Система управления данными в SPD Online Filter

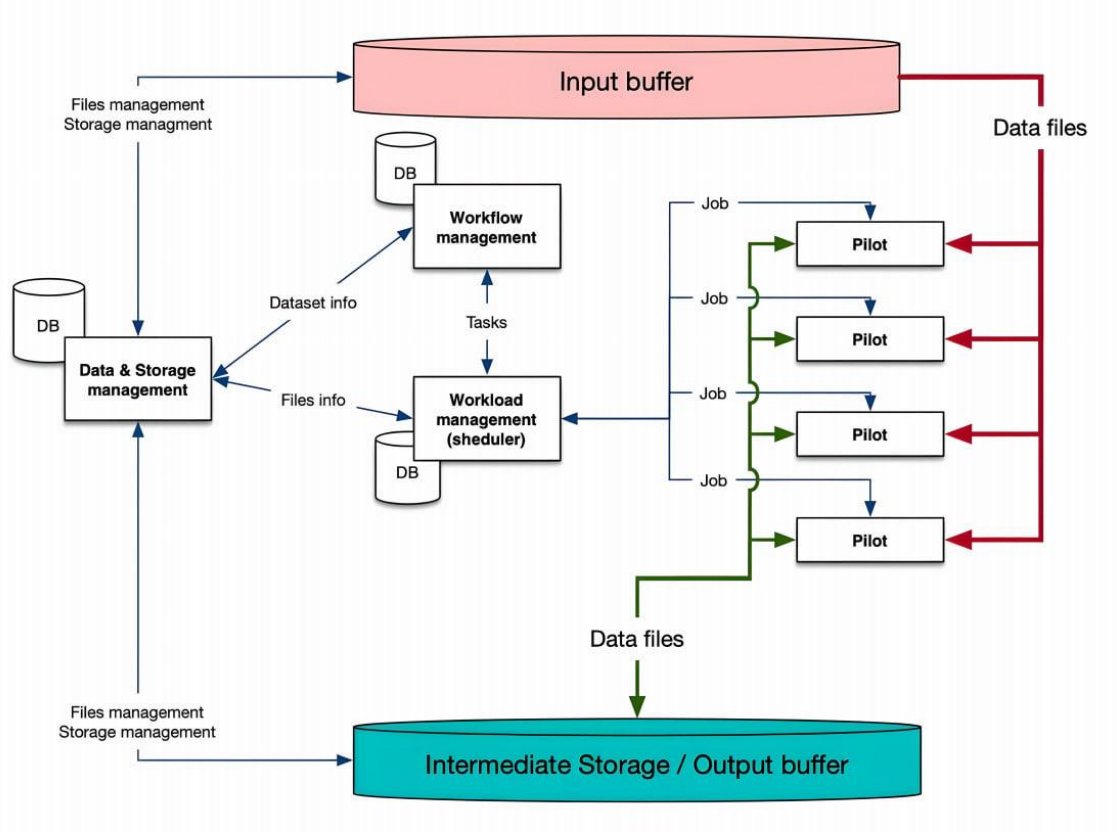
Студент	П. А. Коршунова
Научный руководитель	Е. Ю. Солдатов
Научный консультант	Д. А. Олейник

Online Filter

SPD Online Filter - это высокопроизводительная вычислительная система для высокопропускной обработки данных, собранных детектором SPD

Особенностью **высокопропускной обработки** данных является большой объем данных, как первичных, которые необходимо обработать, так и промежуточных, возникающих в процессе обработки

Цель: существенно сократить объем данных для долговременного хранения, последующей обработки и анализа



Архитектура SPD Online Filter

Промежуточное ПО

Система управления данными

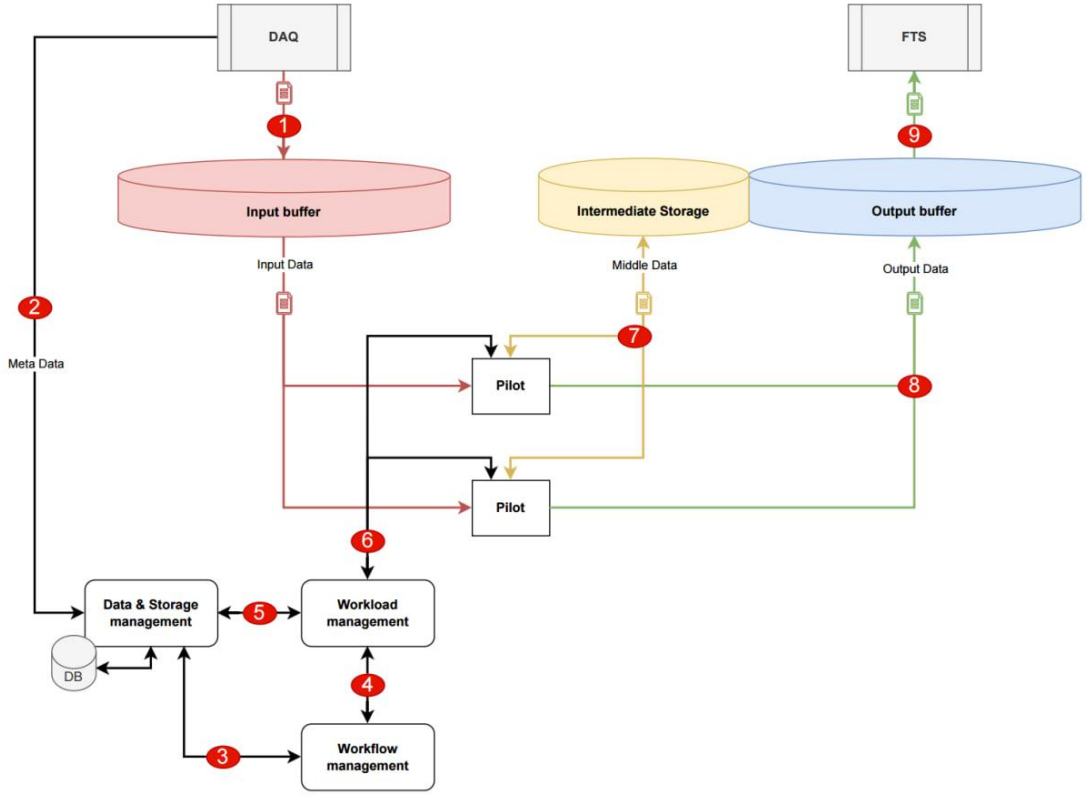
- регистрация, каталогизация, контроль целостности

Система управления процессами обработки

- формирование и контроль исполнения этапов обработки данных

Система управления нагрузкой

- реализация этапов обработки (формирование задач, отправка задач пилотам)



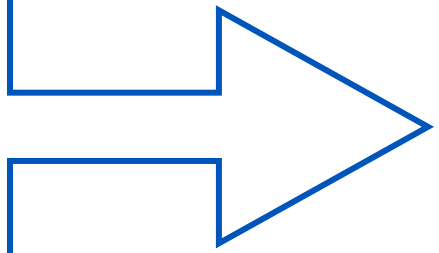
Pilot – приложение, работающее на вычислительном узле и исполняющее задачи

Зачем нужна система управления данными?

Большой поток данных с детектора
до 20 ГБ/сек

+

Большой объем промежуточных данных, полученных в процессе обработки



Необходимость разработки системы управления данными, которая должна обеспечивать контроль над организацией, хранением и целостностью данных

Система управления данными: задачи

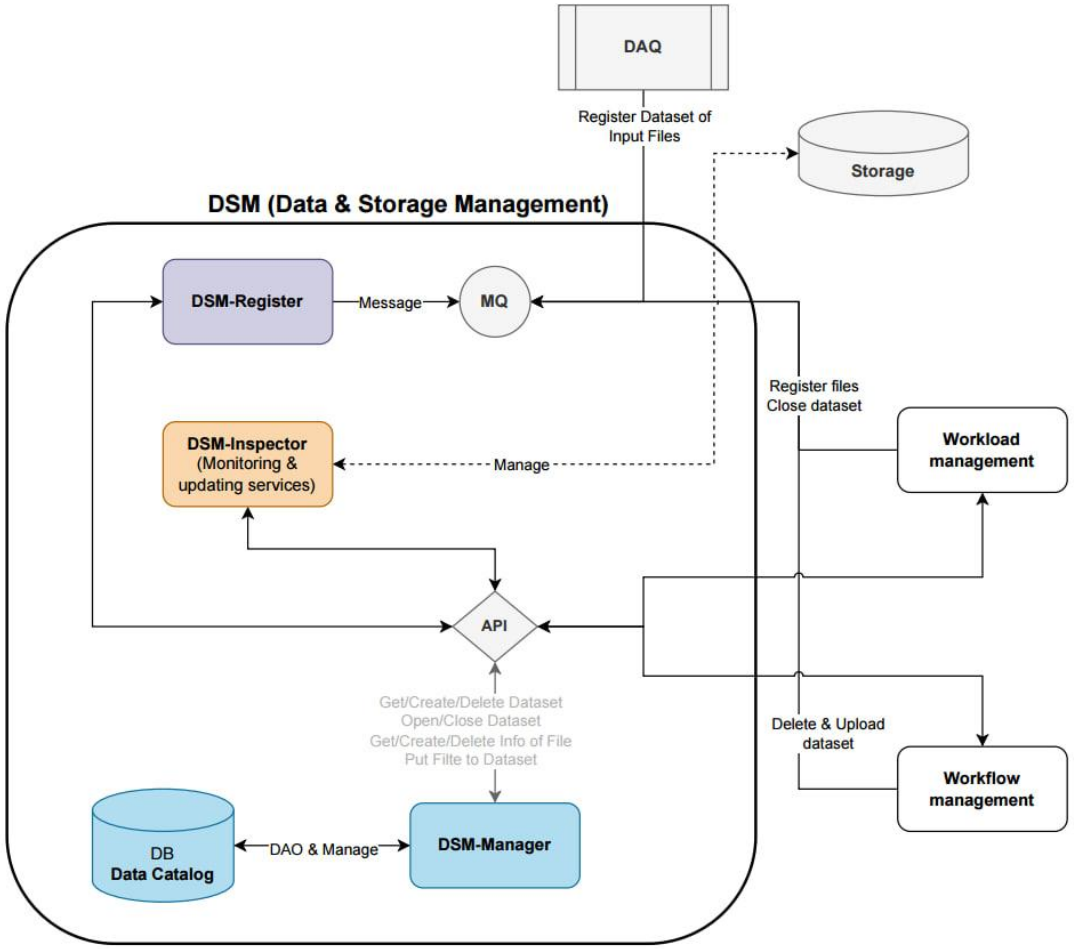
- ✓ Регистрация новых данных
- ✓ Каталогизация
- ✓ Контроль целостности и выгрузки файлов, удаление файлов на хранилищах, мониторинг хранилища

Интерфейс для приема заявок на добавление/удаление данных в системе

Интерфейс к каталогу данных

Набор фоновых задач для обеспечения согласованности хранилища и каталога

Система управления данными



Архитектура DSM

dsm-register

- сервис, принимающий в асинхронном режиме (через очередь сообщений) заявки на добавление/удаление данных в системе

dsm-manager

- сервис, предоставляющий REST API к каталогу данных (размещение данных в каталоге, обращение к каталогу, изменение данных в каталоге)

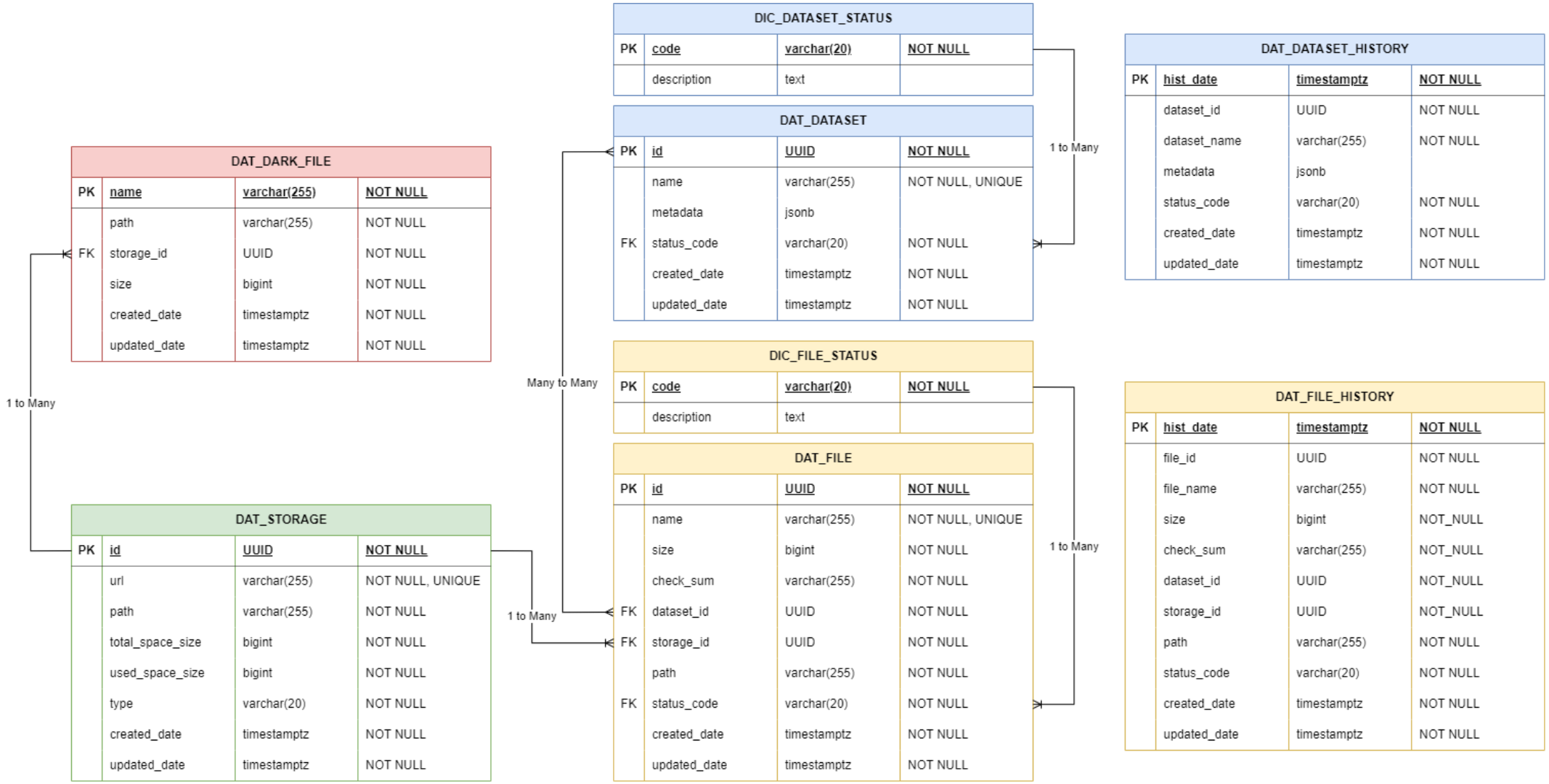
dsm-inspector

- набор фоновых сервисов для мониторинга и контроля состояния данных в хранилище

Задачи

- ✓ Реализация сервиса *dsm-register* для взаимодействия с системой управления процессами обработки и с системой управления нагрузкой посредством очередей сообщений
- ✓ Реализация сервиса *dsm-inspector* для контроля состояния данных на хранилищах
- ✓ Проведение этапа интеграционного тестирования между системами (выявление неточностей в реализованном функционале и его доработка)

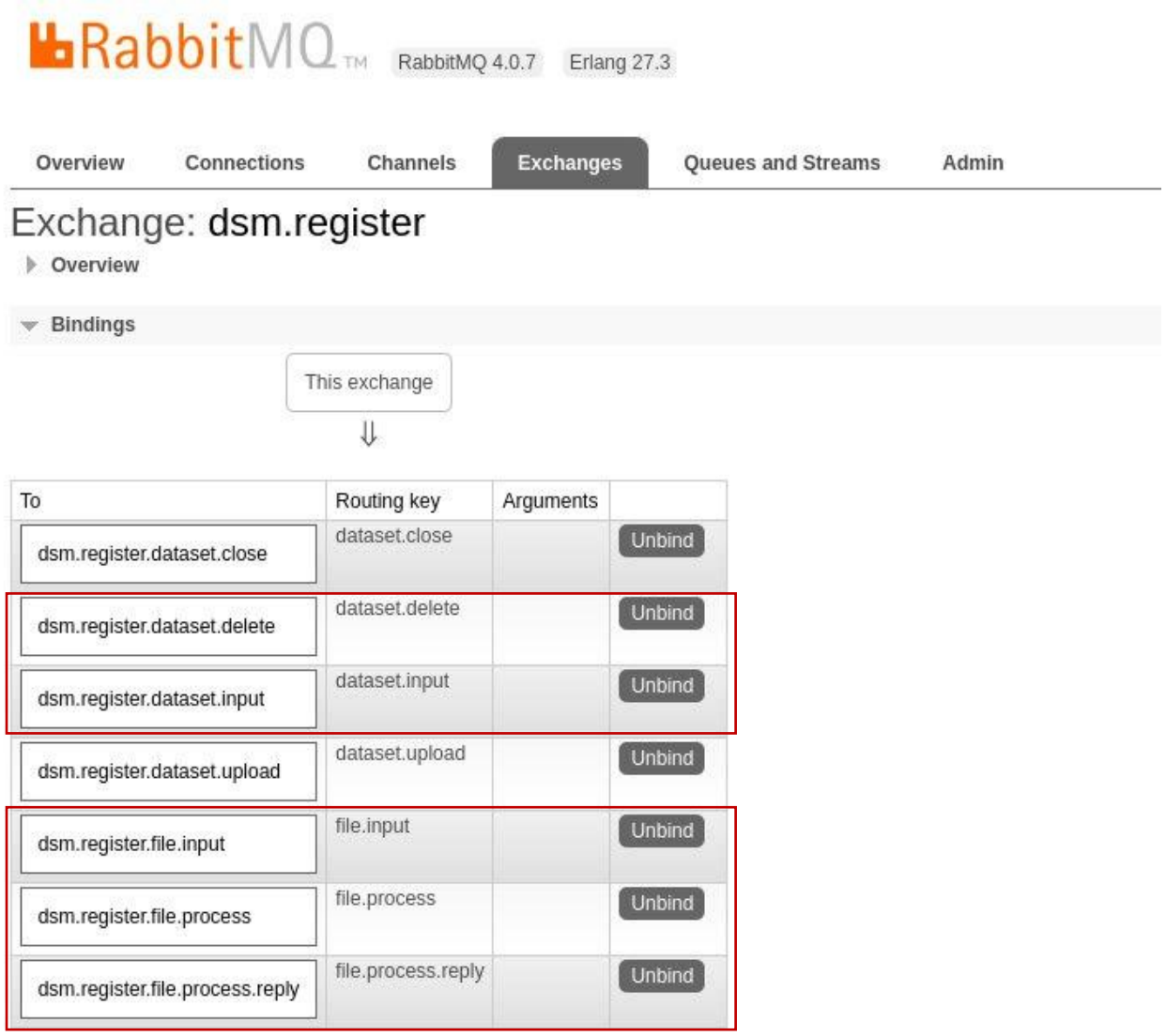
Структура БД



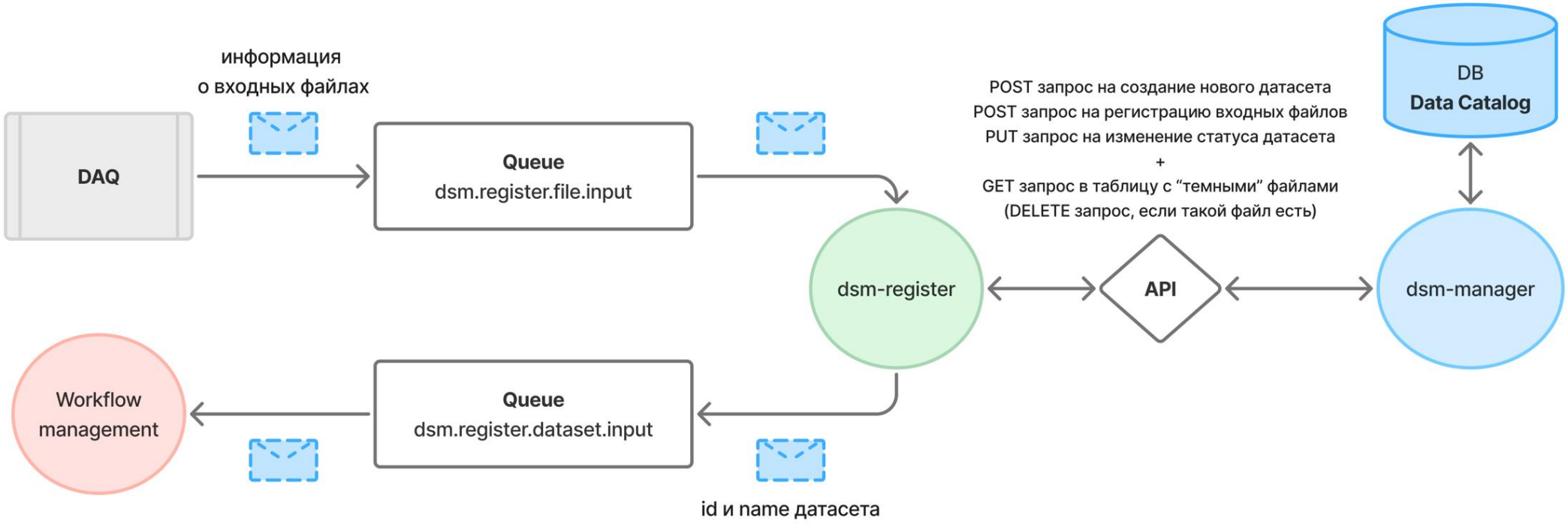
dsm-register: сконфигурированные очереди

Сервис должен слушать очередь сообщений и обрабатывать заявки на добавление/удаление данных в системе.

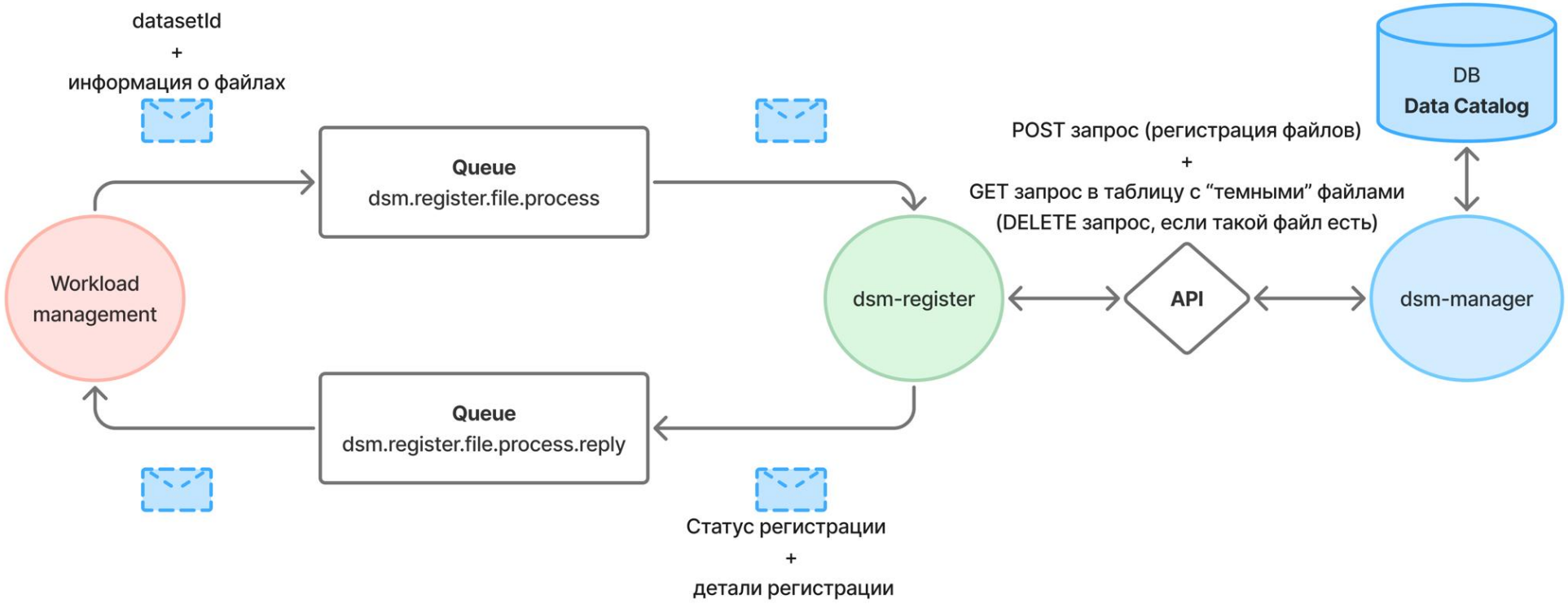
В качестве AMQP-брокера, который осуществляет маршрутизацию и подписку на нужные очереди, используется RabbitMQ.



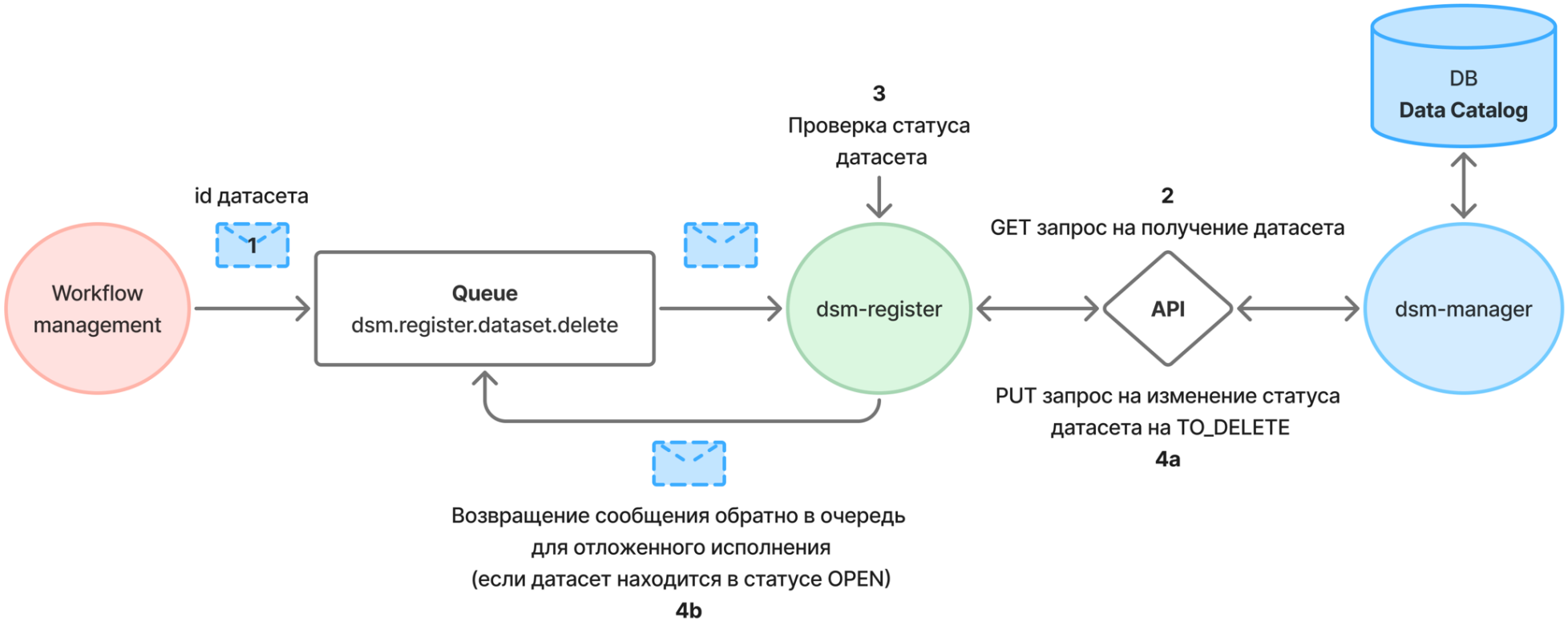
dsm.register.file.input и dsm.register.dataset.input



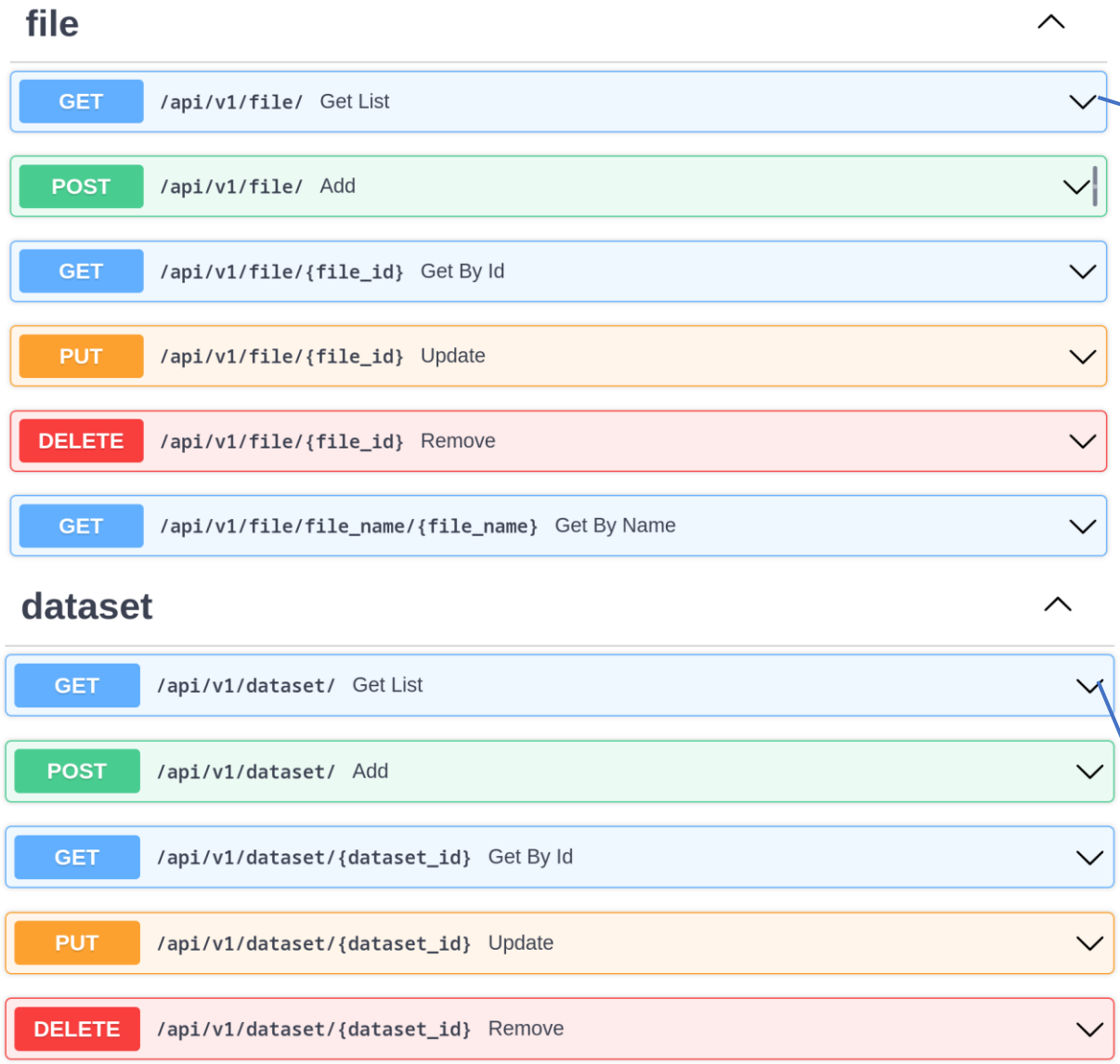
dsm.register.file.process и dsm.register.file.process.reply



dsm.register.dataset.delete



dsm-manager: API к БД



- ✓ Получение содержимого набора
- ✓ Получение списка файлов в определенном статусе

FastAPI

Сервис должен предоставлять REST API к каталогу данных.

В качестве веб-фреймфорка используется асинхронный фреймворк **FastAPI**

- ✓ Получение списка датасетов, в которых есть определенный файл
- ✓ Получение списка датасетов в определенном статусе

dsm-manager: API к БД

storage

- GET `/api/v1/storage/` Get List
- POST `/api/v1/storage/` Add
- GET `/api/v1/storage/{storage_id}` Get By Id
- PUT `/api/v1/storage/{storage_id}` Update
- DELETE `/api/v1/storage/{storage_id}` Remove
- GET `/api/v1/storage/type/{storage_type}` Get By Type

dark_file

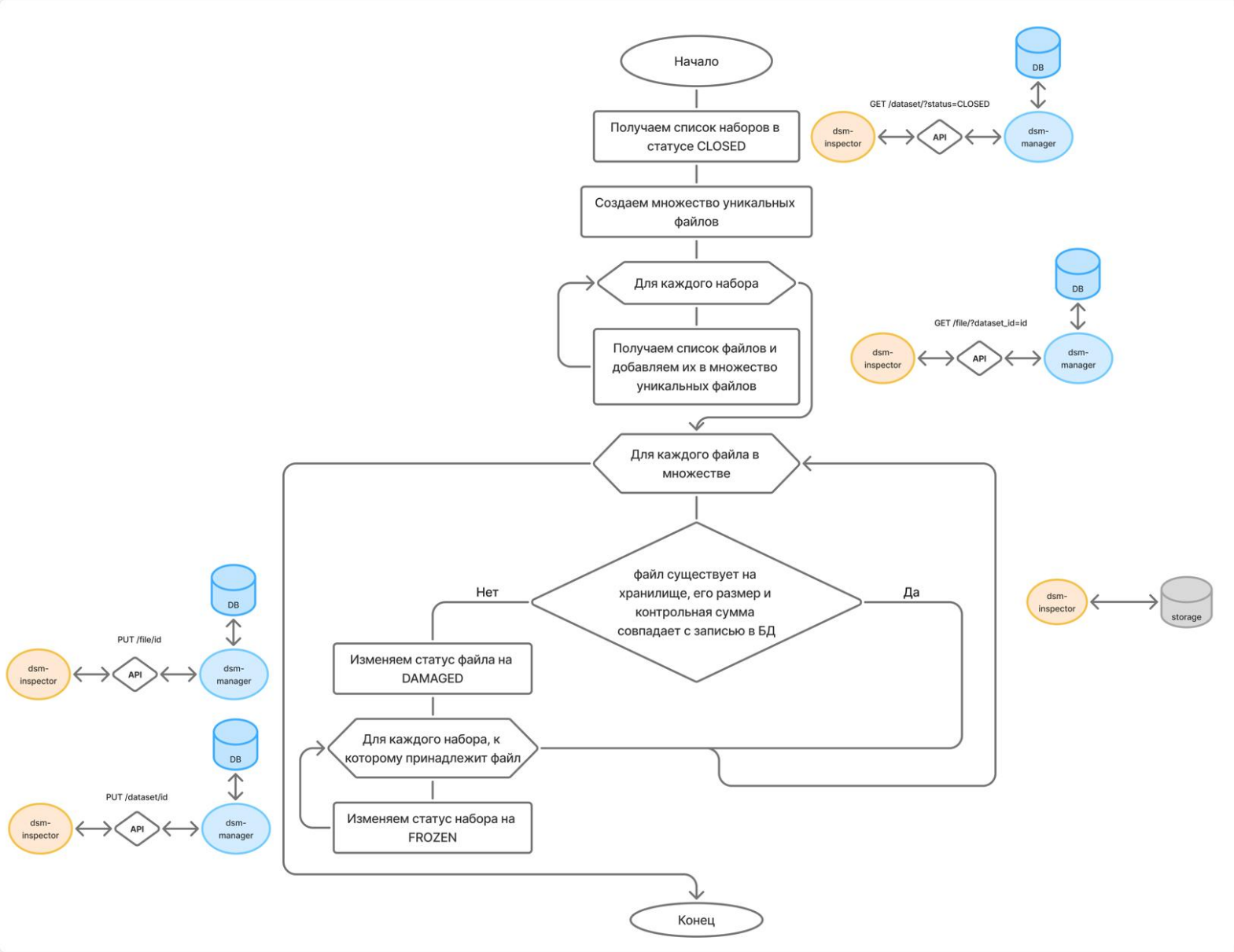
- GET `/api/v1/dark_file/` Get List
- POST `/api/v1/dark_file/` Add
- GET `/api/v1/dark_file/{file_name}` Get By Name
- DELETE `/api/v1/dark_file/{file_name}` Remove

dsm-inspector

Сервис состоит из набора **фоновых** задач:

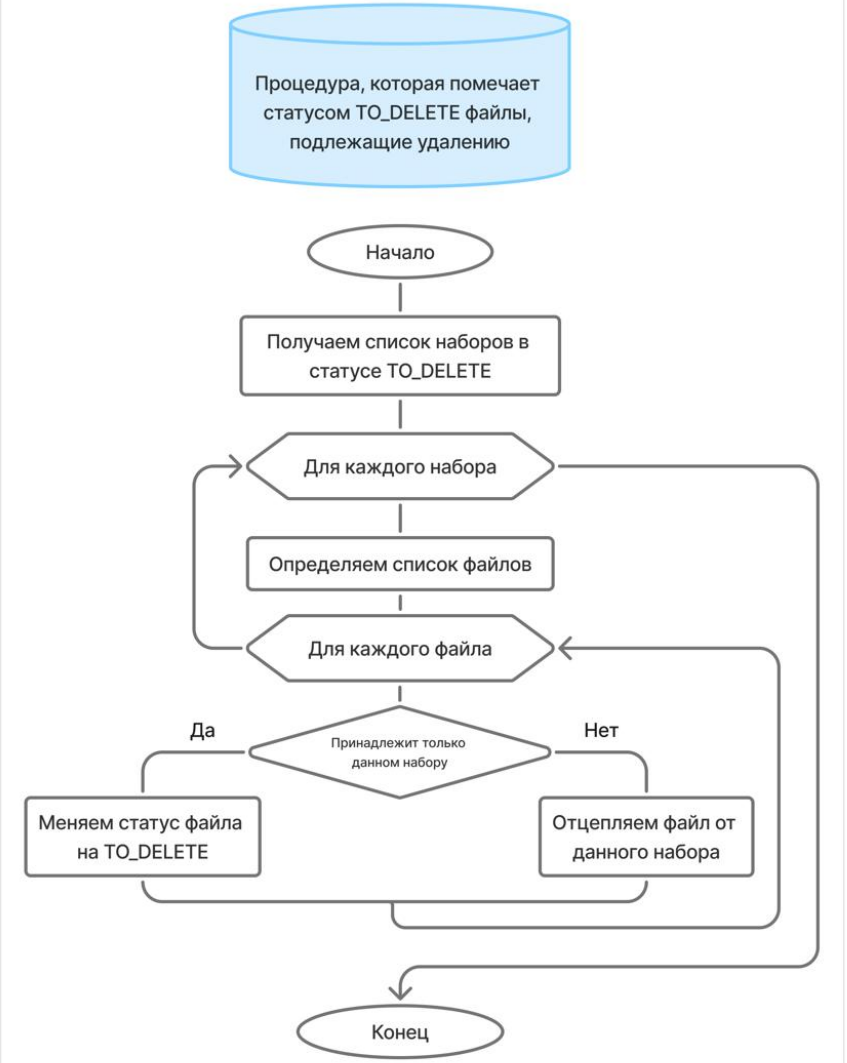
- ✓ Удаление файлов на хранилищах
- ✓ Контроль выгрузки файлов
- ✓ Проверка целостности файлов
- ✓ Контроль использования хранилища

Проверка целостности файлов

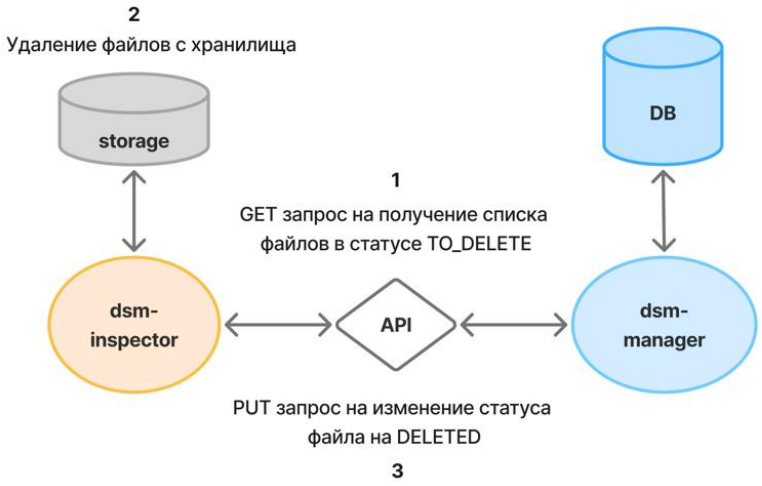


Удаление датасетов и файлов

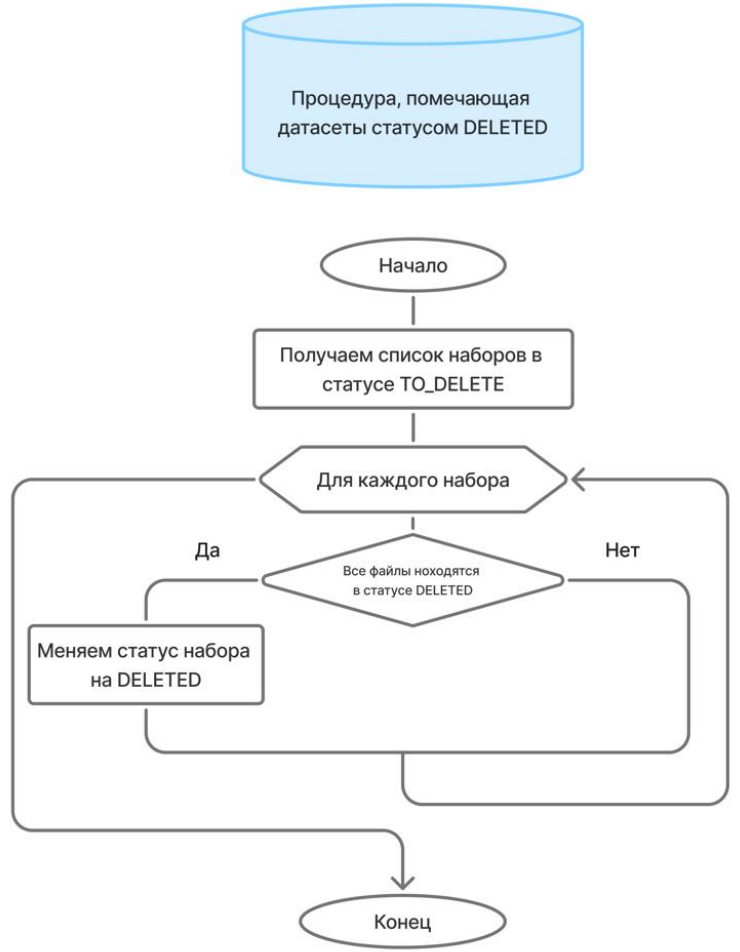
1. Определение списка фалов, подлежащих удалению



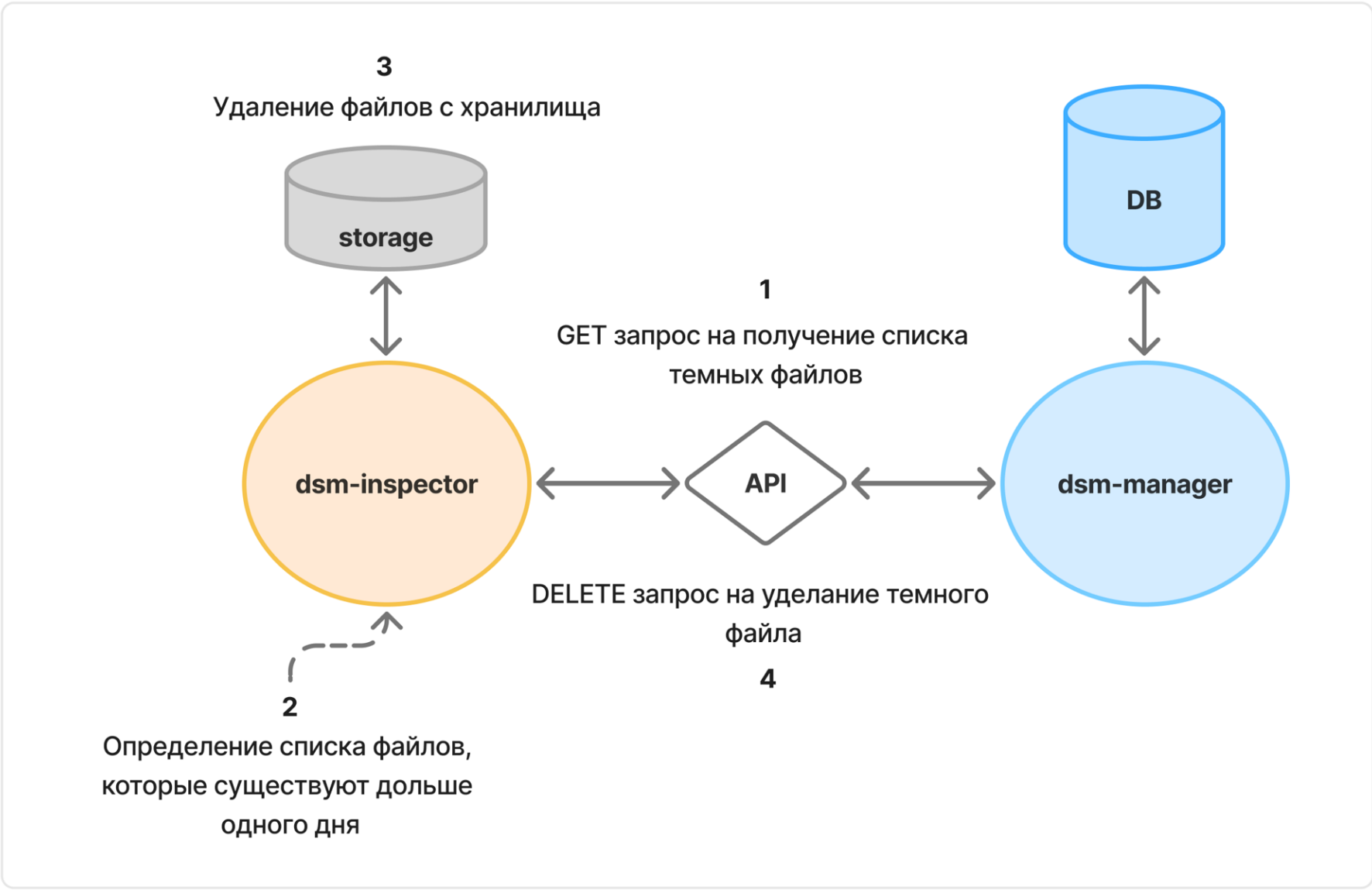
2. Удаление файлов в датасетах



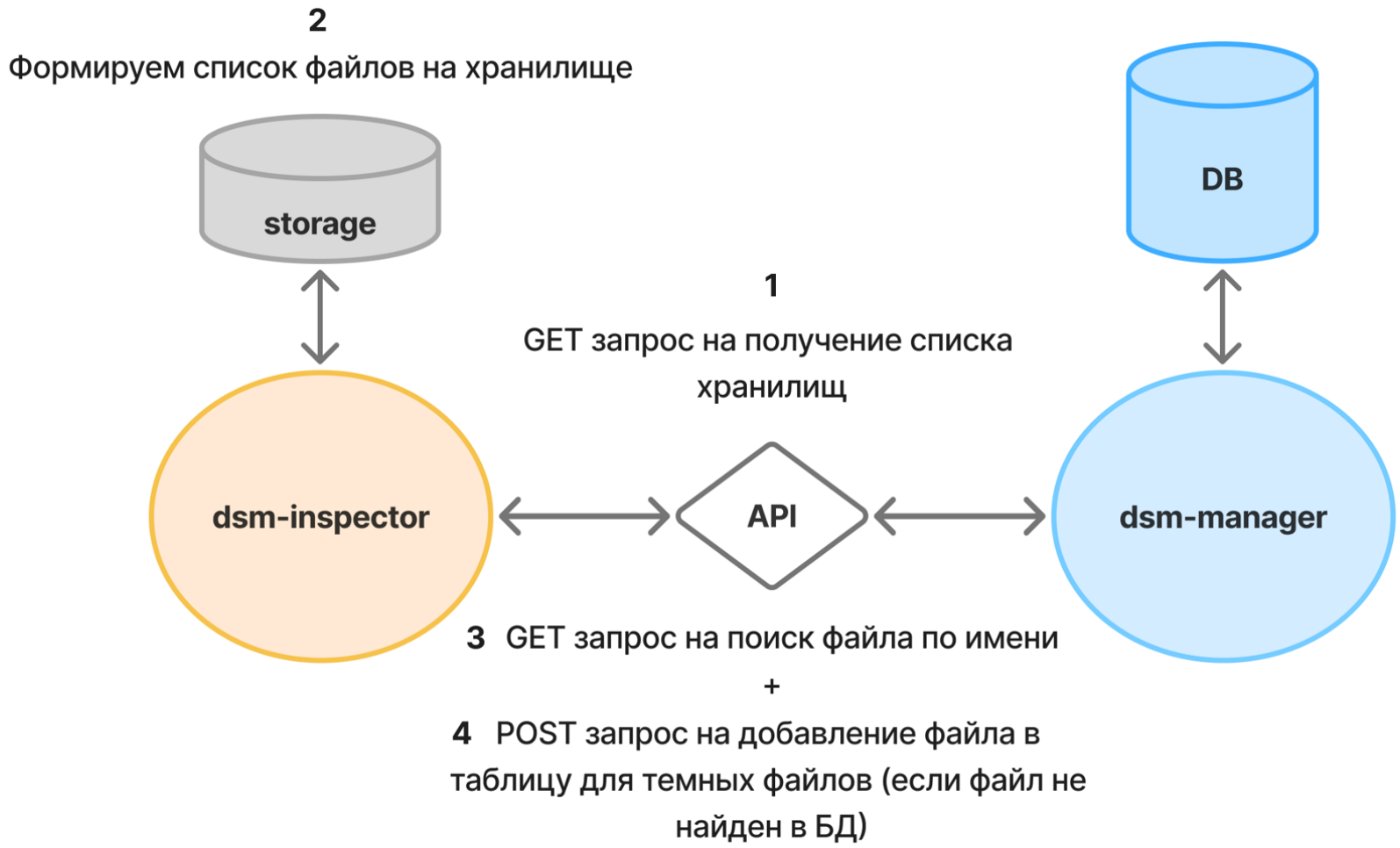
3. Удаление датасетов



Удаление темных файлов



Контроль использования хранилища: мониторинг тёмных файлов



Заключение

Текущие результаты:

- ✓ **dsm-manager** полностью функционален для данного этапа реализации
- ✓ **dsm-register** реализован для данного этапа
- ✓ **dsm-inspector** реализован примерно на 50%

Дальнейшие планы:

dsm-inspector:

Реализовать фоновые сервисы для:

- ➢ Контроля выгрузки файлов
- ➢ Мониторинга заполненности хранилища

dsm-register:

Реализовать обработку сообщений из очередей:

- ➢ dsm.register.dataset.close
- ➢ dsm.register.dataset.upload