Data Management System for SPD Online Filter

Korshunova Polina
JINR MLIT

IX SPD collaboration meeting
AANL Yerevan
2025

Middleware software

Data management system

- data lifecycle support (data catalog, consistency check, cleanup, storage)

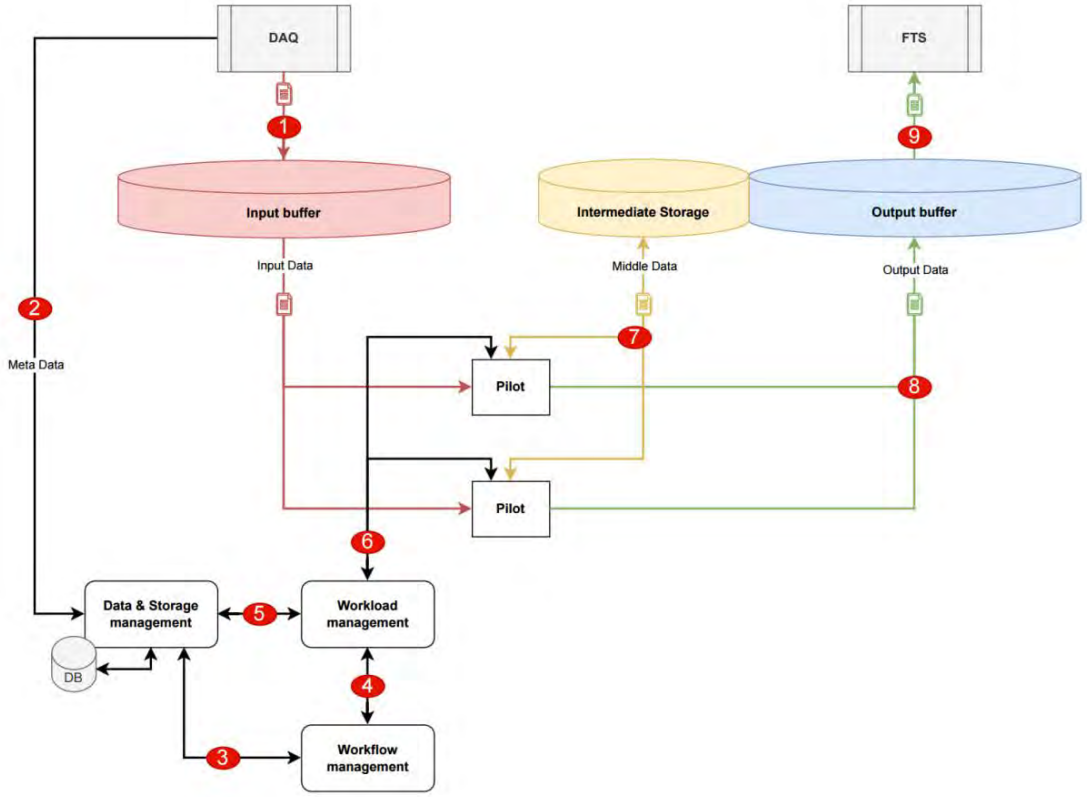
Workflow Management System

- define and execute processing chains by generating the required number of computational tasks

Workload management system

- implementation of processing stages (task generation, sending tasks to pilots)

Pilot – an application running on a computing node and performing tasks

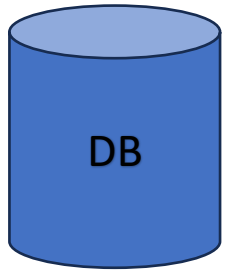


Data management system: tasks

- ✓ Registration of new data
- ✓ Cataloging
- ✓ File integrity and upload control, file deletion on storages, storage monitoring

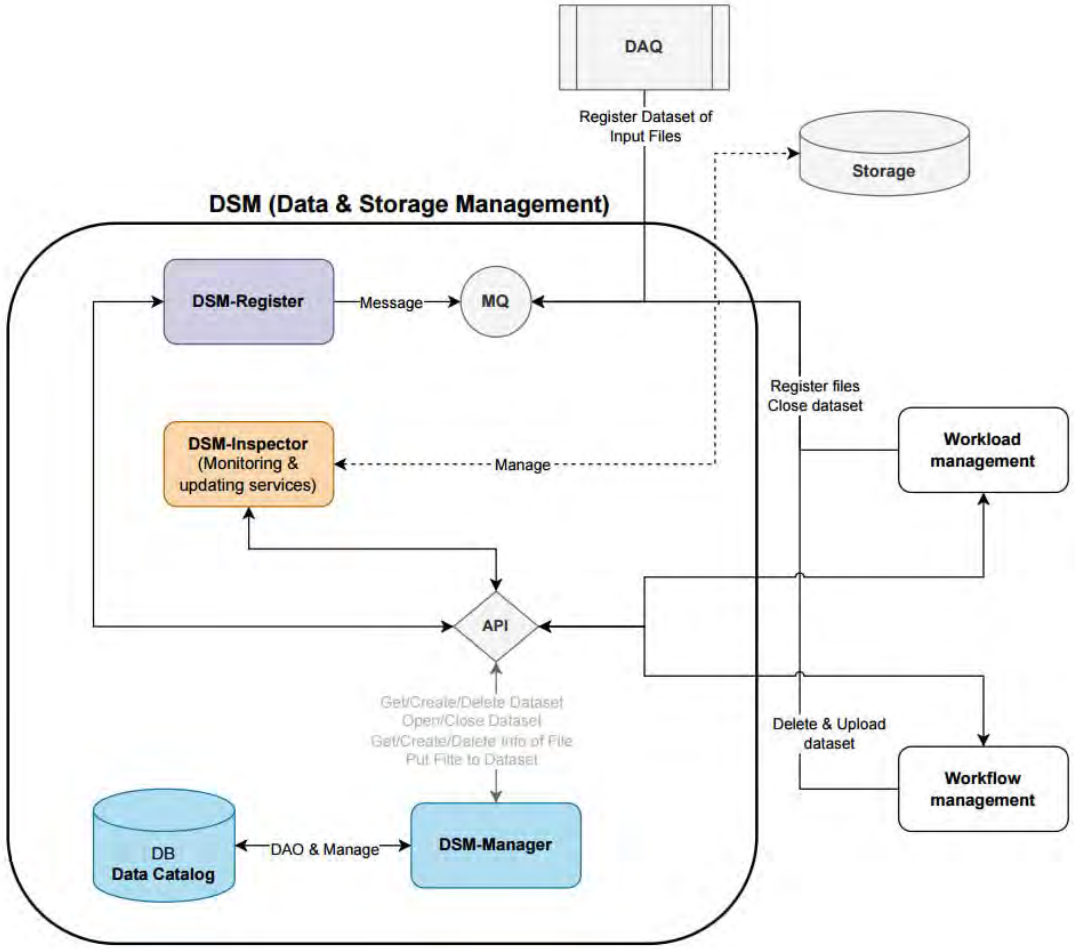
Interface for accepting requests for adding/deleting data in the system

Interface to the data catalog



A set of background tasks to ensure consistency between storage and database

Data management system



Architecture of Data Management System

dsm-register (data registration)

- a service that receive requests for adding/deleting data in the system asynchronously (via MQ). Then the service makes changes to the data catalog via the API of the dsm-manager

dsm-manager (REST API of data catalog)

- file and dataset management (adding data to a database, changing data, deleting data)

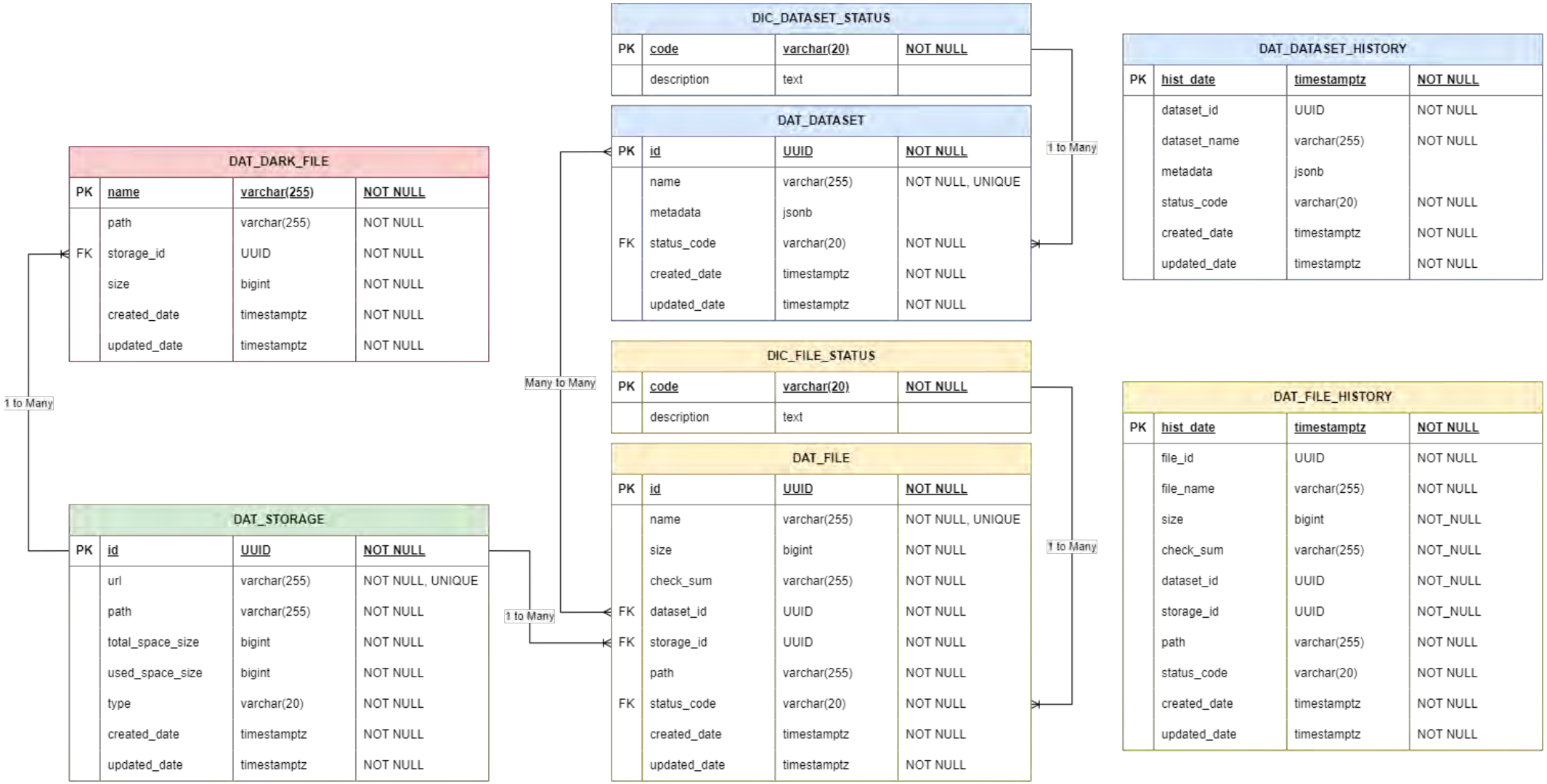
dsm-inspector (daemon tasks)

- delete files on storage, check consistency of files, monitoring the use of storage (for example, "dark" data)

Tasks

- ✓ Implementation of the dsm-register service for interaction with the workflow management system and with the workload management system via message queues
- ✓ Implementation of the dsm-inspector service for monitoring the state of data on storages
- ✓ Implementation of the integration testing stage between the systems (identification of inaccuracies in the implemented functionality and its refinement)

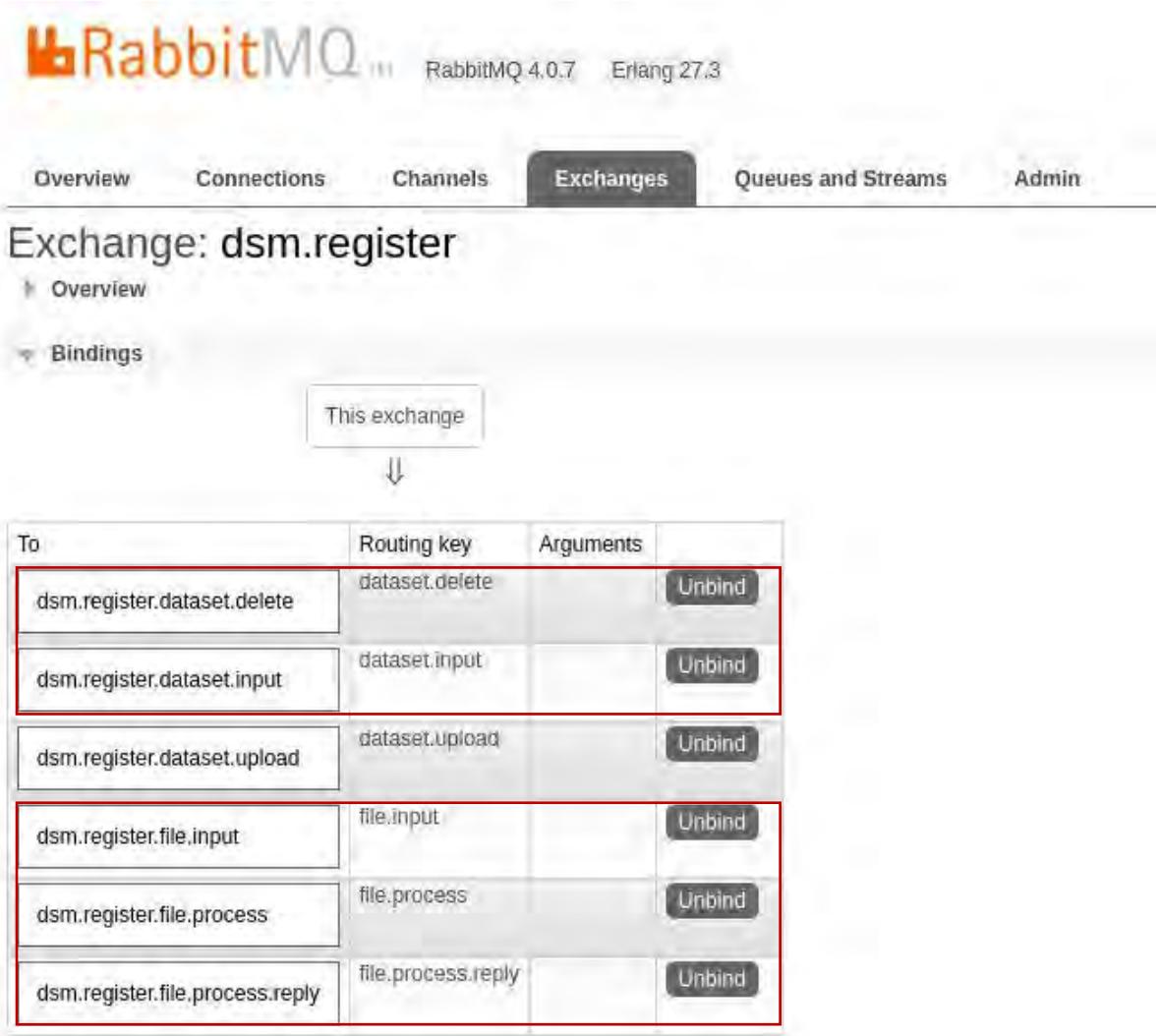
Database structure



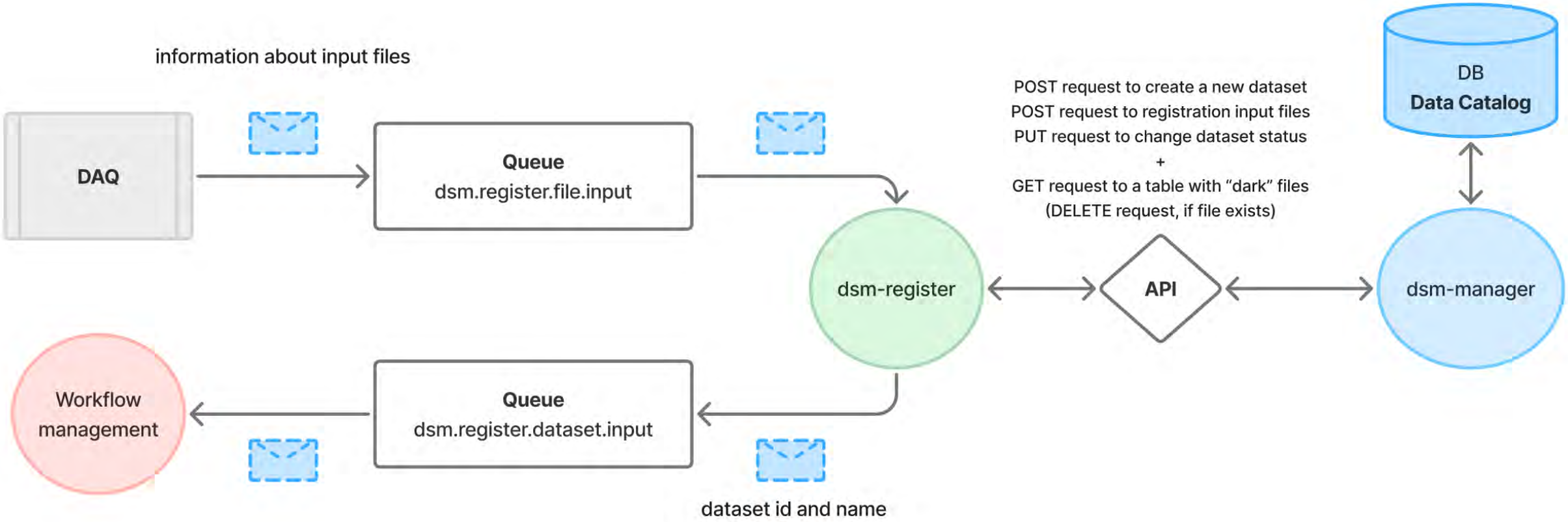
dsm-register: configured queues

The service should listen to the message queue and process requests for adding/deleting data in the system.

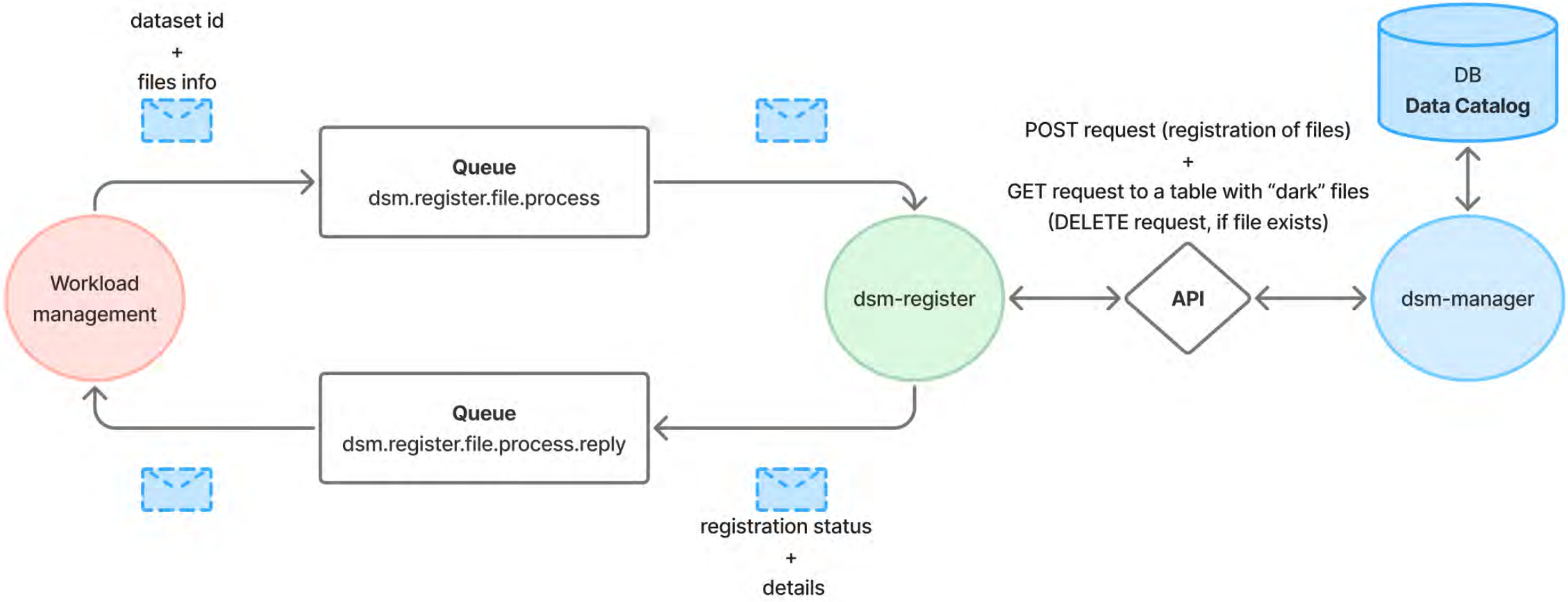
RabbitMQ is used as an AMQP broker that performs routing and subscribing to the necessary queues.



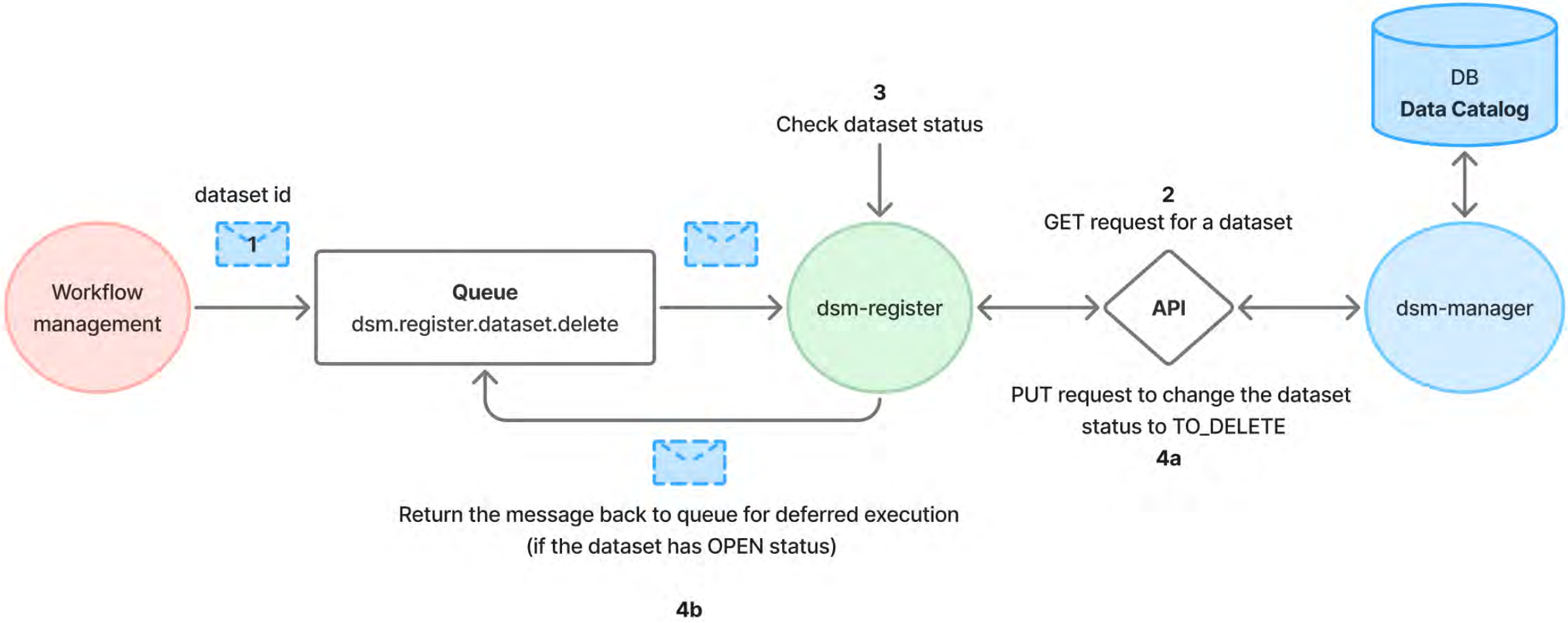
dsm.register.file.input и dsm.register.dataset.input



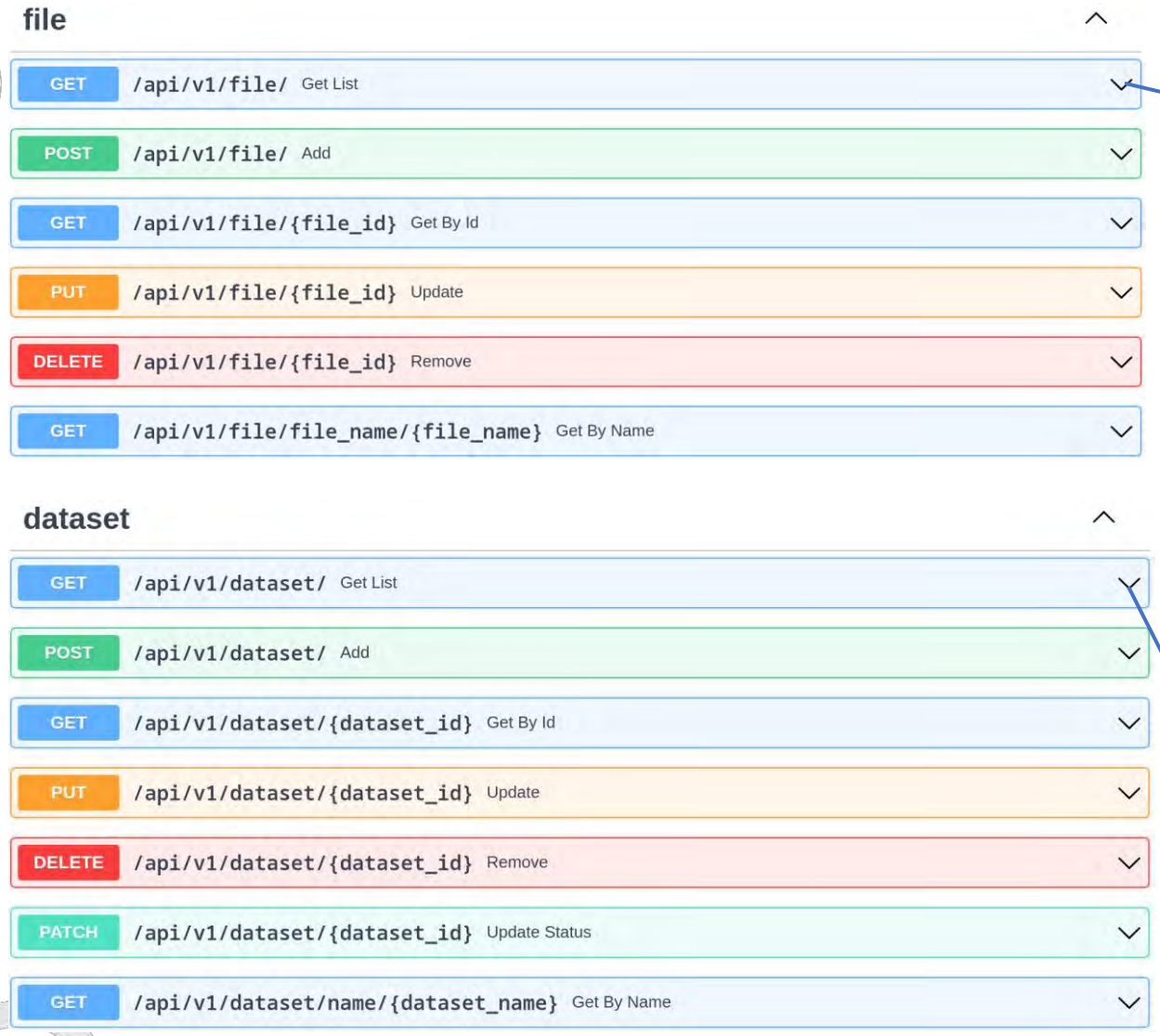
dsm.register.file.process and dsm.register.file.process.reply



dsm.register.dataset.delete



dsm-manager: API to the DB



- ✓ Getting a list of files in a dataset
- ✓ Getting a list of files with a specific status

FastAPI

The service must provide a REST API to the data catalog.

The asynchronous **FastAPI** framework is used as a web framework.

- ✓ Getting a list of datasets that contain a specific file
- ✓ Getting a list of datasets with a specific status

dsm-manager: API to the DB

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

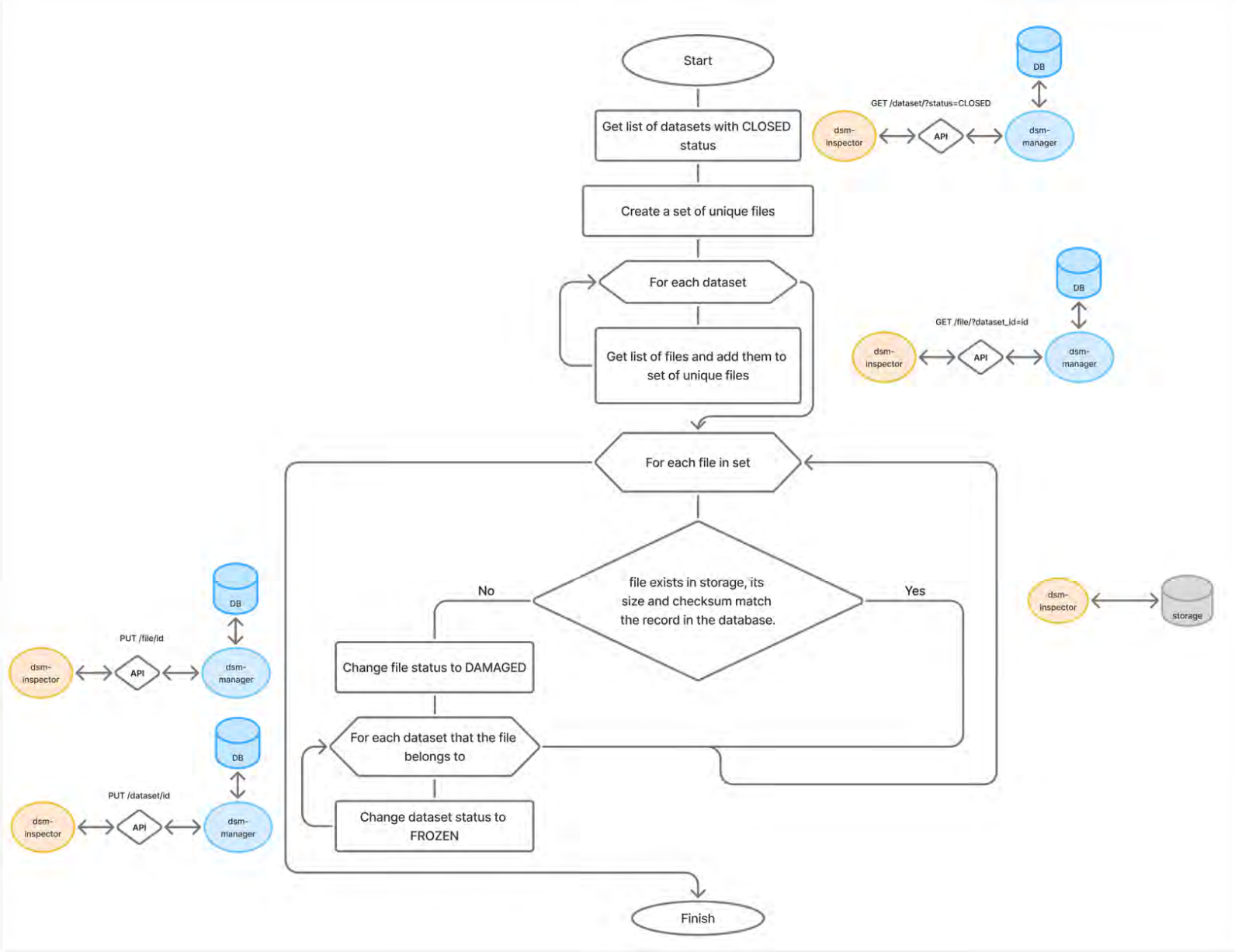
- GET `/api/v1/dark_file/` Get List
- POST `/api/v1/dark_file/` Add
- GET `/api/v1/dark_file/{file_name}` Get By Name
- DELETE `/api/v1/dark_file/{file_name}` Remove

dsm-inspector

The service consists of a set of **background** tasks:

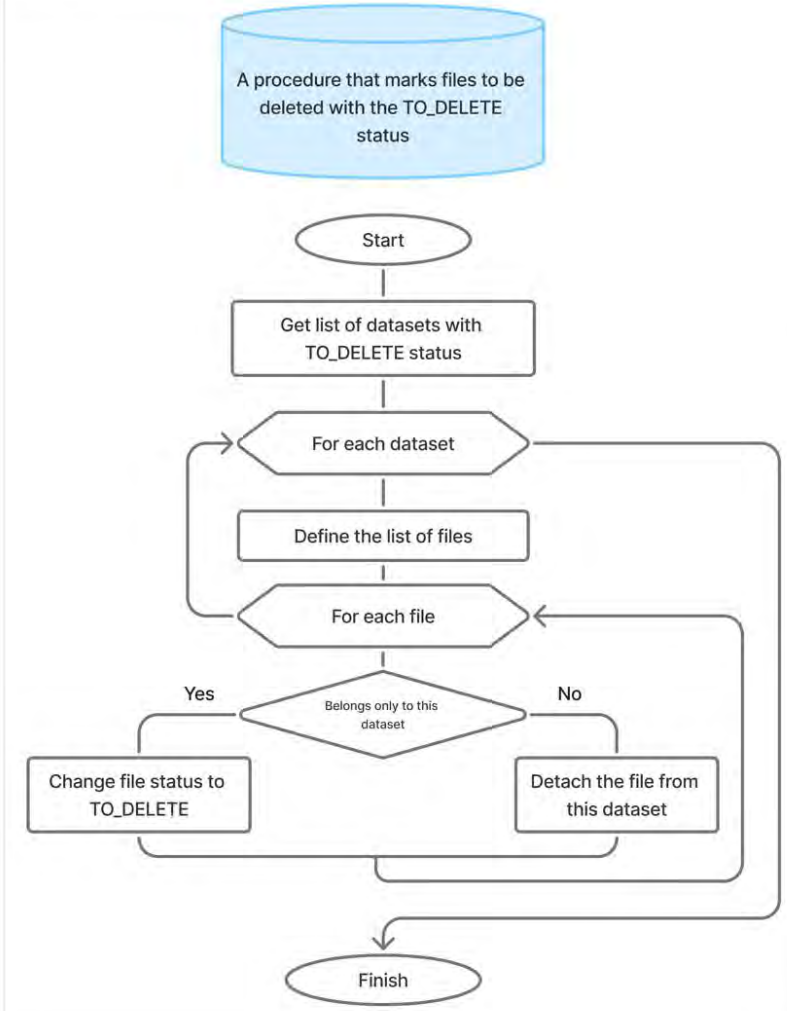
- Deleting files on storages
- File upload control
- File integrity check
- Monitoring storage usage

File Integrity Check

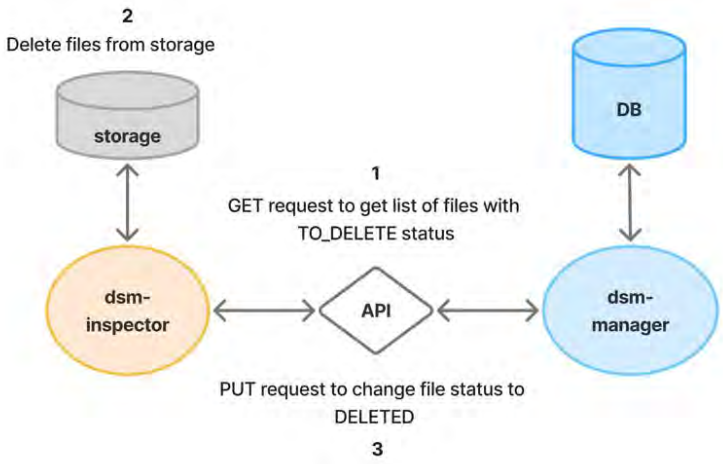


Deleting datasets and files

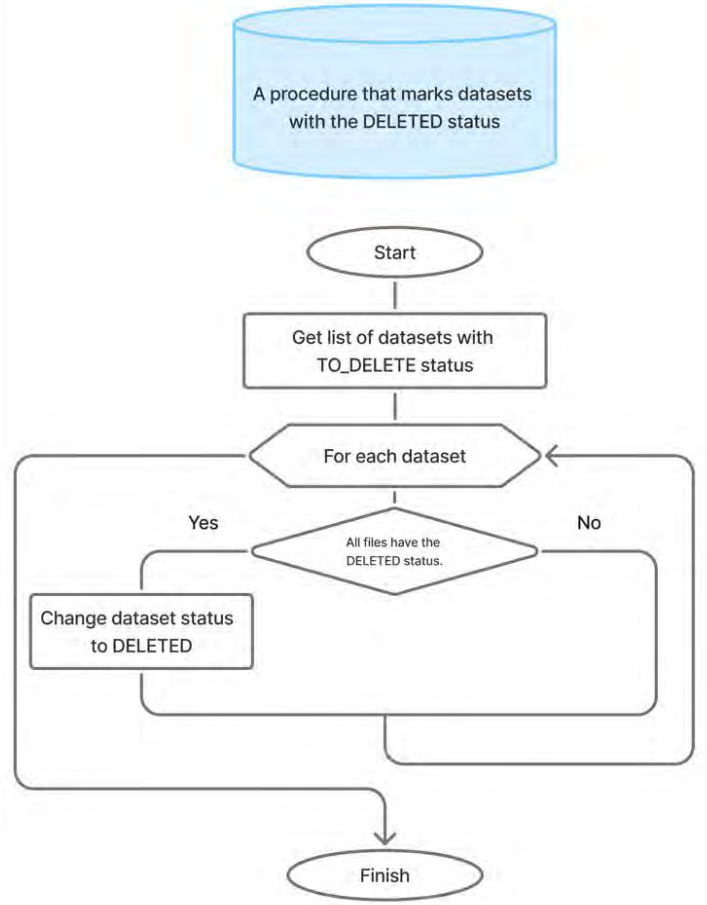
1. Determining the list of files to be deleted



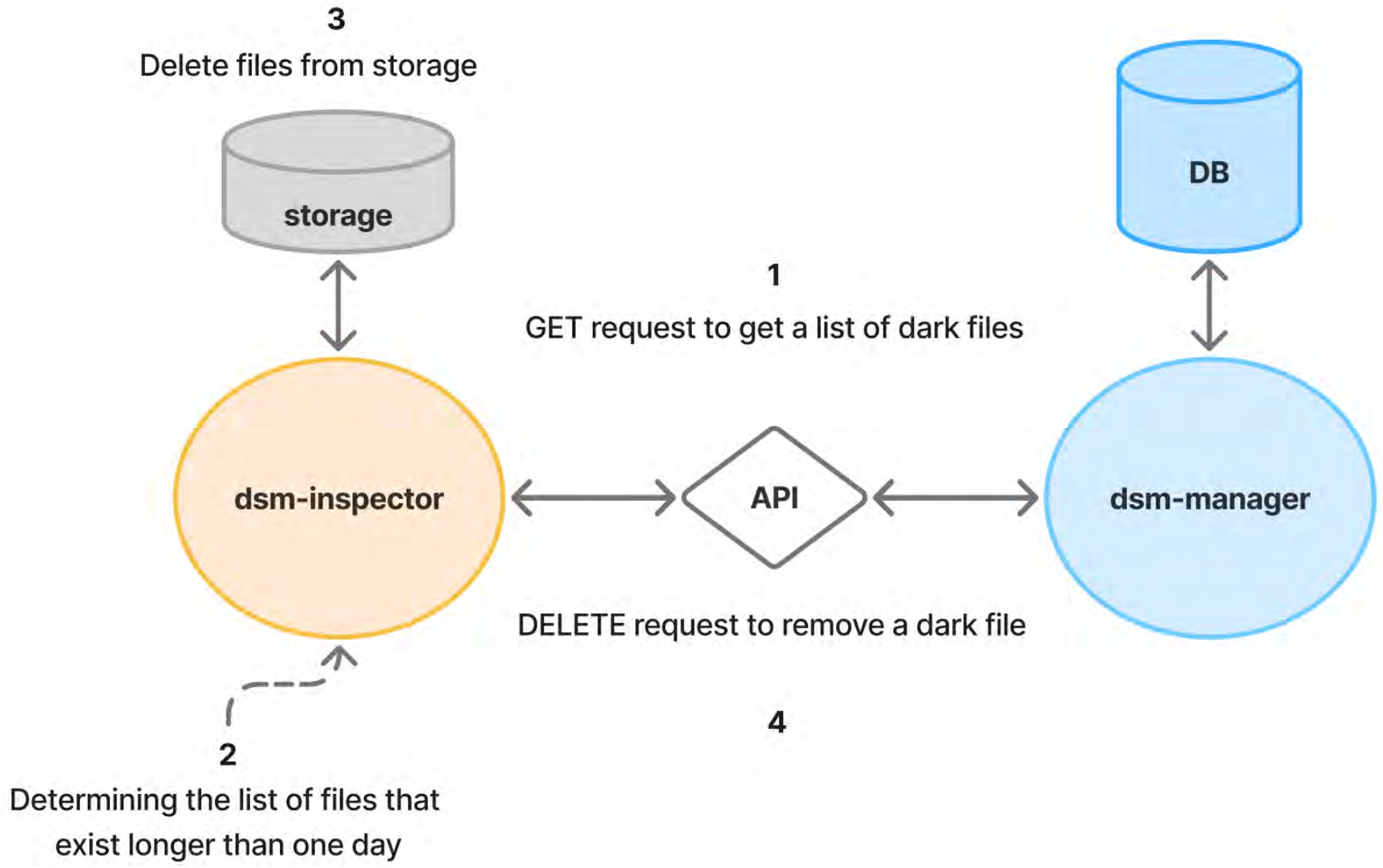
2. Deleting files in datasets



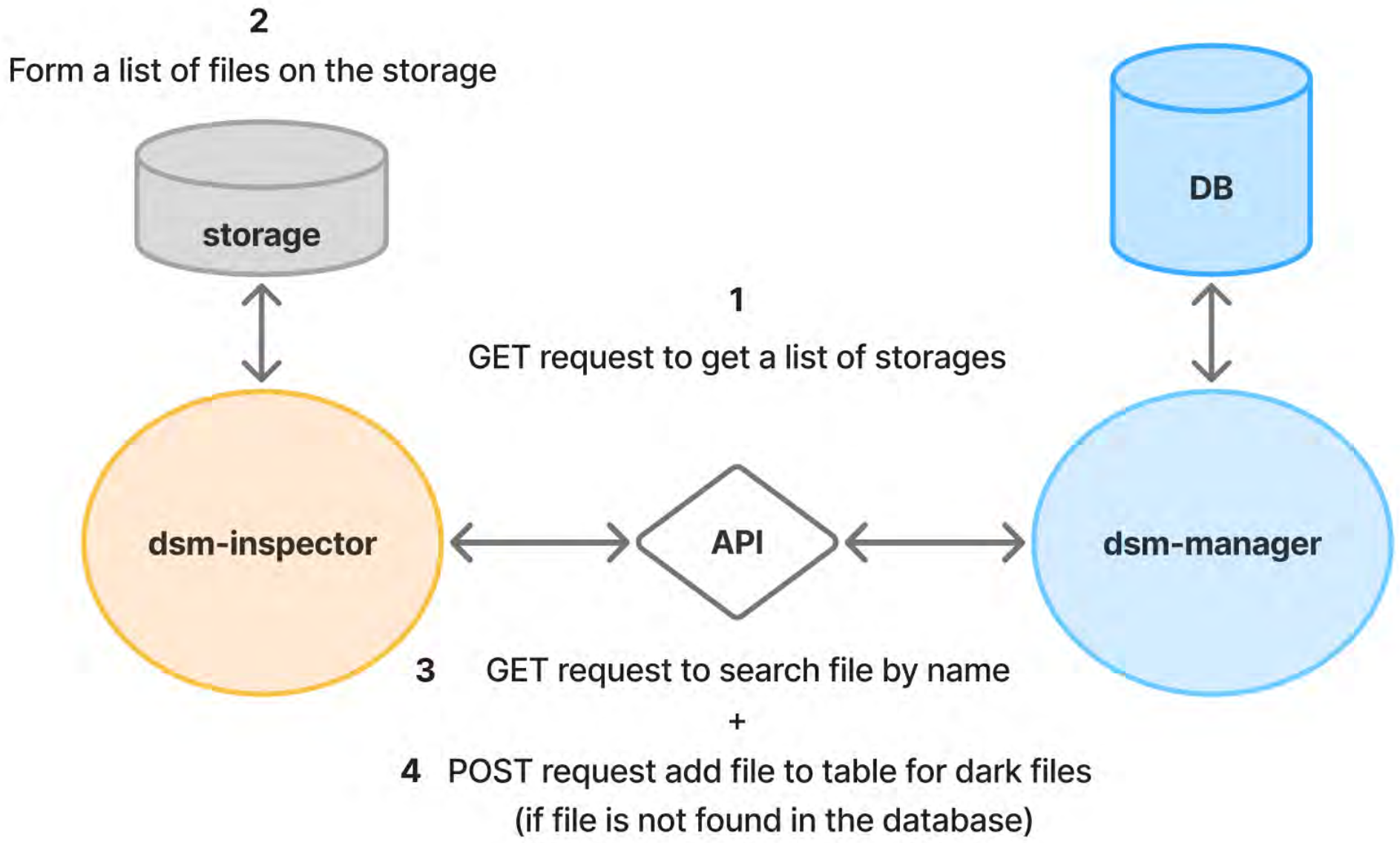
3. Deleting datasets



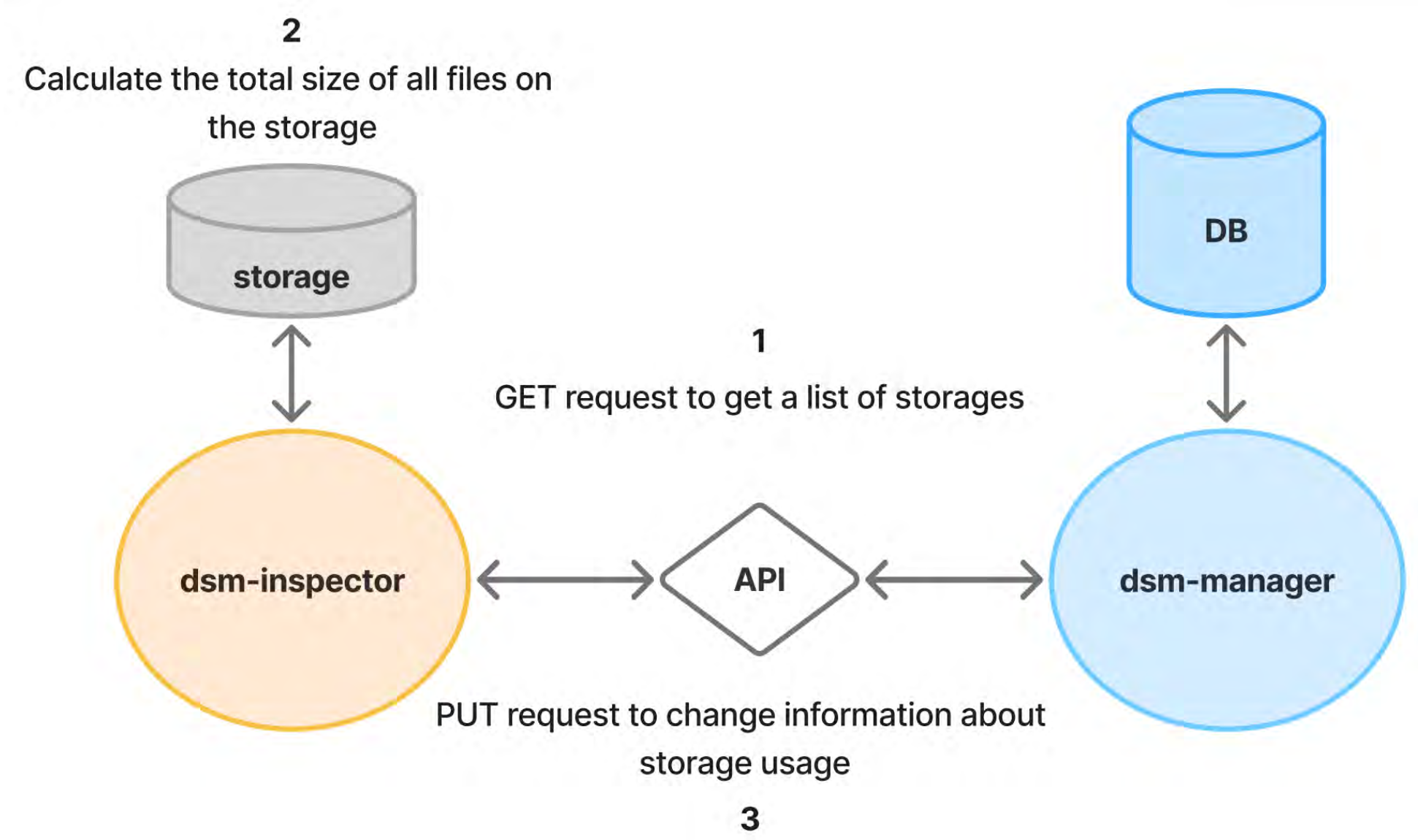
Deleting dark files



Storage usage monitoring: monitoring of dark files



Storage usage monitoring



Conclusion

Current results:

- ✓ **dsm-manager** fully functional for this stage of implementation
- ✓ **dsm-register** implemented for this stage
- ✓ **dsm-inspector** is implemented by 75%

Further plans:

dsm-inspector:

Implement background services for

- ➢ File upload control

dsm-register:

Realize processing of messages from queues:

- ➢ dsm.register.dataset.upload