Диск VK WorkSpace

Диск VK WorkSpace

© VK WorkSpace, 2026 г. Все права защищены

Портал с документацией: https://biz.mail.ru/docs/on-premises/

Оглавление

Потоки данных при загрузке файлов 3

Потоки данных при чтении файла из домашней директории 6

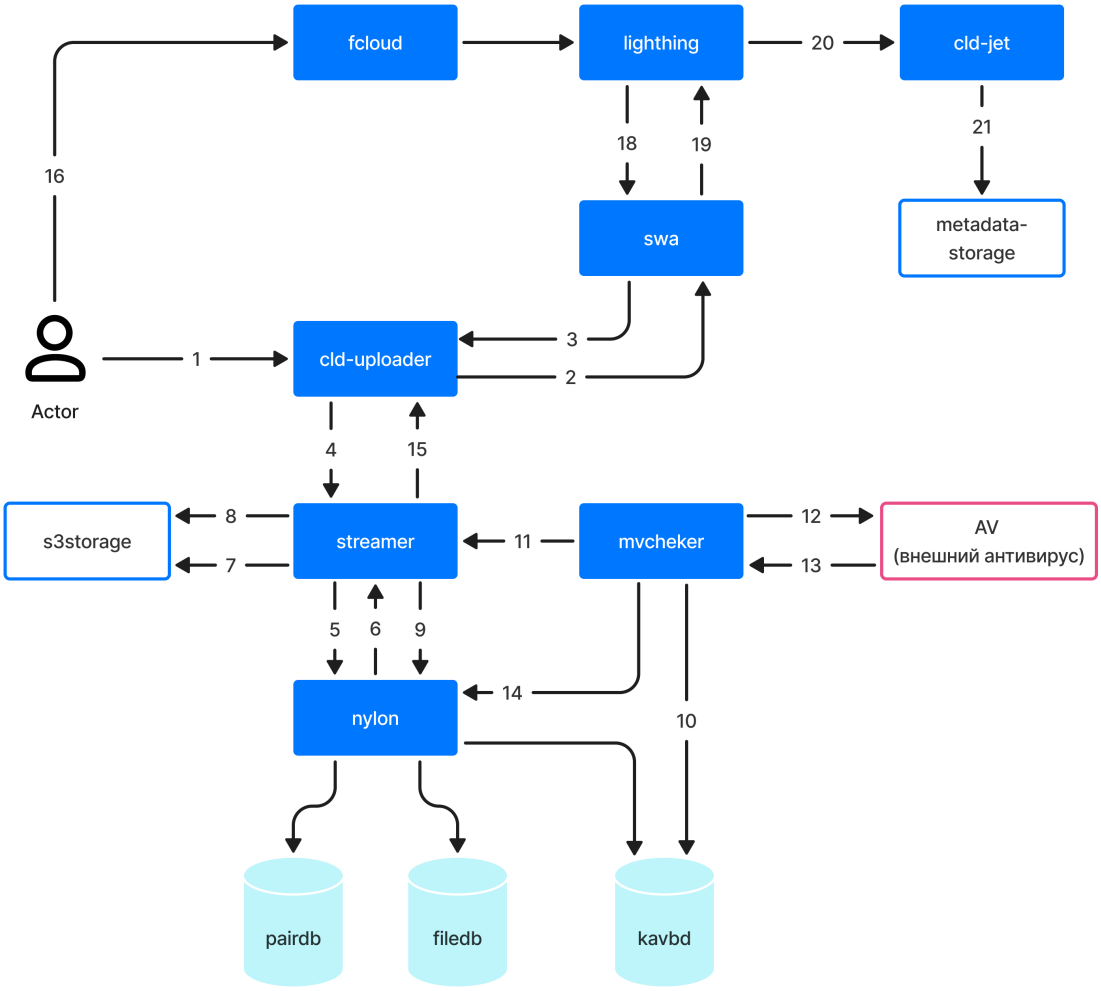
Потоки данных при чтении файла по ссылке 8

Потоки данных при чтении архива 11

Потоки данных при удалении файла 13

Потоки данных при загрузке файлов

Схема взаимодействия сервисов приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

№ потока	Источник	Протокол взаимодействия	Получатель
1	Actor	HTTP	cld-uploader — сервис для загрузки файлов в веб-интерфейс
2	cld-uploader	HTTP	swa — группа сервисов аутентификации (clauth, GoCube, UMI) для выдачи, хранения и валидации сессионных токенов для пользователей
3	swa	HTTP	cld-uploader

№ потока	Источник	Протокол взаимодействия	Получатель
4	cld-uploader	HTTP	streamer — сервис для сохранения и скачивания файлов хранилища
5	streamer	Бинарный протокол Tarantool	nylon — прокси-сервис
6	nylon	Бинарный протокол Tarantool	streamer
7	streamer	HTTP	s3storage — основное хранилище файлов Диска
8	streamer	HTTP	s3storage
9	streamer	Бинарный протокол Tarantool	nylon
10	cld-kavdb	Бинарный протокол Tarantool	mvchecker — сервис интеграции с антивирусом по протоколу ICAP
11	streamer	HTTP	mvchecker
12	mvchecker	ICAP	AV (внешний антивирус)
13	AV (внешний антивирус)	ICAP	mvchecker
14	mvchecker	Бинарный протокол Tarantool	nylon — прокси-сервис
15	streamer	HTTP	cld-uploader
16	Actor	HTTP	fcloud — роутер API-запросов и хранилище статики Диска VK WorkSpace
17	fcloud	HTTP	lightning — веб-API для пользователей
18	lightning	HTTP	swa

№ потока	Источник	Протокол взаимодействия	Получатель
19	swa	HTTP	lightning
20	lightning	HTTP	cld-jet — API для доступа к хранилищу структур файловых хранилищ пользователей
21	cld-jet	HTTP	metadata-storage — группа сервисов для хранения и управления метаданными файлов пользователей

Примечание

При отсутствии взаимодействия с внешним антивирусом потоки с 10 по 14 исключены из схемы взаимодействия.

Последовательность взаимодейтсвия сервисов при загрузке файла:

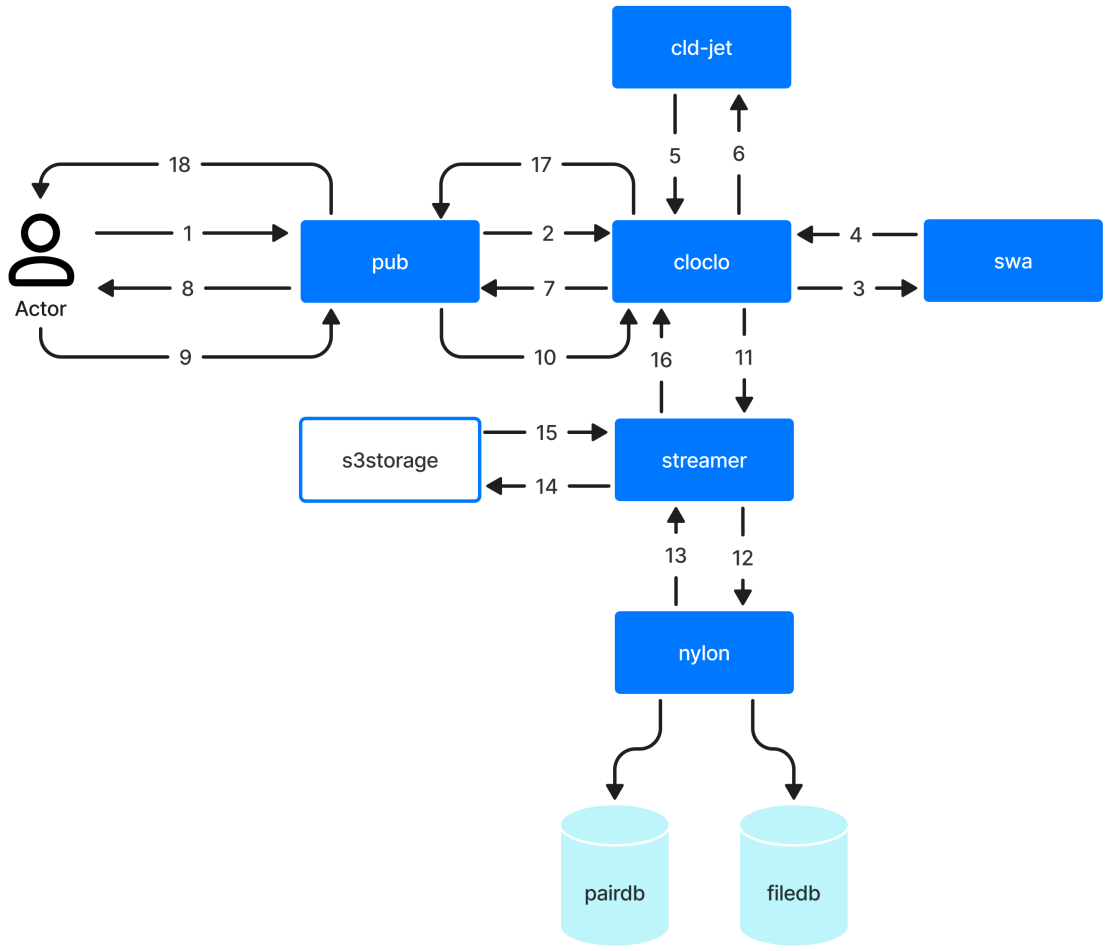
1. Клиент при загрузке файла рассчитывает хэш на своей стороне и передает его в **cld-uploader**. Сервис **cld-uploader** проверяет авторизацию пользователя в **swa** и загружает файл на свой локальный диск и рассчитывает хэш.
2. Если хэши не сходятся, **cld-uploader** вернет ошибку. Если валидация пройдена,бинарный файл отправляется в сервис **streamer**, который формирует запрос в **nylon** по протоколу Tarantool.
3. Сервис **nylon** формирует в **mpairdb** запрос на поиск свободной дисковой пары на запись файла. После чего возвращает в **streamer** два ID, на которые и начинает загрузку на указанную дисковую пару.
4. Далее выполняется запись во временную директорию в **s3storage**, затем перемещается в нужную директорию (по хэшу).
5. По завершении загрузки бинарного файла **streamer** передает в **nylon** информацию — на какой паре лежит файл с определенны хэшом. Информация сохраняется в **mfiledb**. Обмен с базами данных осуществляется по протоколу Tarantool.
6. После чего информация о новом файле отправляется в **cld-kavdb**. Это необходимо для дальнейшей проверки файла на вирусы.
7. Сервис **mvcheker** с установленной периодичностью выполняет опрос сервиса **cld-kavdb** на наличие новых файлов.
8. В случае наличия новых файлов в **cld-kavdb mvcheker** отправляет запрос на получение файла в **streamer** и передает его на проверку сторонней антивирусной системе по протоколу **ICAP**.
9. Если найден вирус в отправленном файле, **mvcheker** отправляет файл в **nylon**.
10. Сервис **nylon** передает полученный файл с вирусом в **mfiledb** для установки соответствующих флагов.

11. Далее выполняется обновление пользовательского дерева и сохранение метаданных. Клиент отправляет информацию в сервис **fcloud** и далее в сервис **lightning**.
12. Через сервис **cld-jet** выполняется сохранение информации и обновление дерева пользователя в **metadata-storage**.

> **Примечание**
>
> Если размер файла меньше 20 байт, данные отправляются в **cld-jet** через **fcloud** и **lightning**, потоки данных до 15 включительно пропускаются.

Потоки данных при чтении файла из домашней директории

Схема взаимодействия сервисов приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

№ потока	Источник	Протокол взаимодействия	Получатель
1	Actor	HTTP	pub — сервис ввода HTTP(S)-трафика в сеть VK WorkSpace
2	pub	HTTP	cloclo — сервис по распределению и сбору информации для получения данных из Диска VK WorkSpace
3	cloclo	HTTP	swa — группа сервисов аутентификации (clauth, GoCube, UMI) для выдачи, хранения и валидации сессионных токенов для пользователей
4	swa	HTTP	cloclo
5	cloclo	HTTP	cld-jet — API для доступа к хранилищу структур файловых хранилищ пользователей
6	cld-jet	HTTP	cloclo
7	cloclo	HTTP	pub
8	pub	HTTP	Actor
9	Actor	HTTP	pub
10	pub	HTTP	cloclo
11	cloclo	HTTP	streamer — сервис для сохранения и скачивания файлов хранилища
12	streamer	Бинарный протокол Tarantool	nylon — прокси-сервис
13	nylon	Бинарный протокол Tarantool	streamer
14	streamer	HTTP	s3storage — основное хранилище файлов Диска

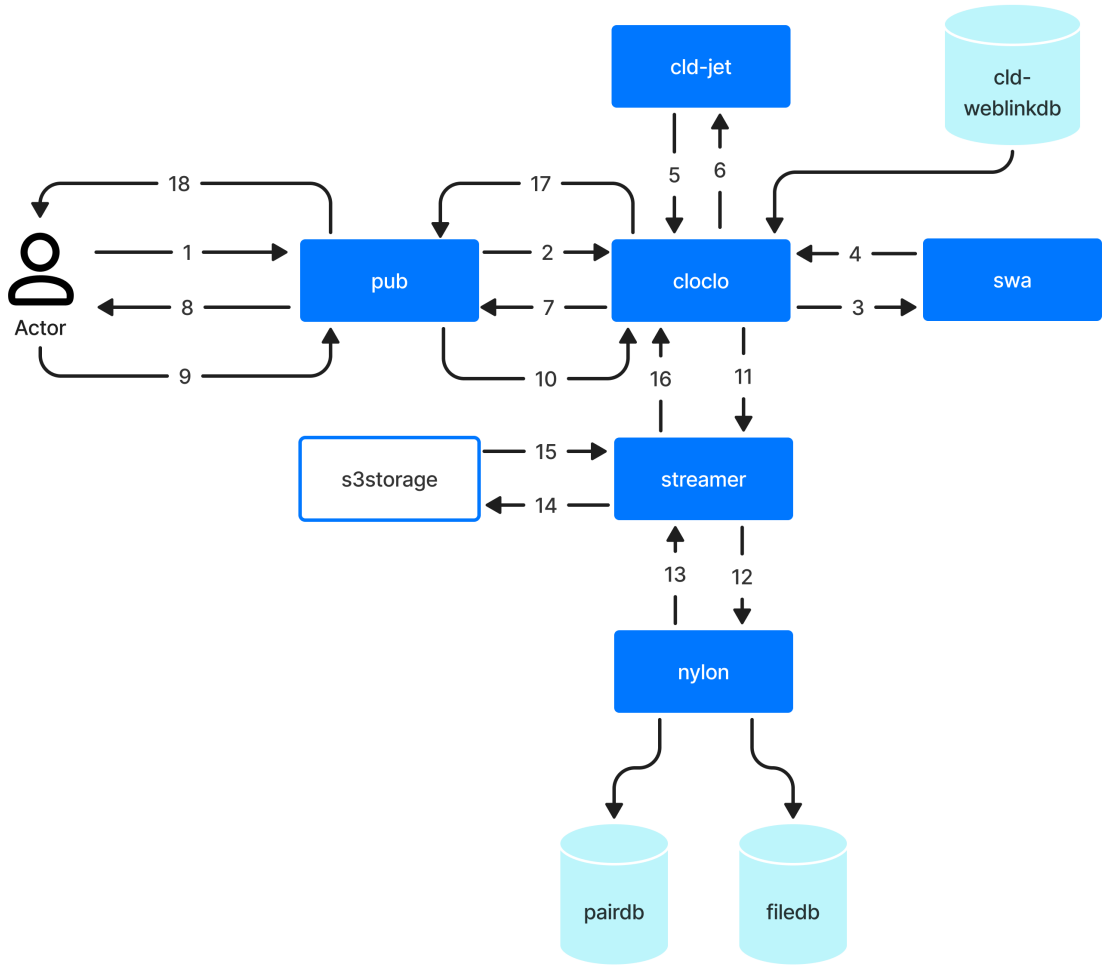
№ потока	Источник	Протокол взаимодействия	Получатель
15	s3storage	HTTP	streamer
16	streamer	HTTP	cloclo
17	cloclo	HTTP	pub
18	pub	HTTP	Actor

Последовательность взаимодейтсвия сервисов при чтении файла из домашней директории:

1. Клиент (пользователь) передает данные о пути в дереве метаданных и авторизационные данные в сервис **cloclo** через **pub**.
2. После подтверждения авторизации в **swa** путь до файла передается в сервис **cld-jet**.
3. В ответе на запрос **cld-jet** возвращает путь и хеш файла.
4. Сервис **cloclo** формирует ответ с кодом состояния HTTP 301. В редиректе клиенту возвращается токен, в котором содержится информация о хэше, пути и пользователе, который инициировал скачивание.
5. В ответ клиент возвращает токен в **cloclo**, где токен валидируется и извлекается хэш.
6. Внутри сервиса **cloclo** выполняется внутреннее перенаправление на указанный URI. Редирект отправляется в **streamer**, потом — в **nylon**.
7. Сервис **nylon** отправляет запрос в **mfiledb** для получения данных о местонахождении файла в **s3storage**.
8. Запрос возвращается в **streamer** и инициируется скачивание файлов из **s3storage**.
9. Затем бинарный файл через **streamer** отправляется клиенту.

Потоки данных при чтении файла по ссылке

Схема взаимодействия сервисов приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

№ потока	Источник	Протокол взаимодействия	Получатель
1	Actor	HTTP	pub — сервис ввода HTTP(S)-трафика в сеть VK WorkSpace
2	pub	HTTP	cloclo — сервис по распределению и сбору информации для получения данных из Диска VK WorkSpace
3	cloclo	HTTP	swa — группа сервисов аутентификации (clauth, GoCube, UMI) для выдачи, хранения и валидации сессионных токенов для пользователей
4	swa	HTTP	cloclo
5	cloclo	HTTP	cld-jet — API для доступа к хранилищу структур файловых хранилищ пользователей

№ потока	Источник	Протокол взаимодействия	Получатель
6	cld-jet	HTTP	cloclo
7	cloclo	HTTP	pub
8	pub	HTTP	Actor
9	Actor	HTTP	pub
10	pub	HTTP	cloclo
11	cloclo	HTTP	streamer — сервис для сохранения и скачивания файлов хранилища
12	streamer	Бинарный протокол Tarantool	nylon — прокси-сервис
13	nylon	Бинарный протокол Tarantool	streamer — сервис для сохранения и скачивания файлов хранилища
14	streamer	HTTP	s3storage — основное хранилище файлов Диска
15	s3storage	HTTP	streamer
16	streamer	HTTP	cloclo
17	cloclo	HTTP	pub
18	pub	HTTP	Actor

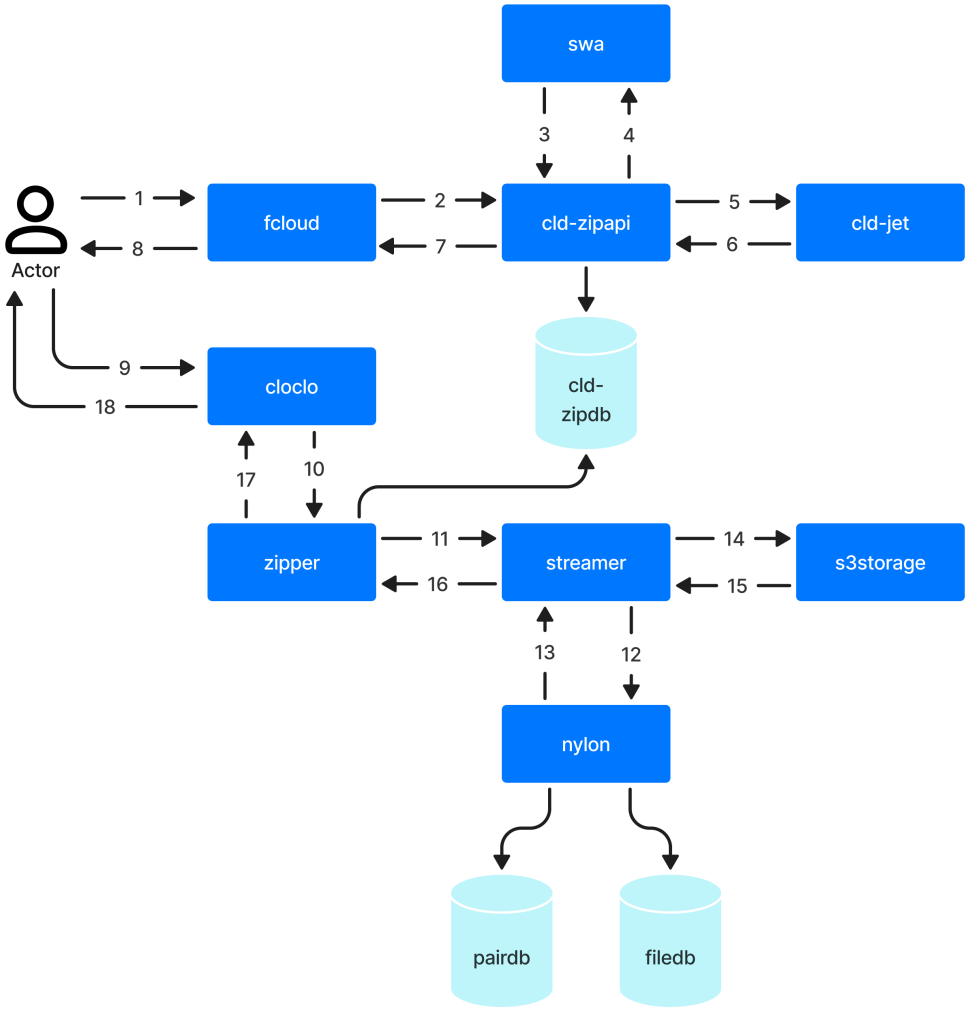
Последовательность взаимодейтсвия сервисов при чтении файла по ссылке:

1. При открытии ссылки пользователем выписывается токен.
2. Затем, при нажатии кнопки **Скачать**, клиент передает в сервис **cloclo** токен, путь к файлу внутри ссылки и авторизационные данные пользователя (при наличии).
3. После проверки авторизации из **cloclo** отправляется запрос с **weblink_id** (две последние части ссылки с доступом к файлу) в **cld-weblinkdb**.
4. Ответ с UID владельца ссылки и путем до файла возвращается в **cloclo** и перенаправляется в **cld-jet**.
5. Из **cld-jet** в **cloclo** возвращается путь до файла и хэш.

6. На основе этих данных **cloclo** генерирует токен, который передает клиенту с кодом состояния HTTP 301.
7. В ответ клиент возвращает токен в **cloclo**, где токен валидируется и извлекается хэш.
8. Внутри сервиса **cloclo** через **nginx** выполняется внутреннее перенаправление на указанный URI.
9. Сервис **nylon** отправляет запрос в **mfiledb** для получения данных о местонахождении файла в **s3storage**.
10. Запрос возвращается в **streamer** и инициируется скачивание файлов из **s3storage**.
11. Затем бинарный файл через **streamer** отправляется клиенту.

Потоки данных при чтении архива

Схема взаимодействия сервисов приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

№ потока	Источник	Протокол взаимодействия	Получатель
1	Actor	HTTP	

№ потока	Источник	Протокол взаимодействия	Получатель
			fcloud — роутер API-запросов и хранилище статики Диска VK WorkSpace
2	fcloud	HTTP	cld-zipapi — API для инициализации формирования zip-архивов
3	cld-zipapi	HTTP	swa — сервис, обслуживающий авторизационные запросы клиентов
4	swa	HTTP	cld-zipapi
5	cld-zipapi	HTTP	cld-jet — API для доступа к хранилищу структур файловых хранилищ пользователей
6	cld-jet	HTTP	cld-zipapi
7	cld-zipapi	HTTP	fcloud
8	fcloud	HTTP	Actor
9	Actor	HTTP	cloclo — сервис по распределению и сбору информации для получения данных из Диска VK WorkSpace
10	cloclo	HTTP	zipper — сервис для формирования и отдачи архивов конечным пользователям
11	zipper	HTTP	streamer — сервис для сохранения и скачивания файлов хранилища
12	streamer	Бинарный протокол Tarantool	nylon — прокси-сервис
13	nylon	Бинарный протокол Tarantool	streamer
14	streamer	HTTP	s3storage — основное хранилище файлов Диска

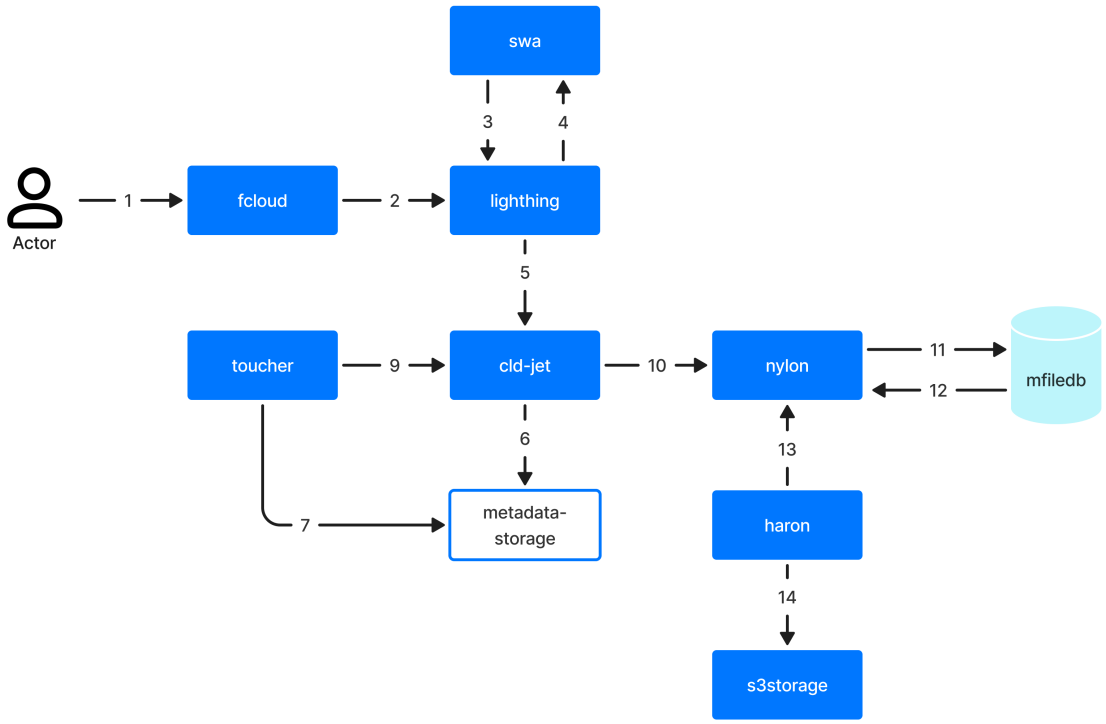
№ потока	Источник	Протокол взаимодействия	Получатель
15	s3storage	HTTP	streamer
16	streamer	HTTP	zipper
17	zipper	HTTP	cloclo
18	cloclo	HTTP	Actor

Последовательность взаимодействия сервисов при чтении файла из архива:

1. Путь до директории (или токен, если скачивание производится по ссылке) передается через **fcloud** в сервис **cld-zipapi**.
2. Сервис **cld-zipapi** передает запрос клиента в **swa** для подтверждения авторизации, после подтверждения авторизации сервис **cld-zipapi** формирует запрос в **cld-jet** для получения списка хэшей файлов в директории.
3. Сервис **cld-zipapi** преобразовывает список хэшей в токен и записывает соотношение токена и полученных хэшей в **cld-zipdb**, и передает токен клиенту через **fcloud**.
4. Вторым запросом от клиента токен передается в сервис **zipper** для получения по токену списка хешей в **cld-zipdb**.
5. Полученную информацию **zipper** отправляет в **streamer** с запросом выгрузки необходимого списка хэшей.
6. Сервис **streamer** отправляет запрос в **mfiledb** через сервис **nylon**, для получения местонахождения файлов в **s3storage** а затем выгружает их из нужных дисковых пар.
7. Формирование zip-архива выполняется налету и передается клиенту.

Потоки данных при удалении файла

Схема взаимодействия сервисов приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

№ потока	Источник	Протокол взаимодействия	Получатель
1	Actor	HTTP	fcloud — роутер API-запросов и хранилище статики Диска VK WorkSpace
2	fcloud	HTTP	lightning — веб-API для пользователей
3	lightning	HTTP	swa — сервис, обслуживающий авторизационные запросы клиентов
4	swa	HTTP	lightning
5	lightning	HTTP	cld-jet — API для доступа к хранилищу структур файловых хранилищ пользователей
6	cld-jet	HTTP	metadata-storage — группа сервисов для хранения и управления метаданными файлов пользователей
7	toucher	HTTP	metadata-storage
9	toucher	Бинарный протокол Tarantool	cld-jet

№ потока	Источник	Протокол взаимодействия	Получатель
10	cld-jet	Бинарный протокол Tarantool	nylon — прокси-сервис
11	nylon	HTTP	mfiledb — база данных для хранения таблицы связей постоянных вложений: файл — пара дисков
12	mfiledb	HTTP	nylon
13	haron	HTTP	nylon
14	haron	HTTP	s3storage — основное хранилище файлов

Последовательность удаления файла пользователем:

1. Клиент (пользователь) отправляет запрос на удаление файла
2. Сервис **fcloud** передает запрос на сервис **lightning** и отдает статические файлы клиенту.
3. Сервис **lightning** передает запрос клиента в **swa** для подтверждения авторизации, после подтверждения авторизации запрос на удаление передается на сервис **cld-jet**.
4. Далее сервис **cld-jet** передает запрос в **metadata-storage** для удаления из дерева метаднных.
5. **metadata-storage** удаляет всю информацию о файле — с этого момента пользователь не видит файл у себя в директории.

Последовательность удаления файла по истечении периода времени:

1. Сервис **toucher** в асинхронном режиме работы циклически по очереди запрашивает информацию о деревьях метаданных каждого пользователя в **metadata-storage** и обновляет **touch_time** по хэшам файлов, и выполняет запись данных в **mfiledb** через **nylon**.
2. Если файл перестает откликаться на запросы сервиса **toucher** — начинает увеличиваться **touch_time** файла.
3. Если **touch_time** достигает срока в полгода, запускается процесс удаления файла из **s3storage**.
4. Процесс удаления файла инициирует сервис **haron** по данным **touch_time**, полученным из **mfiledb** через **nylon**.
5. По истечении полугода в параметре **touch_time** сервис **haron** удаляет файл в **s3storage** и с дисковой пары.

27 апреля 2026 г.