросов
Маршрутизация запросов
«container»
Auth Service
[Go/Node.js]
Аутентификация и авторизация
«container»
User Service
[Go]
Управление аккаунтами и квотами
«container»
File Service
[Go/Java]
Управление загрузкой, скачиванием и метаданными файлов
«container»
Background Jobs
[Python/Go]
Обработка загрузок, резервного копирования
Управляет пользователями
Читает/пишет метаданные
Публикует события (upload, delete)
Потребляет события
Выполняет операции с хранилищем
«container»
Metadata DB
[PostgreSQL]
Хранение данных о пользователях, файлах, папках
«container»
Message Queue
[RabbitMQ/Kafka]
Асинхронная обработка задач
«container»
Storage Adapter
[Go]
Интерфейс к S3-совместимому хранилищу
Загрузка/удаление файлов
«external_system»
Объектное хранилище
AWS S3 / MinIO

Задание №2 Анализ Угроз

Угрозы Аутентификации и Авторизации

Кража или подделка JWT токенов

Если токены доступа хранятся в localStorage браузера, они уязвимы для XSS-атак. Если время жизни токена слишком велико, злоумышленник получит длительный доступ.

Отсутствие механизма ротации Refresh-токенов или слабый алгоритм подписи JWT.

Insecure Direct Object Reference

Пользователь А меняет ID файла в URL (например, /files/123 -> /files/124) и получает доступ к файлу Пользователя Б.

Отсутствие проверки принадлежности файла конкретному пользователю на уровне Backend (File Service).

Угрозы при работе с Файлами

Zip Bombs

Загрузка архива, который при распаковке (например, антивирусом ClamAV) занимает терабайты места или вешает CPU, вызывая отказ в обслуживании всей системы

Загрузка вредоносного ПО

Валидация на фронтенде легко обходится. Если загруженный файл будет сохранен в директорию, где веб-сервер может его исполнить, это приведет к удаленному выполнению кода.

Угрозы Хранения и Конфиденциальности Данных

Публичный доступ к S3 Bucket

Неправильная конфигурация прав доступа S3, позволяющая сканировать и скачивать все файлы хранилища в обход приложения.

Отсутствие шифрования данных

Если злоумышленник получит доступ к физическим дискам или S3, он прочитает файлы. Файлы должны шифроваться на сервере.

Утечка через Presigned URLs.

Для скачивания используются подписанные URL (S3 Presigned URLs). Если срок действия ссылки слишком большой и ссылка утечет, любой в интернете сможет скачать файл

Угрозы Внешних Интеграций и Инфраструктуры

Утечка API ключей (Secrets Management)

Если ключи захардкожены в коде или попадают в систему контроля версий.

Атака на бюджет (использование квоты отправки писем для спама), доступ к данным, проведение фальшивых возвратов.

SSRF

Злоумышленник укажет адрес внутренней сети (например, http://localhost:9090/metrics или метаданные облака), чтобы получить доступ к внутренней инфраструктуре. Сервис File Service выполняет запрос и возвращает "содержимое файла", которое на самом деле является временными ключами администратора сервера. В итоге произойдет полный захват облачной инфраструктуры.