Data Management for SPD Online Filter: Status Update

Korshunova Polina
JINR MLIT

XI SPD collaboration meeting
Tomsk
2026

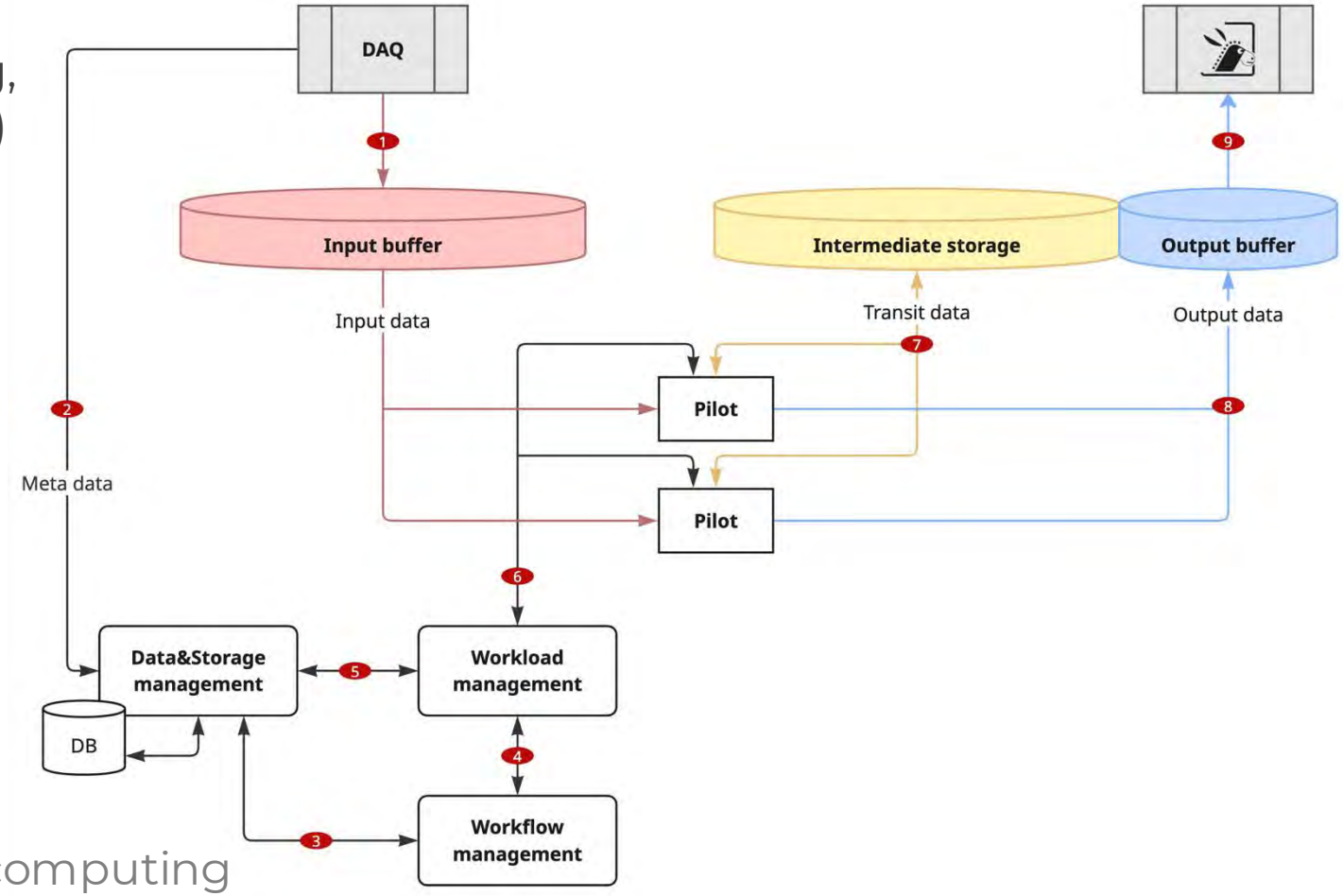
Reminder: Middleware Software

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



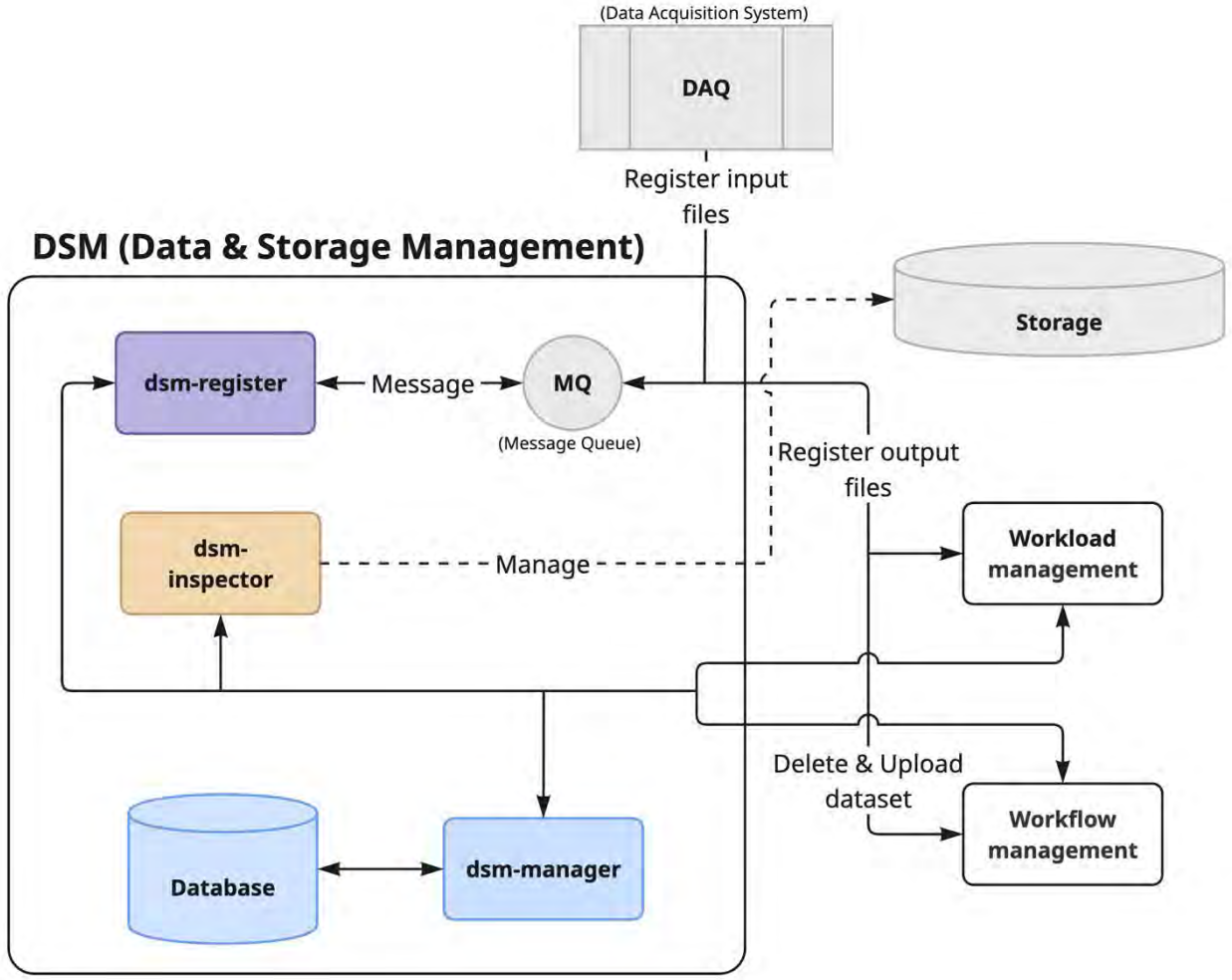
Reminder: Key Features of DMS

Purpose: Automation of lifecycle management of experimental data at the stage of primary processing

Key Features:

- Separation of metadata and physical file storage (cataloging using a database)
- Data lifecycle tracking (maintaining a status model of file and dataset)
- Registration of files and datasets via API
- Ensuring consistency between the physical storage and the database using background services

Reminder: Architecture of DMS

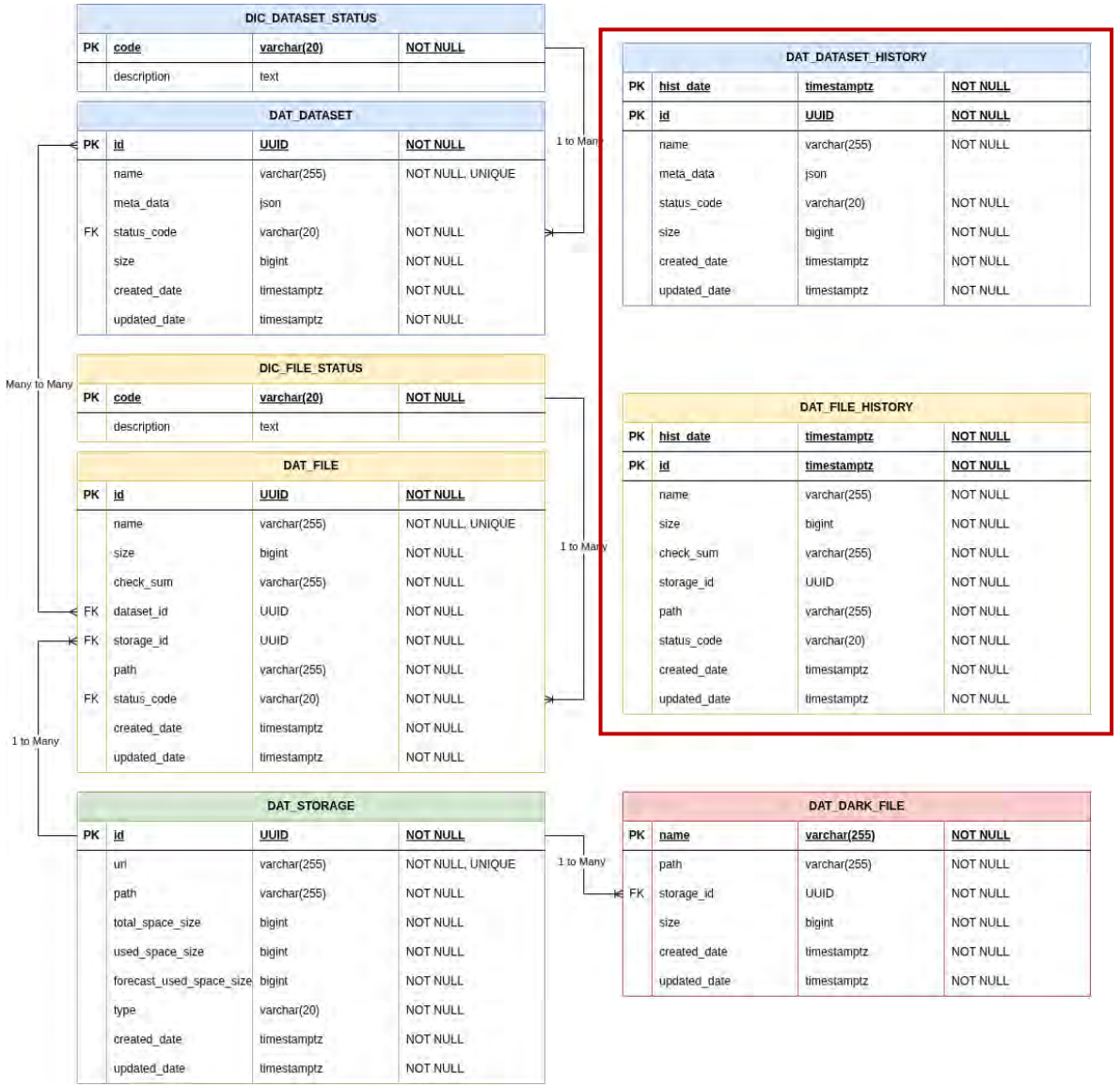


dsm-register (data registration)
- a service that receive requests for adding/deleting/uploading data in the system asynchronously (via MQ)

dsm-manager (REST API of data catalog)
- file and dataset management (adding data to a database, changing data, deleting data)

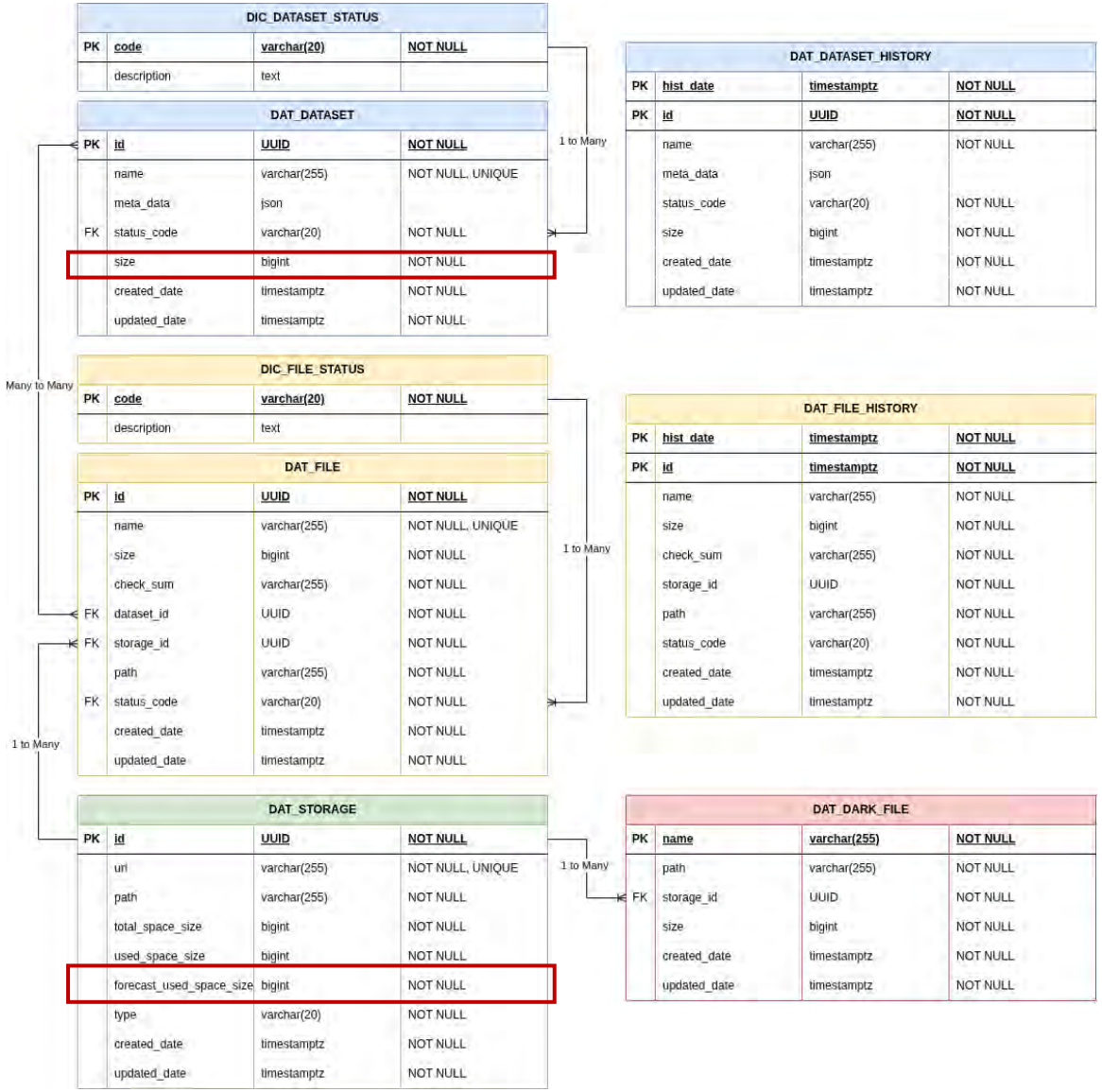
dsm-inspector (daemon tasks)
- a set of background tasks for monitoring data stage: checking consistency, monitoring storage, deleting files, and uploading to external storage

Database Structure: Update



- Need history tables to keep track of all file and dataset states
  - ✓ Add history tables for files and datasets
  - ✓ Automatic filling of table using insert/update triggers
  - ✓ Set up automatic partitioning using pg_partman with a partition size equal to a month

Database Structure: Update

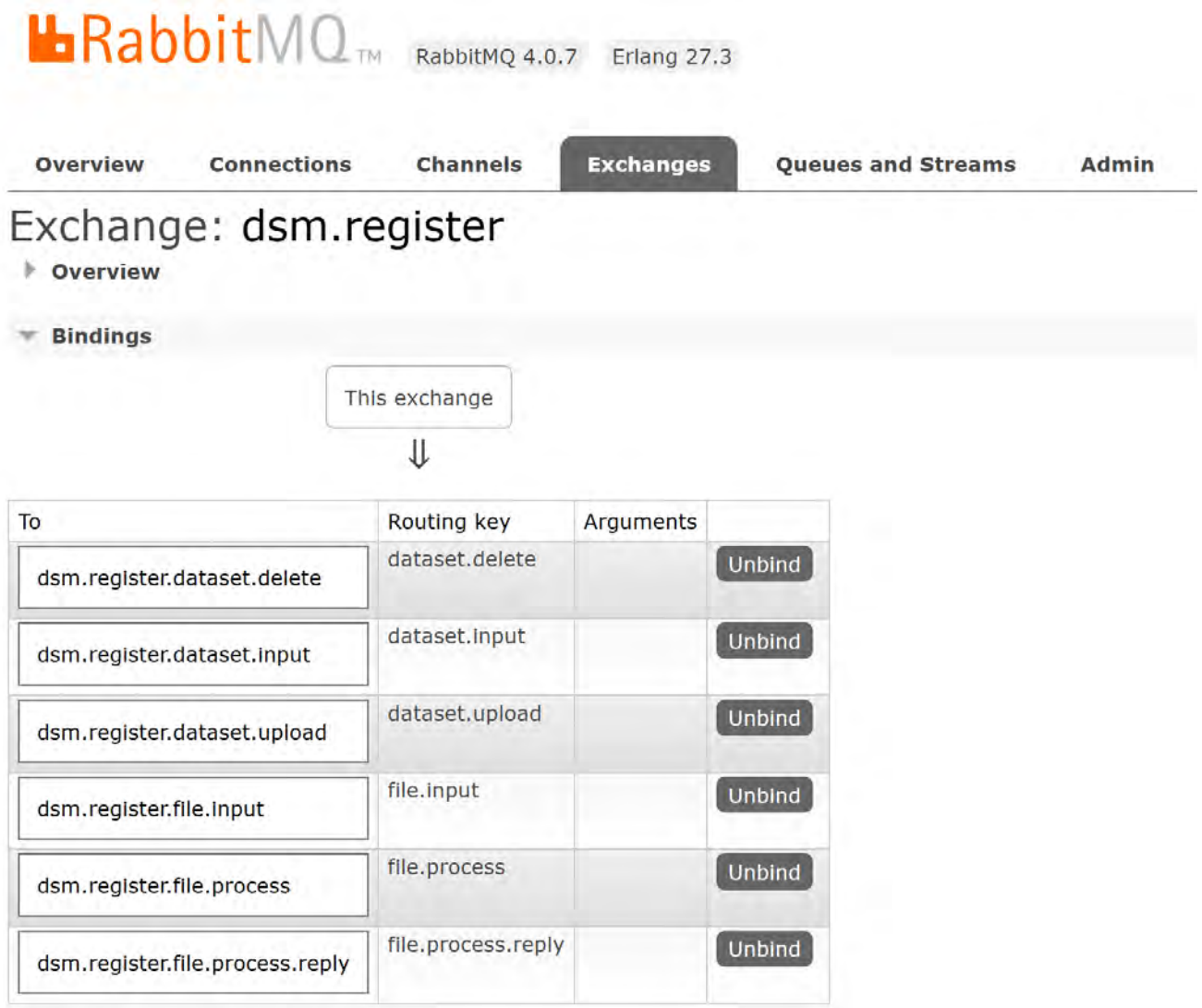


- WFMS needs to know about the size of the dataset in order to make a decision about deletion
  - ✓ Add dataset size
  - ✓ Increasing the size of the dataset when adding files to it, reducing the size of the dataset when deleting files from storage
- WFMS needs to know the current size of the occupied storage space
  - ✓ Add forecast used space size for storages
  - ✓ The field is changed by a trigger
    - when adding a new file, the size increases
    - when the file status changes to DELETED, the size decreases
    - takes into account only registered files

Dsm-register: Update

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

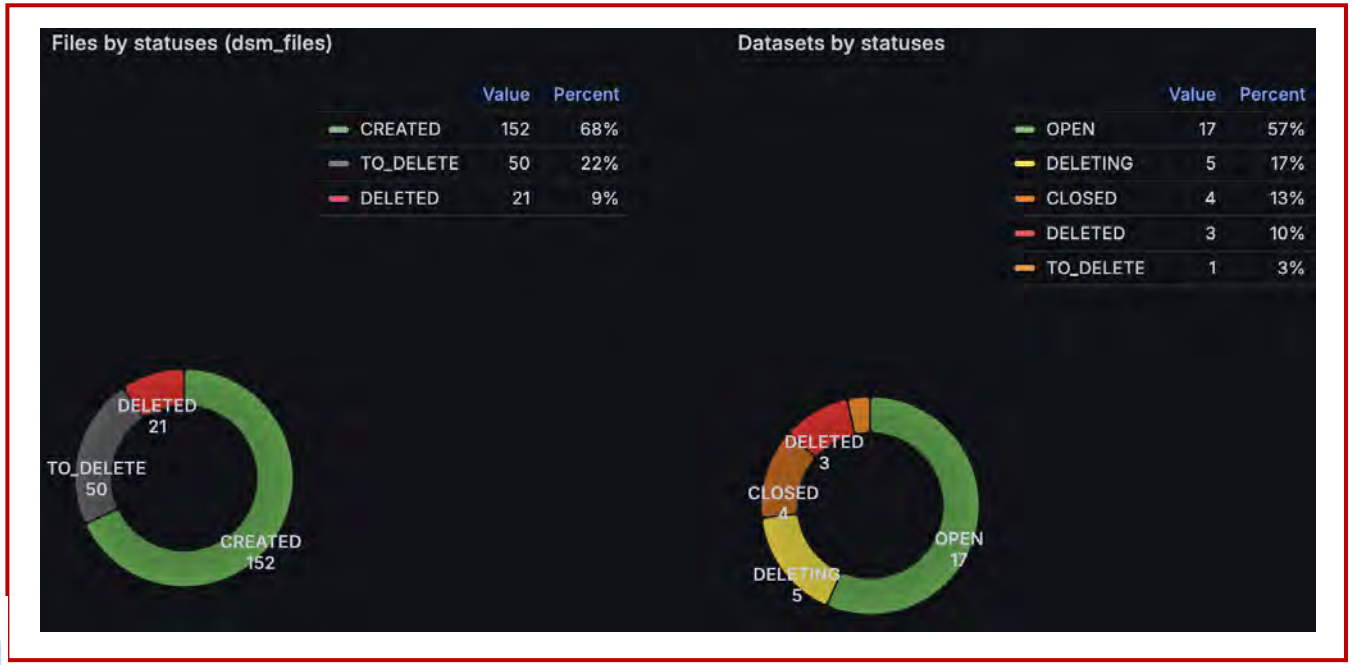
✓ All queues are fully operational



Dsm-manager: Update

The service should provide a REST API to the database

- ✓ Improved query efficiency
- ✓ Sending messages to monitoring is configured (for each data change operation)



Dsm-inspector: Update

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

- Deleting files on storages
  - Deleting datasets and files
  - Deleting dark files
- File integrity check
- Storage monitoring
  - Dark files monitoring
  - Monitoring storage usage
- File upload control
  - Setting tasks for uploading datasets
  - Uploading monitoring

✓ Improved efficiency of procedures

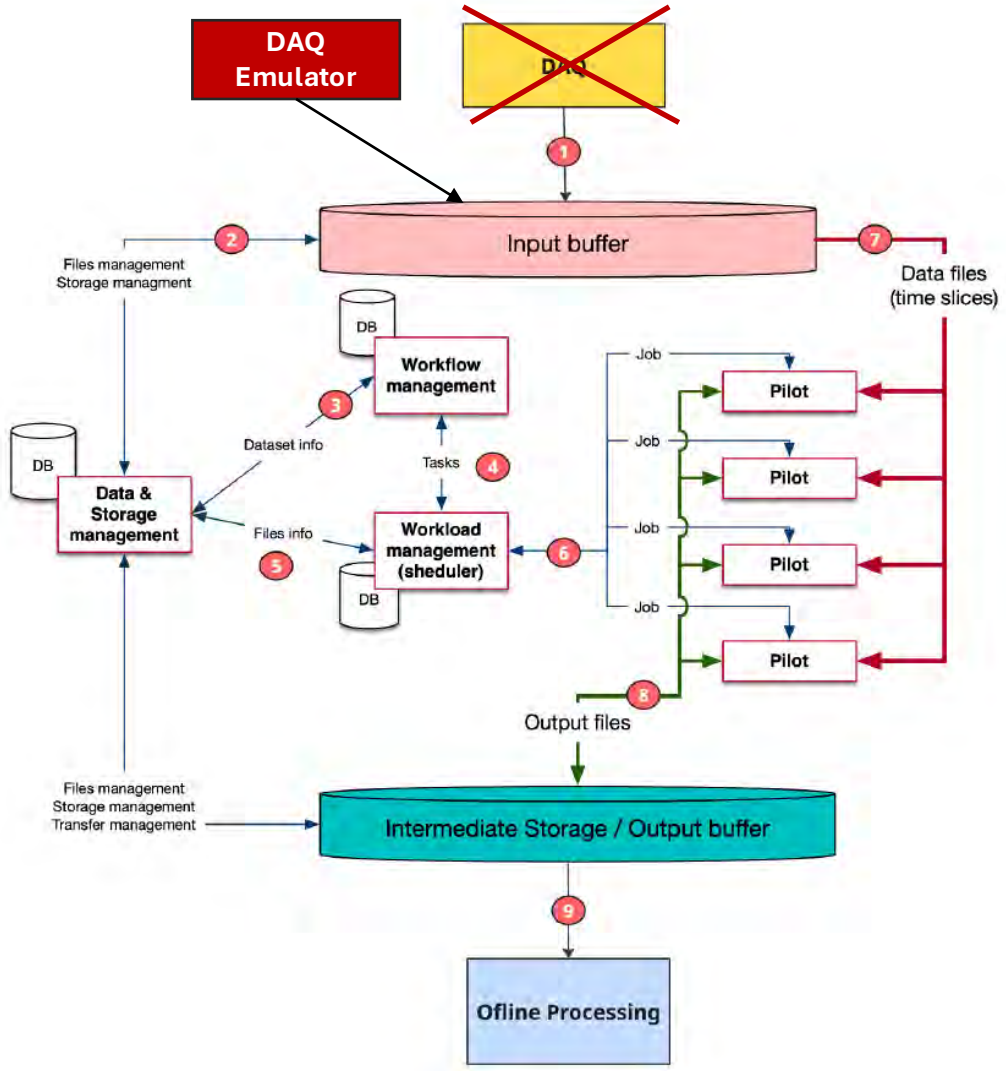
✓ pg_agent tasks are configured to perform scheduled procedures

✓ Deleting deleted data

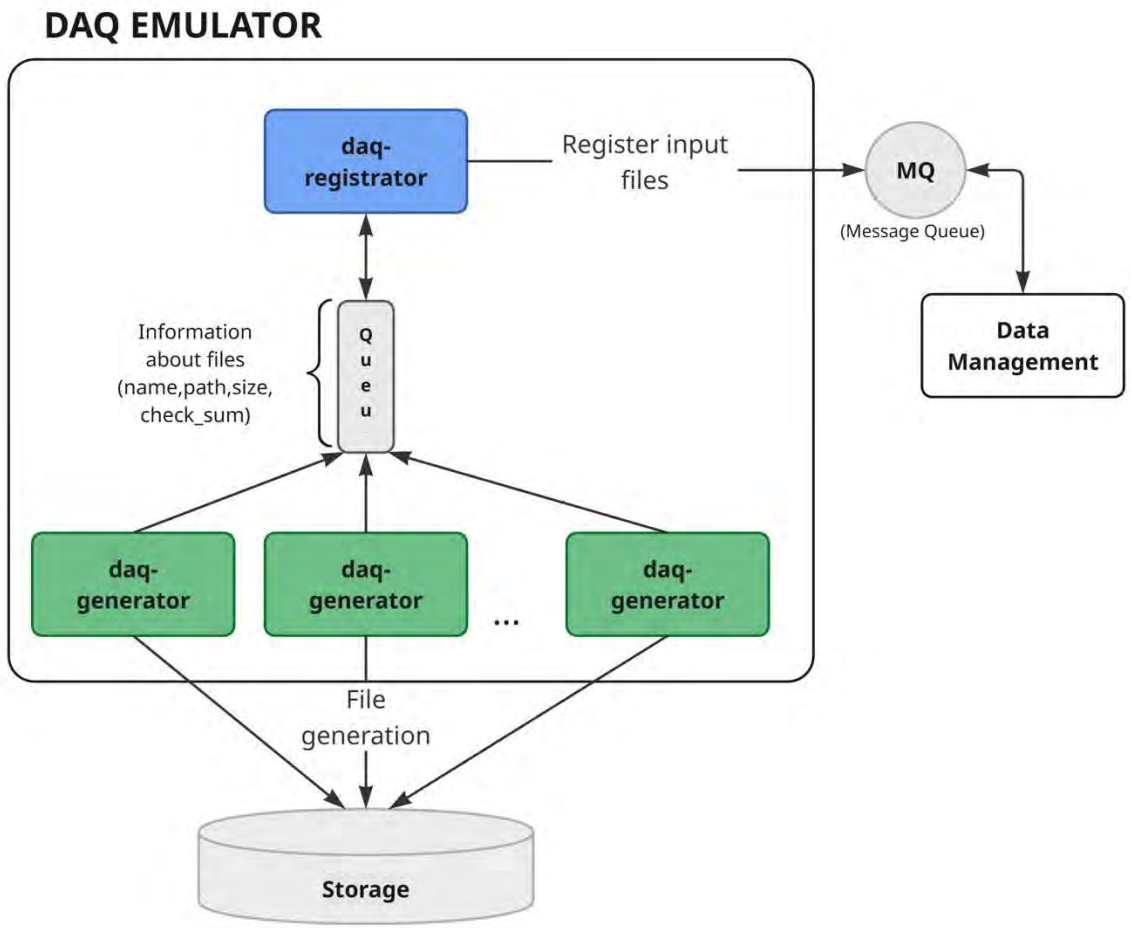
DAQ Emulator: Purpose

- Provide a continuous data stream for testing the SPD Online Filter Middleware
- Simulate the operation of the DAQ while real detector data is not available yet
- Enable load testing of the entire middleware

✓ Emulates the load, but not the content!



DAQ Emulator: Architecture



daq-generator

- creates files in the storage system and place corresponding information into queue, multiple generators can run simultaneously

daq-registrator

- register input files in DMS via a message broker when the number of accumulated files reaches the threshold required to form a dataset

Two independent processes communicating through a queue

DAQ Emulator: Configuration Parameters

config.ini

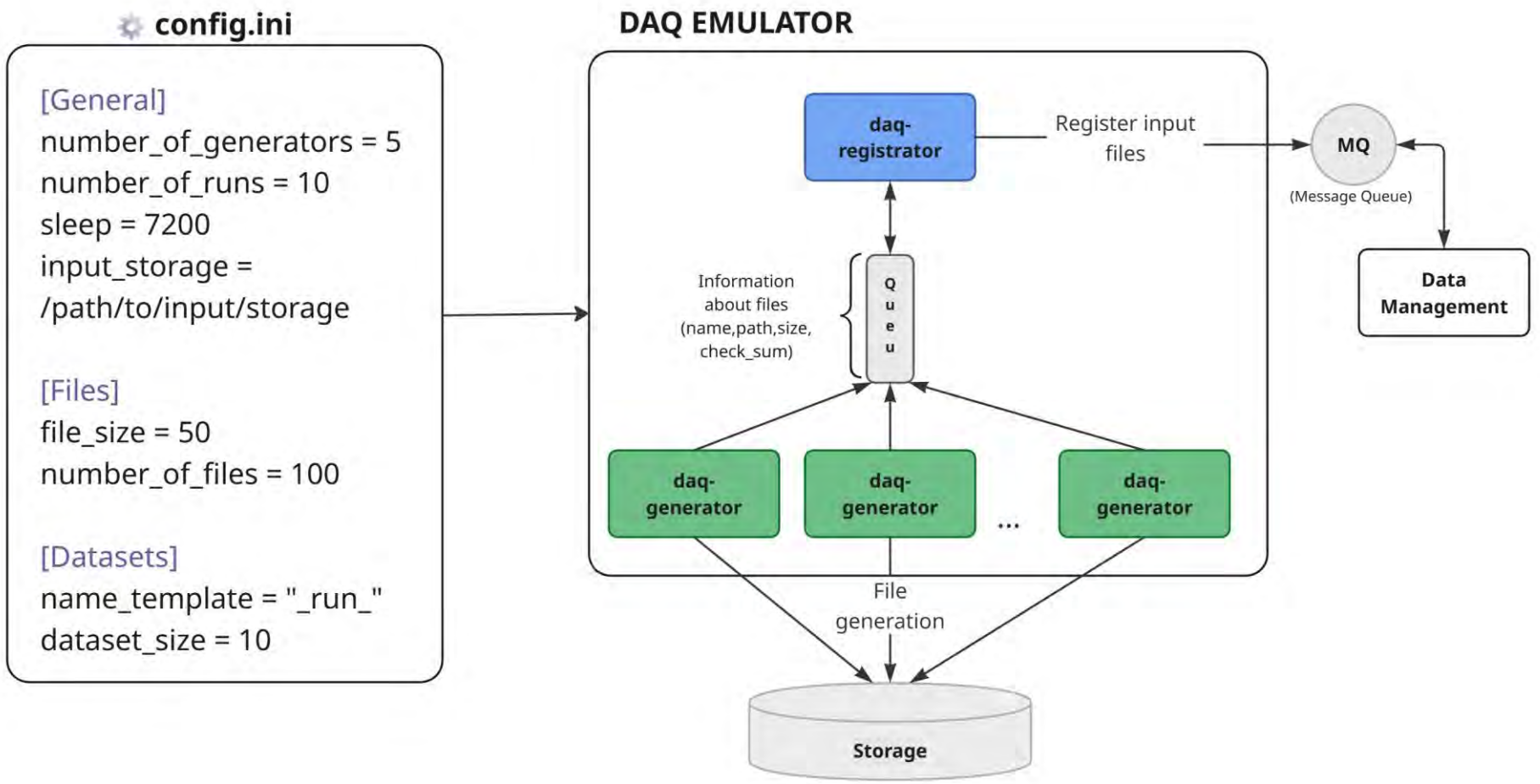
```
[General]
number_of_generators = 5
number_of_runs = 10
sleep = 7200
input_storage =
/path/to/input/storage

[Files]
file_size = 50
number_of_files = 100

[Datasets]
name_template = "_run_"
dataset_size = 10
```

Parameter	Description
Number of generators	How many daq-generator instances run simultaneously
Number of runs	How many generation cycles are performed
Number of files per run and their size	Volume of generated data
Dataset size	Number of files per dataset
Dataset name template	Naming convention for datasets

DAQ Emulator: Key Features



- ✓ Multiple parallel data generators
- ✓ Read the configuration file every new run (you can change the configuration without restarting the application)

DAQ Emulator: Key Features

- ✓ Provides an API for quick launch of the run
- ✓ Collecting metrics for monitoring

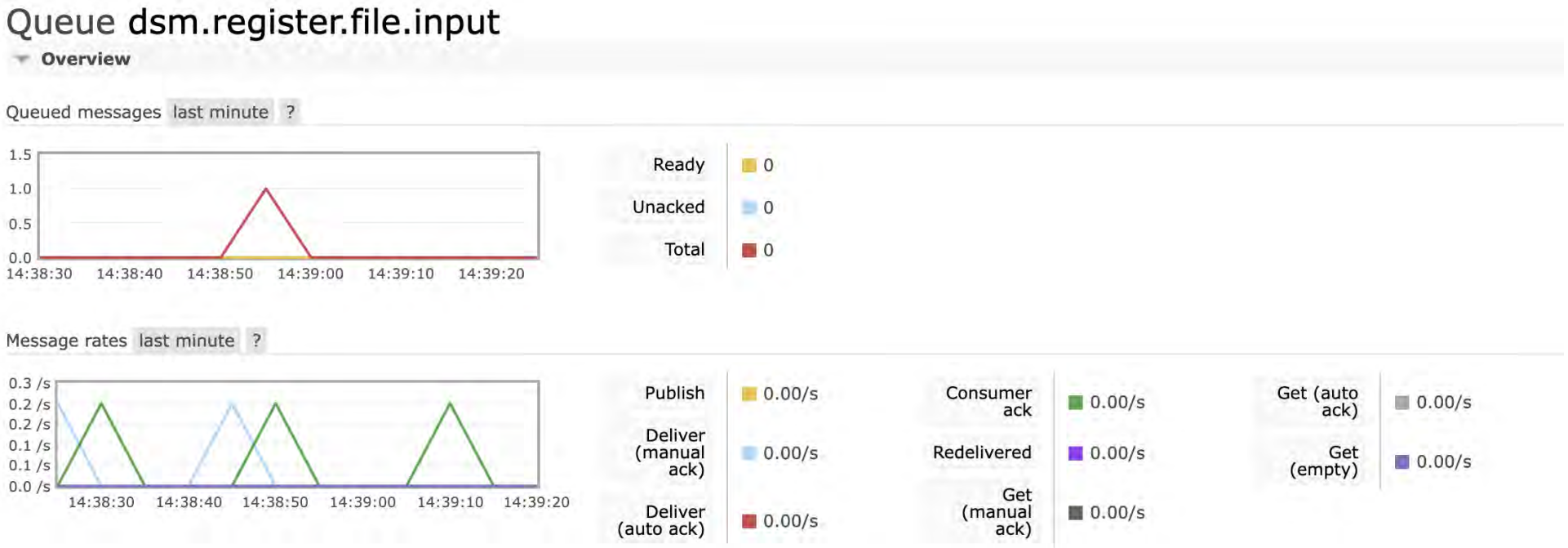
Load Testing

daq_data table

run	fileName	fileSize	sessionStart	File generation time
1	run-1-100-20260423_114144.dat	537 MB	2026-04-23_11:37:59	2.02 s
1	run-1-99-20260423_114138.dat	537 MB	2026-04-23_11:37:59	3.51 s
1	run-1-98-20260423_114138.dat	537 MB	2026-04-23_11:37:59	3.59 s
1	run-1-97-20260423_114138.dat	537 MB	2026-04-23_11:37:59	3.74 s
1	run-1-96-20260423_114132.dat	537 MB	2026-04-23_11:37:59	3.71 s
1	run-1-95-20260423_114131.dat	537 MB	2026-04-23_11:37:59	3.01 s
1	run-1-94-20260423_114131.dat	537 MB	2026-04-23_11:37:59	3.04 s
1	run-1-93-20260423_114126.dat	537 MB	2026-04-23_11:37:59	2.94 s
1	run-1-92-20260423_114126.dat	537 MB	2026-04-23_11:37:59	2.99 s
1	run-1-91-20260423_114125.dat	537 MB	2026-04-23_11:37:59	3.45 s

- Produce 10 datasets of 10 files
- File generation time ~ 3-4 seconds

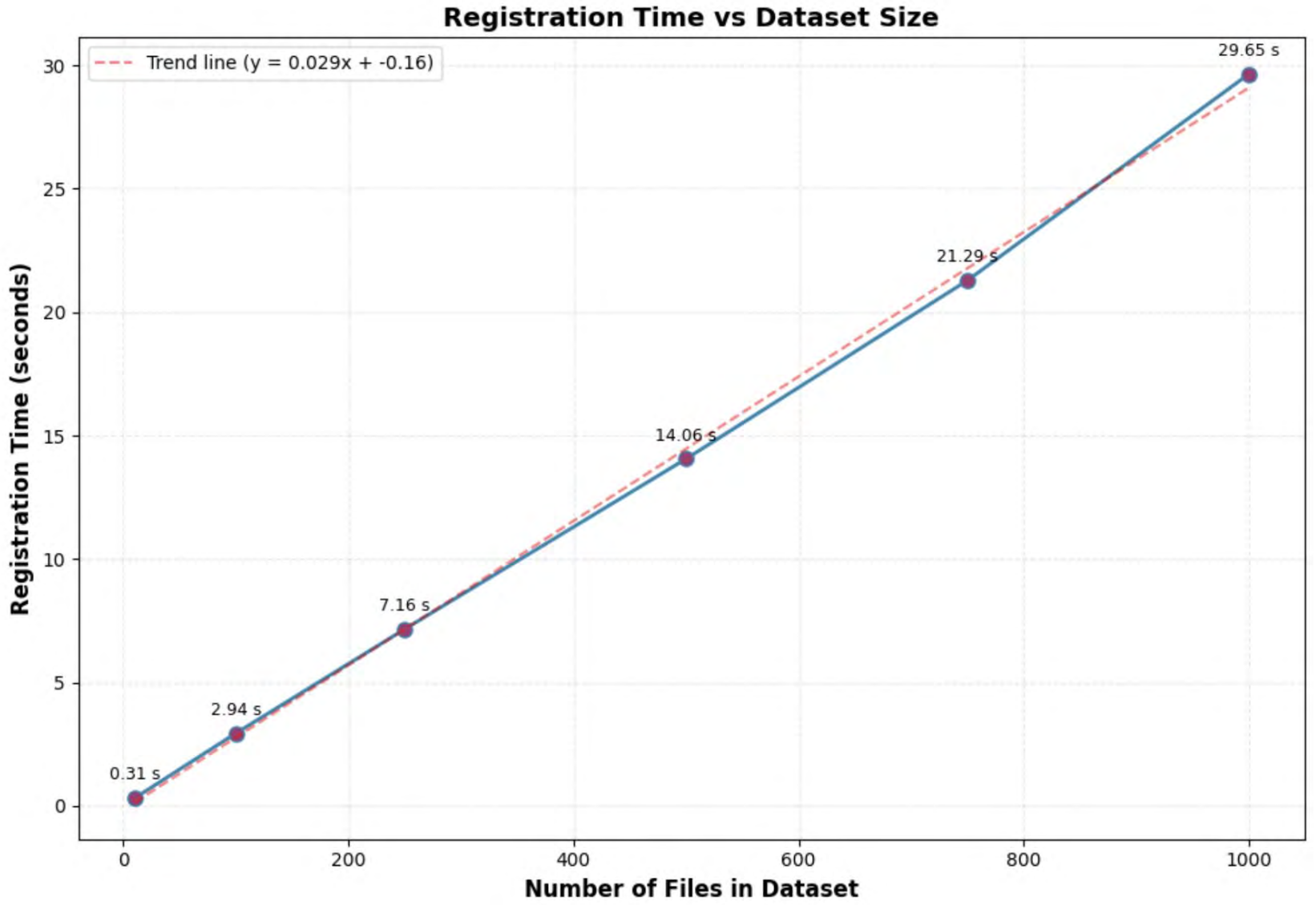
Load Testing




```
dsm-register  | 2026-04-23 11:38:30 INFO: Registration time = 0.32 sec
dsm-register  | 2026-04-23 11:38:50 INFO: Registration time = 0.31 sec
dsm-register  | 2026-04-23 11:39:11 INFO: Registration time = 0.3 sec
dsm-register  | 2026-04-23 11:39:52 INFO: Registration time = 0.35 sec
dsm-register  | 2026-04-23 11:39:55 INFO: Registration time = 0.31 sec
dsm-register  | 2026-04-23 11:40:18 INFO: Registration time = 0.3 sec
dsm-register  | 2026-04-23 11:40:47 INFO: Registration time = 0.29 sec
dsm-register  | 2026-04-23 11:41:09 INFO: Registration time = 0.31 sec
dsm-register  | 2026-04-23 11:41:29 INFO: Registration time = 0.31 sec
dsm-register  | 2026-04-23 11:41:51 INFO: Registration time = 0.28 sec
```

- Send message to dsm-register
- Dsm-register creates a new dataset and registers files
- Registration time for one dataset ~ 0.3 seconds

Dataset Registration Time



Conclusion

- **DAQ Emulator**
  - ✓ Load generation for testing the entire SPD Online Filter
  - ✓ Parallel file generation
  - ✓ Provides an API for quick launch of the run
  - ✓ Collecting metrics for monitoring
  - ✓ Switch to python 3.14 for the possibility of using true parallelism -> 512 MB file generation time decreased from 3-4s to 2-3s

Conclusion

- **Data Management System**
  - ✓ Dsm-register and dsm-manager fully functional
  - ✓ Add history tables to track all data states and configure automatic partitioning
  - ✓ pg_agent tasks are configured to perform scheduled procedures
  - ✓ Collecting metrics for monitoring
  - ✓ Deleting files and datasets, checking data consistency, monitoring storage for dark files, and monitoring storage usage successfully implemented
  - ✓ Ongoing work on uploading data to external storage

Next steps

- ❑ Configure authorization in dsm-manager
- ❑ Implement file upload control for dsm-inspector (waiting RSE to start working)