SPD Online Filter Middleware Status Update

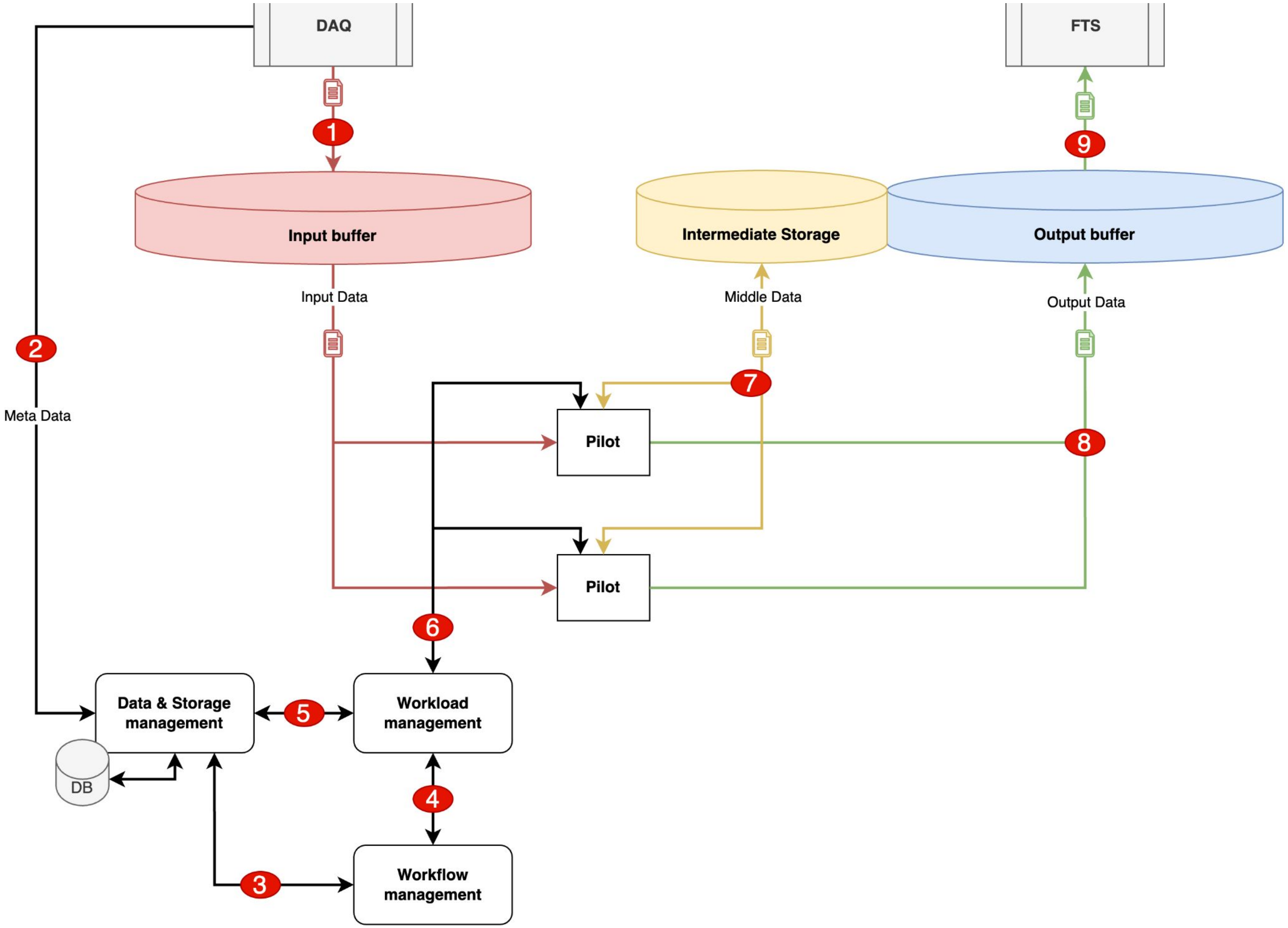
Nikita Greben

Joint Institute for Nuclear Research, MLIT, Dubna

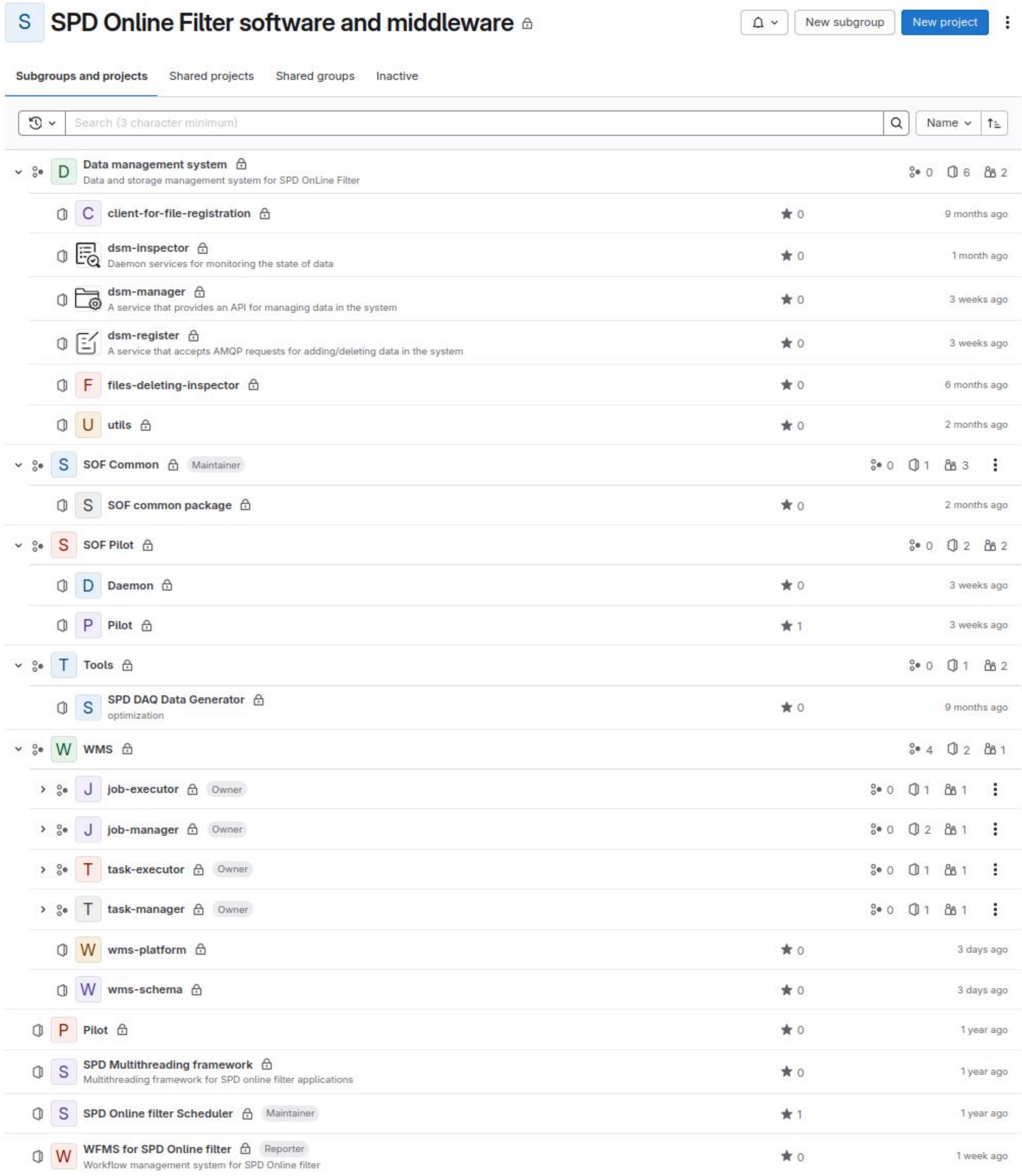
IX SPD Collaboration Meeting
AANL Yerevan
14.05.2025

Reminder: main components

- ❖ **Data & Storage Management (Polina Korshunova - master graduate)**
  - ➢ Data lifecycle support (data catalog, consistency check, cleanup, storage);
- ❖ **Workflow Management System (Artem Plotnikov - master graduate)**
  - ➢ Define and execute processing chains by generating the required number of computational tasks;
- ❖ **Workload management system (Nikita Greben, Leonid Romanychev):**
  - ➢ Create the required number of processing jobs to perform the task;
  - ➢ Control job execution through pilots working on compute nodes;



SPD Online Filter middleware code base



SPD Online Filter software and middleware

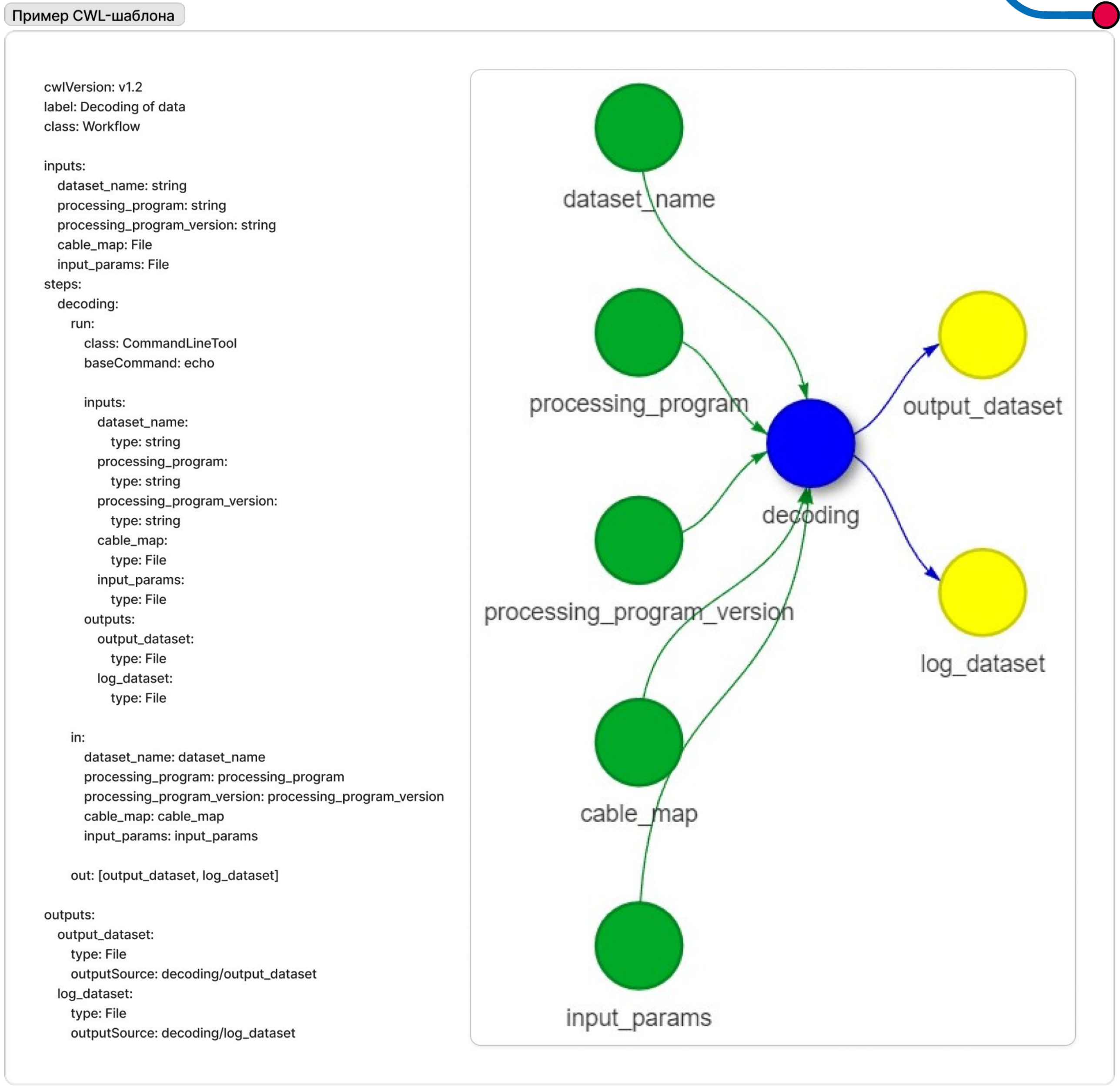
Data management system	Data and storage management system for SPD OnLine Filter		0	6	2		
client-for-file-registration						★ 0	9 months ago
dsm-inspector	Daemon services for monitoring the state of data					★ 0	1 month ago
dsm-manager	A service that provides an API for managing data in the system					★ 0	3 weeks ago
dsm-register	A service that accepts AMQP requests for adding/deleting data in the system					★ 0	3 weeks ago
files-deleting-inspector						★ 0	6 months ago
utils						★ 0	2 months ago
SOF Common		Maintainer	0	1	3		
SOF common package						★ 0	2 months ago
SOF Pilot			0	2	2		
Daemon						★ 0	3 weeks ago
Pilot						★ 1	3 weeks ago
Tools			0	1	2		
SPD DAQ Data Generator	optimization					★ 0	9 months ago
WMS			4	2	1		
job-executor		Owner	0	1	1		
job-manager		Owner	0	2	1		
task-executor		Owner	0	1	1		
task-manager		Owner	0	1	1		
wms-platform						★ 0	3 days ago
wms-schema						★ 0	3 days ago
Pilot						★ 0	1 year ago
SPD Multithreading framework	Multithreading framework for SPD online filter applications					★ 0	1 year ago
SPD Online filter Scheduler		Maintainer				★ 1	1 year ago
WFMS for SPD Online filter	Workflow management system for SPD Online filter	Reporter				★ 0	1 week ago

- ➢ Around ~25 000 lines of code for the entire SPD Online Filter Middleware;
- ➢ Full deployment requires ~16 Docker containers: one container per microservice;
- ➢ Configured CI/CD pipeline, currently only for the Workload Management System;
- ❗ May need to be reorganized to deploy as a standalone project on the testbed
  - ○ Hardware for the prototyping of a compute cluster
- ❗ Deploying Pilot Agents to Compute Nodes.

Workflow Management System - Core logic

The main objectives of Workflow Management System:

1. Retrieves input datasets from Data Management System;
2. Maps these datasets with the appropriate CWL template;
3. Generates the workchain from this template;
4. Generates tasks and sends them to the Workload Management System for further execution;
5. Oversees datasets: decision making for creation, closure, deletion;
6. Manages the concurrent execution of workchains and tasks.



Пример CWL-шаблона

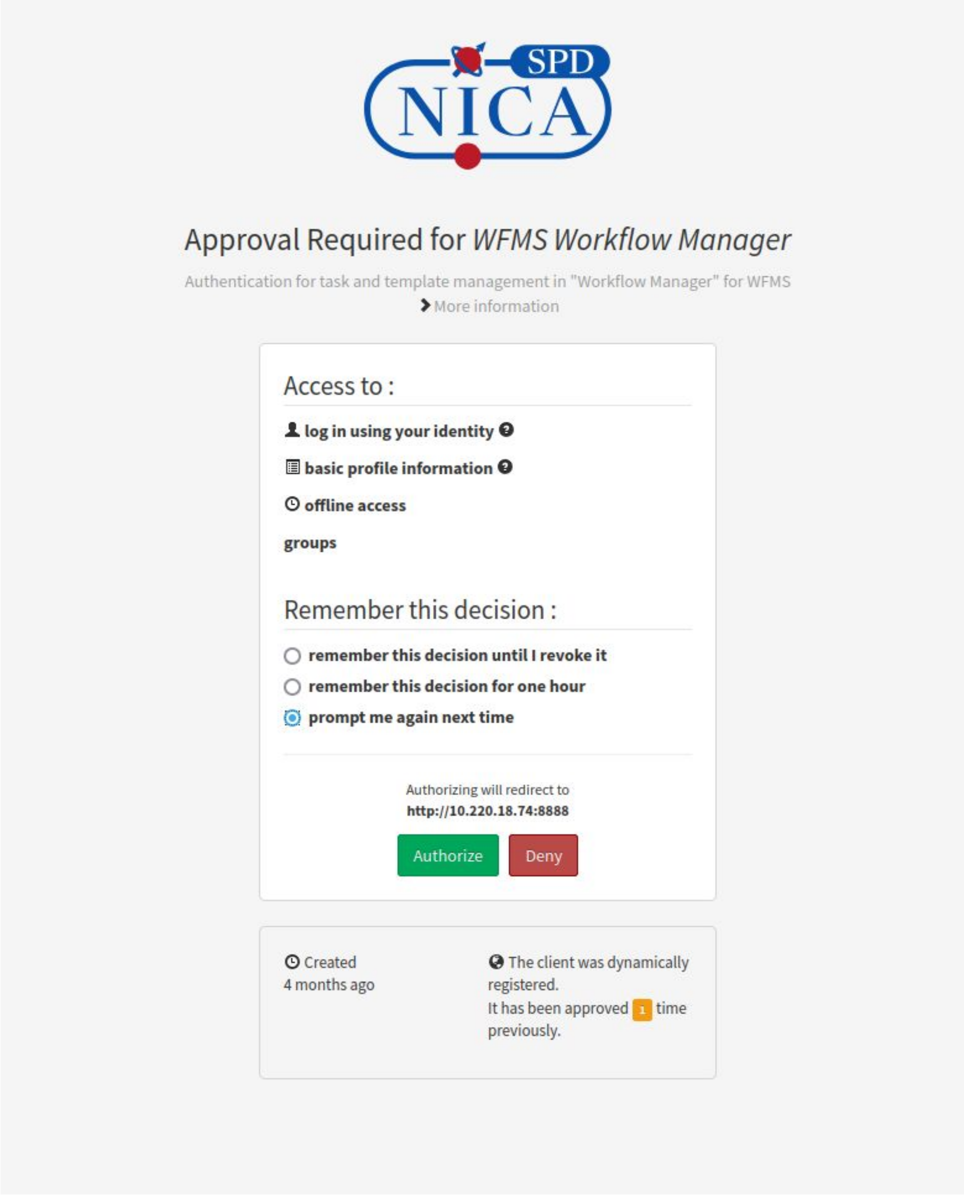
```
cwlVersion: v1.2
label: Decoding of data
class: Workflow

inputs:
  dataset_name: string
  processing_program: string
  processing_program_version: string
  cable_map: File
  input_params: File
steps:
  decoding:
    run:
      class: CommandLineTool
      baseCommand: echo

      inputs:
        dataset_name:
          type: string
        processing_program:
          type: string
        processing_program_version:
          type: string
        cable_map:
          type: File
        input_params:
          type: File
      outputs:
        output_dataset:
          type: File
        log_dataset:
          type: File

    in:
      dataset_name: dataset_name
      processing_program: processing_program
      processing_program_version: processing_program_version
      cable_map: cable_map
      input_params: input_params

    out: [output_dataset, log_dataset]

outputs:
  output_dataset:
    type: File
    outputSource: decoding/output_dataset
  log_dataset:
    type: File
    outputSource: decoding/log_dataset
```

Workflow Management System Update

1. Rewritten to take advantage of asynchronous features;
2. Added the ability to clone templates;
3. Add support for loading a template from a file;
4. Added possibility to delete a template in LOADED status;
5. The internal authorization system has been abandoned and integration with SPD-IAM has been performed;

✅ Implemented the service to interact with the Workload Management System.

Debugging the interaction with the **Workload Management System**.



Approval Required for *WFMS Workflow Manager*

Authentication for task and template management in "Workflow Manager" for WFMS

› More information

Access to :

log in using your identity

basic profile information

offline access

groups

Remember this decision :

○ remember this decision until I revoke it

○ remember this decision for one hour

● prompt me again next time

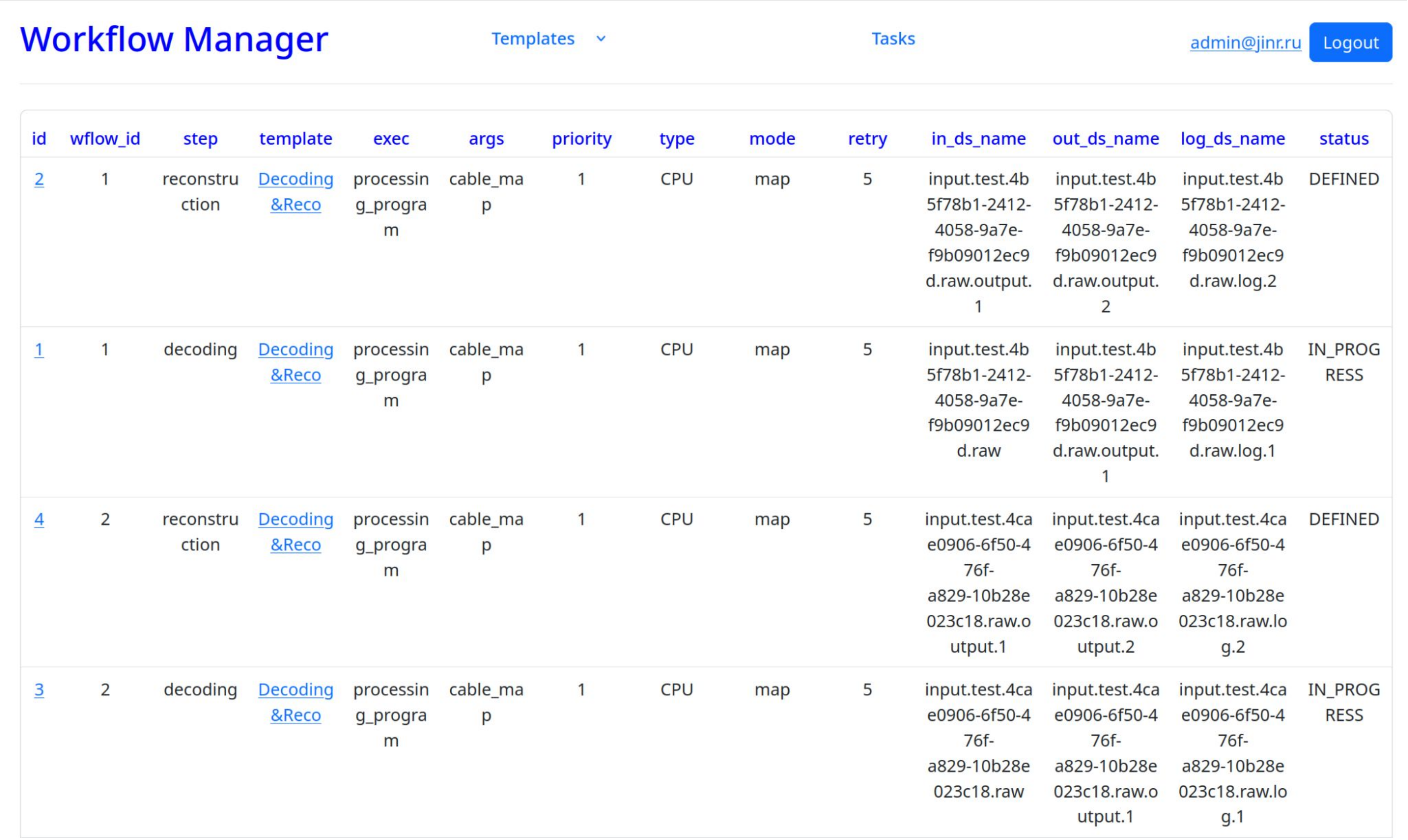
Authorizing will redirect to
http://10.220.18.74:8888

Created
4 months ago

The client was dynamically registered.
It has been approved 1 time previously.

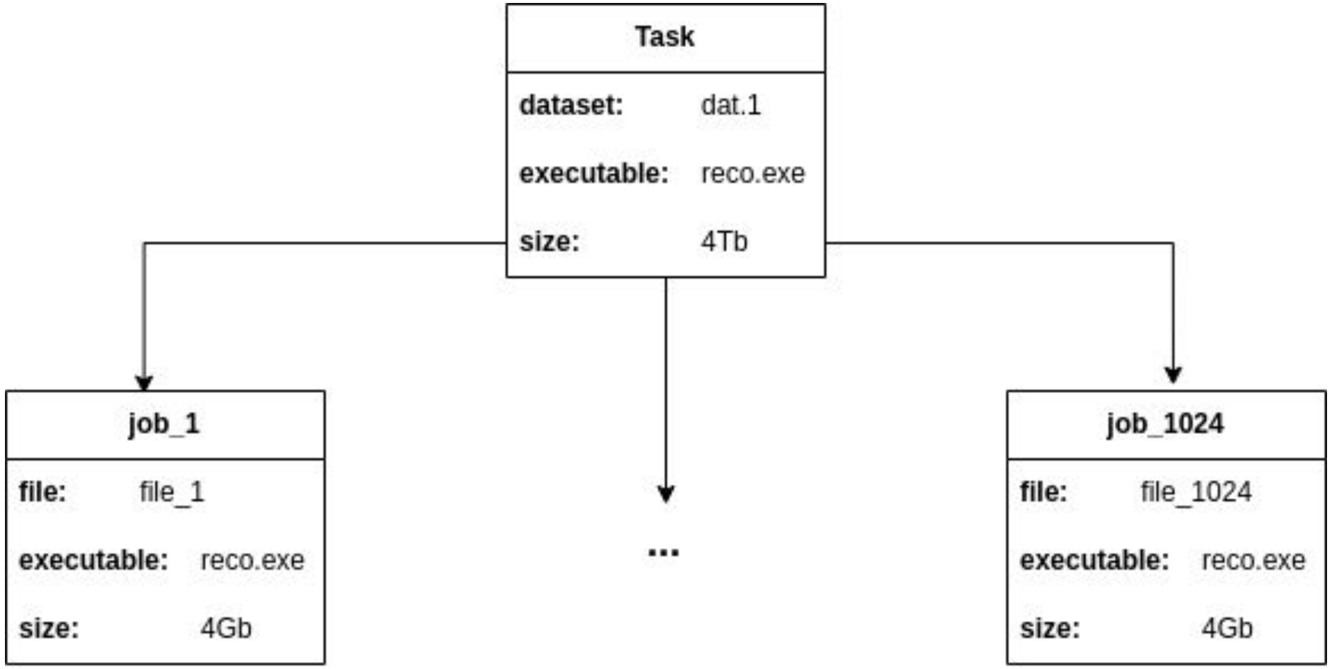
Integration with SPD-IAM

Workflow Management System Update

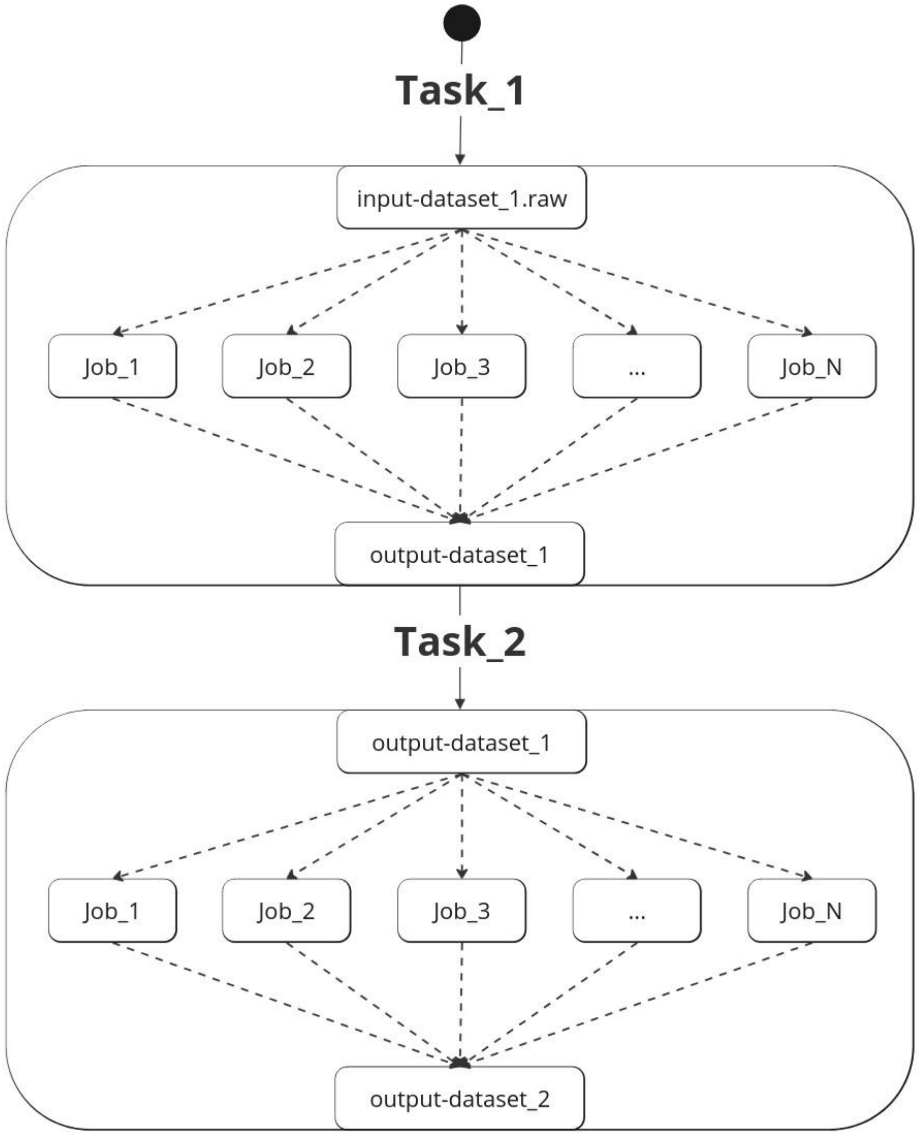


id	wflow_id	step	template	exec	args	priority	type	mode	retry	in_ds_name	out_ds_name	log_ds_name	status
2	1	reconstruction	Decoding&Reco	processing_program	cable_map	1	CPU	map	5	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw.output.1	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw.output.2	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw.log.2	DEFINED
1	1	decoding	Decoding&Reco	processing_program	cable_map	1	CPU	map	5	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw.output.1	input.test.4b5f78b1-2412-4058-9a7e-f9b09012ec9d.raw.log.1	IN_PROGRESS
4	2	reconstruction	Decoding&Reco	processing_program	cable_map	1	CPU	map	5	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw.output.1	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw.output.2	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw.log.2	DEFINED
3	2	decoding	Decoding&Reco	processing_program	cable_map	1	CPU	map	5	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw.output.1	input.test.4cae0906-6f50-476f-a829-10b28e023c18.raw.log.1	IN_PROGRESS

Task-job relationship (reminder)

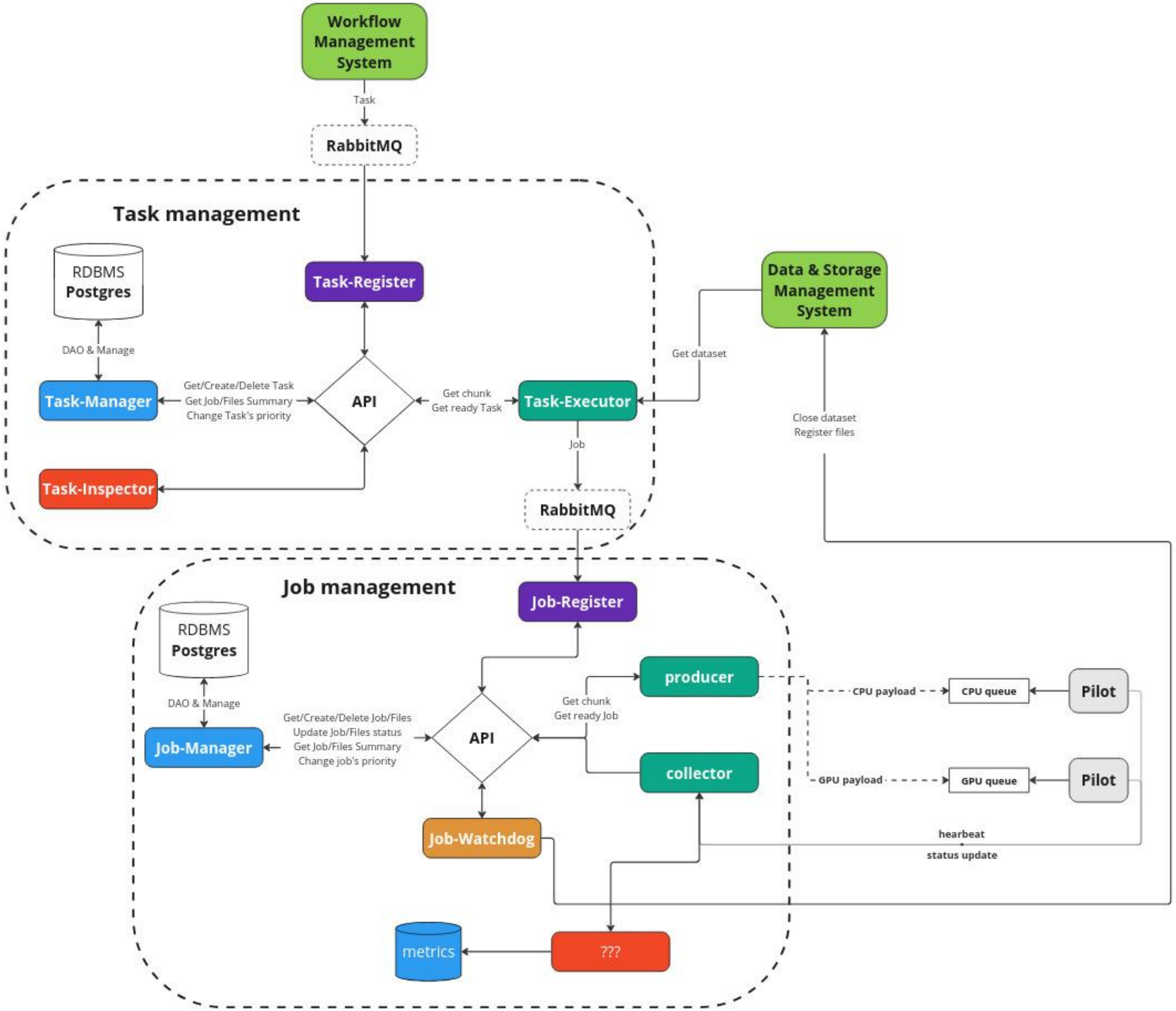


Task-job relationship



Data processing workflow example

Workload Management System Major Update



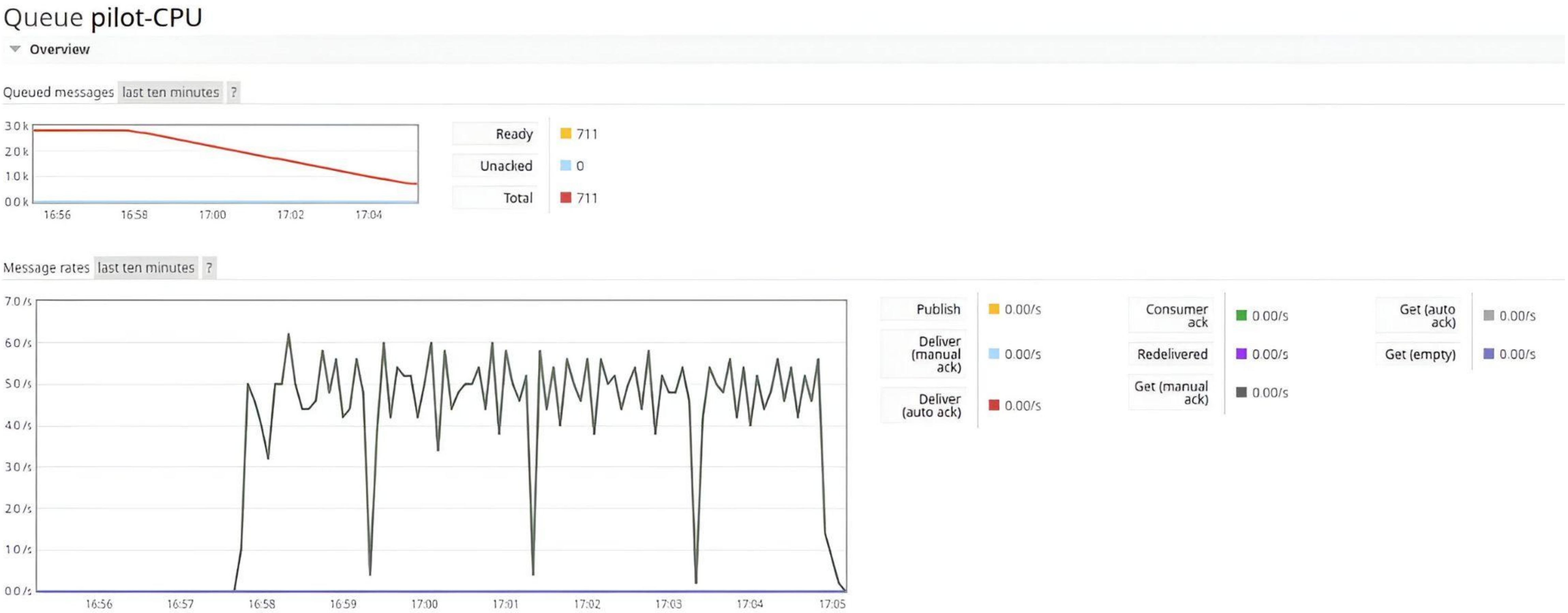
- ➢ Task-Register and Job-Register were added;
- ➢ Major refactoring of Task-Management and Job-Management;
- ➢ Rewritten using dependency injection approach (easy to maintain and evolve);
- ➢ Producer/Collector services completely reworked;
- ✓ Implemented task-executor (**first approximation scheduler**);
- ✓ Launched an execution of one task across the system;
- ✓ Implemented the task-inspector/watchdog service.

Next steps:

1. Implement metric-collection service;
2. Major refactoring and testing is needed;
3. Monitoring service, traces collection

First “load testing”

1. 100 concurrently running pilots
2. ~2100 jobs completed in 7 min
3. Pilot works for ~15 seconds



Queue **pilot-CPU**

▾ **Overview**

Queued messages last ten minutes ?

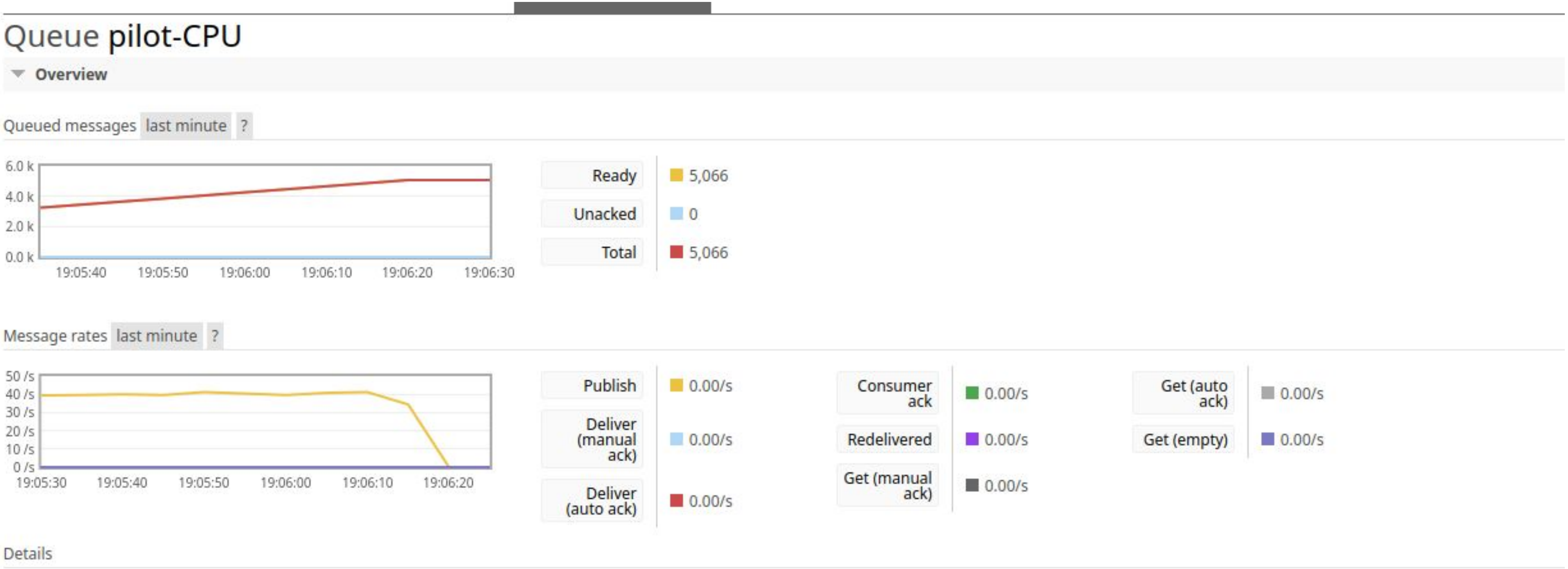
Ready	🟨 711
Unacked	🟦 0
Total	🟥 711

Message rates last ten minutes ?

Publish	🟨 0.00/s	Consumer ack	🟩 0.00/s	Get (auto ack)	■ 0.00/s
Deliver (manual ack)	🟦 0.00/s	Redelivered	🟪 0.00/s	Get (empty)	🟪 0.00/s
Deliver (auto ack)	🟥 0.00/s	Get (manual ack)	■ 0.00/s		

First “load testing”

1. Workload Management System generates ~5000 jobs in less than a minute
2. Must be tested on meaningful data and payload, the system may not need to be over engineered more



Queue pilot-CPU

▼ Overview

Queued messages last minute ?

Ready	🟨 5,066
Unacked	🟦 0
Total	🟥 5,066

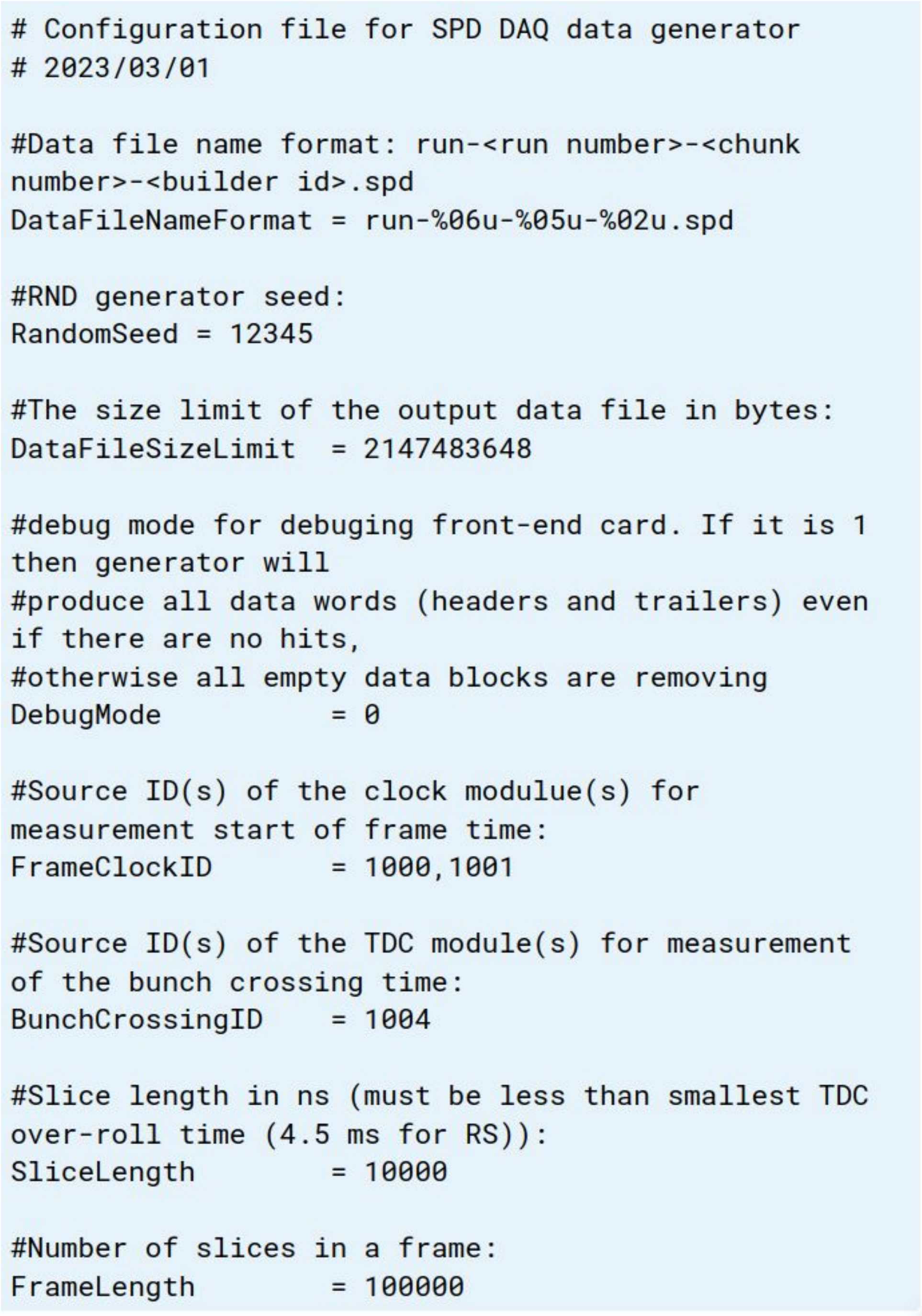
Message rates last minute ?

Publish	🟨 0.00/s	Consumer ack	🟩 0.00/s	Get (auto ack)	⬜ 0.00/s
Deliver (manual ack)	🟦 0.00/s	Redelivered	🟪 0.00/s	Get (empty)	🟪 0.00/s
Deliver (auto ack)	🟥 0.00/s	Get (manual ack)	⬛ 0.00/s		

Details

DAQ data generator

1. Using **SPD DAQ Data Generator**, we've generated **50 files**, each ~**2Gb**;
2. Input dataset has been registered with these files;
3. Task has been processed (or 50 jobs);
4. The payload for **Pilot** is simple: compute the MD5/BLAKE3 hash, as there is no actual computation involved at this stage.;
5. Takes about ~**7 min** to generate a file, using JINR Cloud VM: 12x 1-core Intel Xeon E5-2650
6. Registration of the entire dataset: ~**10 sec**



```
# Configuration file for SPD DAQ data generator
# 2023/03/01

#Data file name format: run-<run number>-<chunk
number>-<builder id>.spd
DataFileNameFormat = run-%06u-%05u-%02u.spd

#RND generator seed:
RandomSeed = 12345

#The size limit of the output data file in bytes:
DataFileSizeLimit  = 2147483648

#debug mode for debuging front-end card. If it is 1
then generator will
#produce all data words (headers and trailers) even
if there are no hits,
#otherwise all empty data blocks are removing
DebugMode            = 0

#Source ID(s) of the clock modulue(s) for
measurement start of frame time:
FrameClockID         = 1000,1001

#Source ID(s) of the TDC module(s) for measurement
of the bunch crossing time:
BunchCrossingID      = 1004

#Slice length in ns (must be less than smallest TDC
over-roll time (4.5 ms for RS)):
SliceLength          = 10000

#Number of slices in a frame:
FrameLength          = 100000
```

Next steps/milestones

✅ **Task and workflow processing has been achieved**

- ❑ Execution of the entire workflow set up on the level of **Workflow Management System**
- ❑ The entire workflow - a chain of dependent tasks
- ❗ The major cycle of refactoring and test coverage is required

❓ **Middleware and applied software integration**

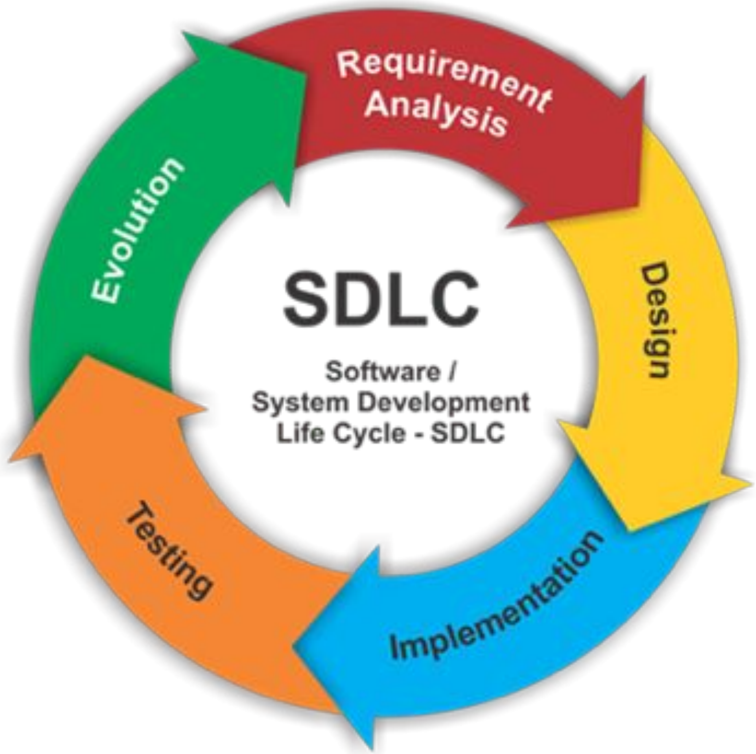
- ❑ Requires prototyped applied software and simulated data
- ❑ Non-functional requirements for applied software
- ❑ Move to the execution of the jobs on the pilot with a "real" payload

❓ **Middleware deployment and release management**

- ❑ Focus on shipping SPD Online Filter as standalone software
- ❑ Work on the deployment on the **upcoming testbed** (256 CPU Cores, 1TB RAM, 120TB HDD)
- ❑ Select the appropriate release management strategy

Next major steps

- ❑ **Distributed tracing**
  - ❑ Monitor and track the path of requests as they pass through multiple, interconnected microservices within SPD Online Filter.
- ❑ **Logging**
  - ❑ Currently, each microservice logs are mapped to the host via a shared file system between Docker and the host.
  - ❑ Ideally – **ELK** (*Elastic-Logstash-Kibana*) stack to build a log analysis platform.
- ❑ **Configuration**
  - ❑ Consider to centralize some of the shared configurations across multiple services (*Consul, Etcd*), using Gitlab Secrets for now.
- ❑ **Metrics and monitoring**
  - ❑ For example, service query-per-second, API responsiveness, service latency etc. (*InfluxDB, Prometheus, Graphana*)
- ❑ **Documentation**
  - ❑ Given the increasing complexity of the internal logic of the software, it is necessary to document each step of the development.



Never ending cycle

Future plans

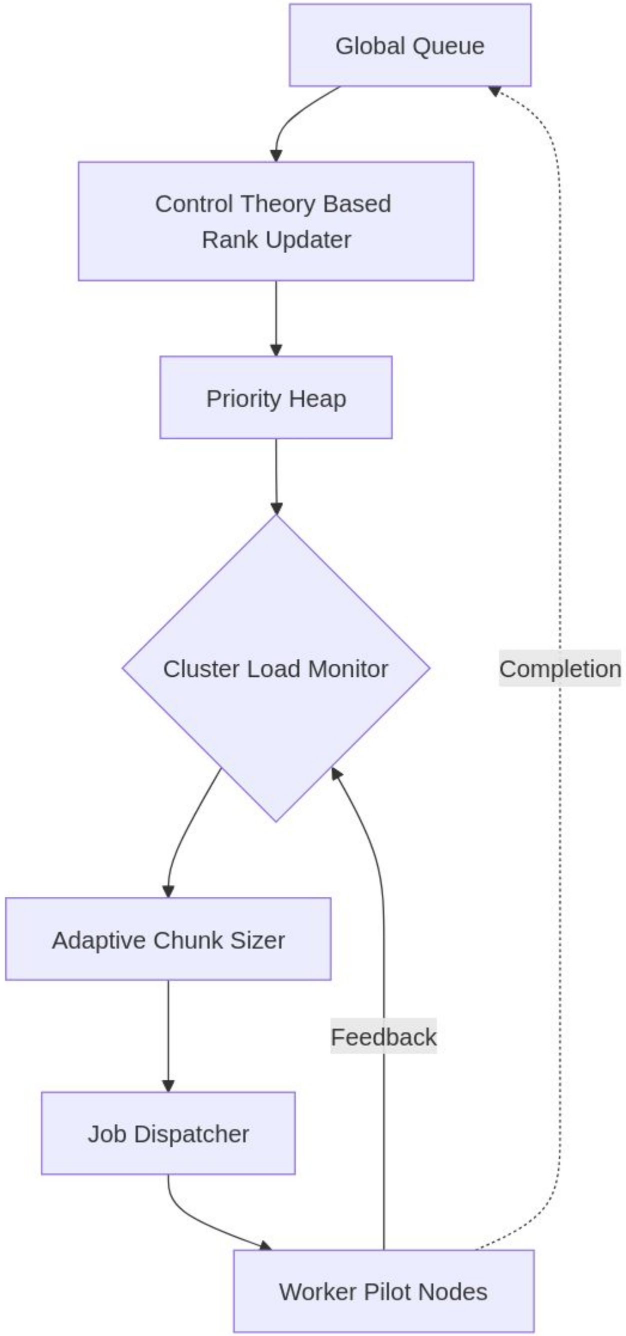
Task-executor (Scheduler)

1. Expected to process tasks from a global queue;
2. Each dataset has a rank (priority) that determines its processing order;
3. Tasks are processed in priority order, with dynamic updates to maintain system responsiveness;
4. **Priority-based task scheduling mechanism** is expected, with rank update scheme involving **Control Theory** (option to be explored later);
5. Not applicable at this stage of the development process.

$$\mathrm{r}_{i+1} = \underbrace{\alpha \ln(x_i + 1)}_{\text{Aging}} - \underbrace{\beta 2^{y_i}}_{\text{Retry Penalty}} + \underbrace{\gamma r_i}_{\text{History}} + \underbrace{\delta(1 - L)}_{\text{Load}}$$

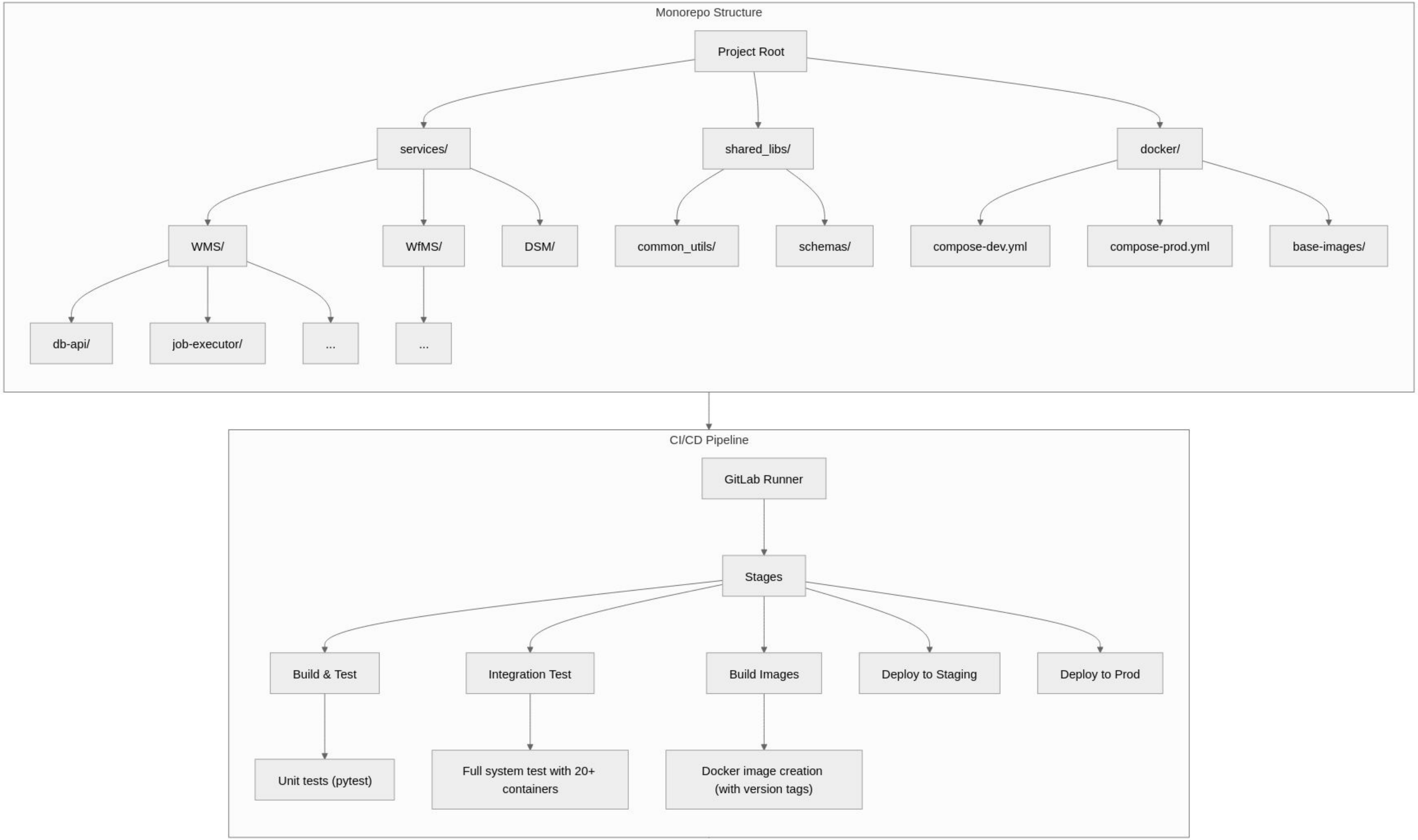
$$\mathbf{r}_{i+1} = \Gamma \mathbf{r}_i + \alpha \ln(\mathbf{x}_i + \mathbf{1}) - \beta \cdot 2^{\mathbf{y}_i} + \delta(1 - L)\mathbf{1}$$

$$\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_N) \text{ (job-specific history weights)}$$
$$\mathbf{x}_i = [x_i^{(1)}, \ldots, x_i^{(N)}]^\top \text{ (job ages)}$$
$$\mathbf{y}_i = [y_i^{(1)}, \ldots, y_i^{(N)}]^\top \text{ (retry counts)}$$

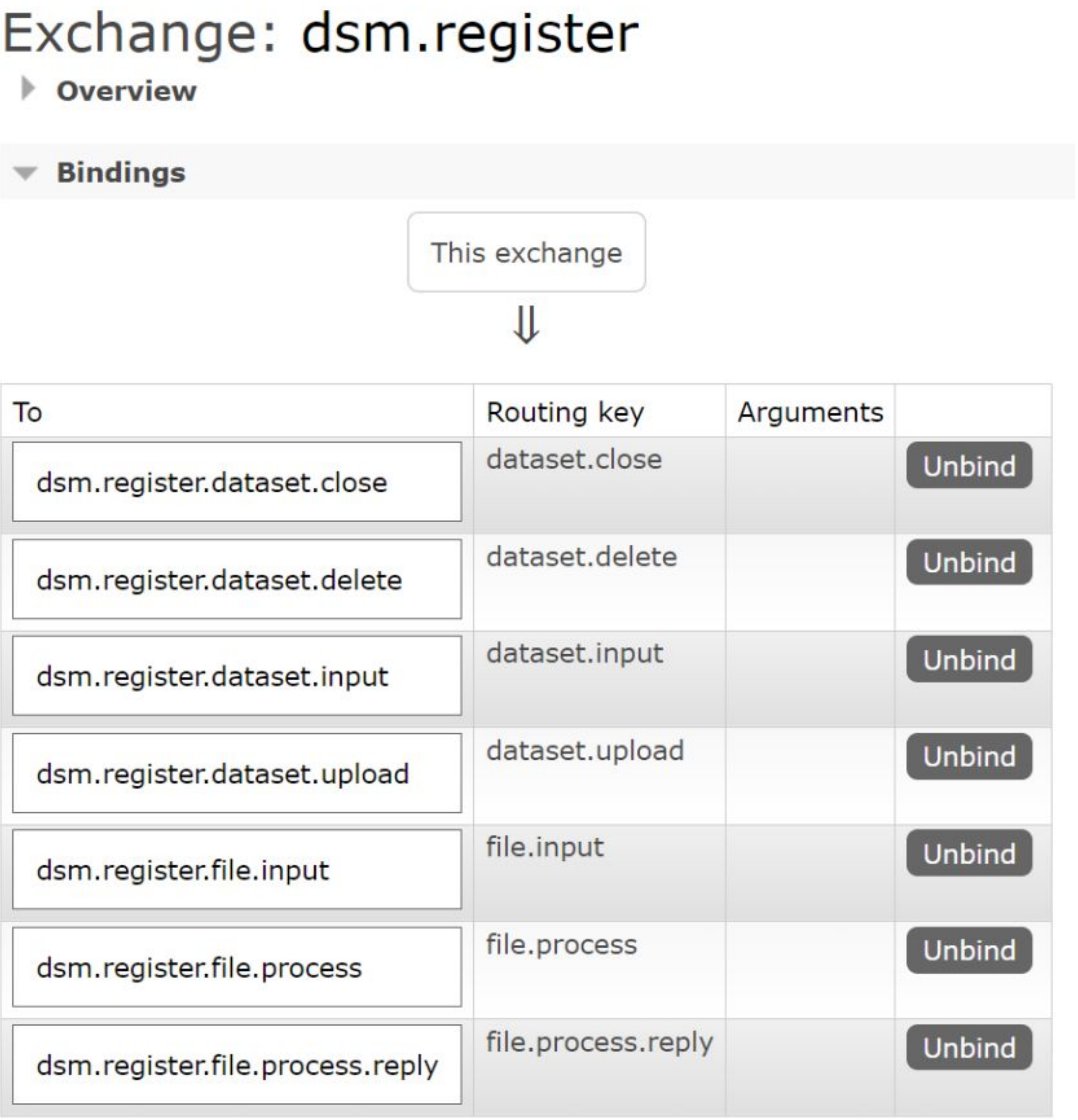


Backup slides

Gitlab project structure for CI/CD



RabbitMQ configured queues



Exchange: dsm.register

Overview

Bindings

This exchange

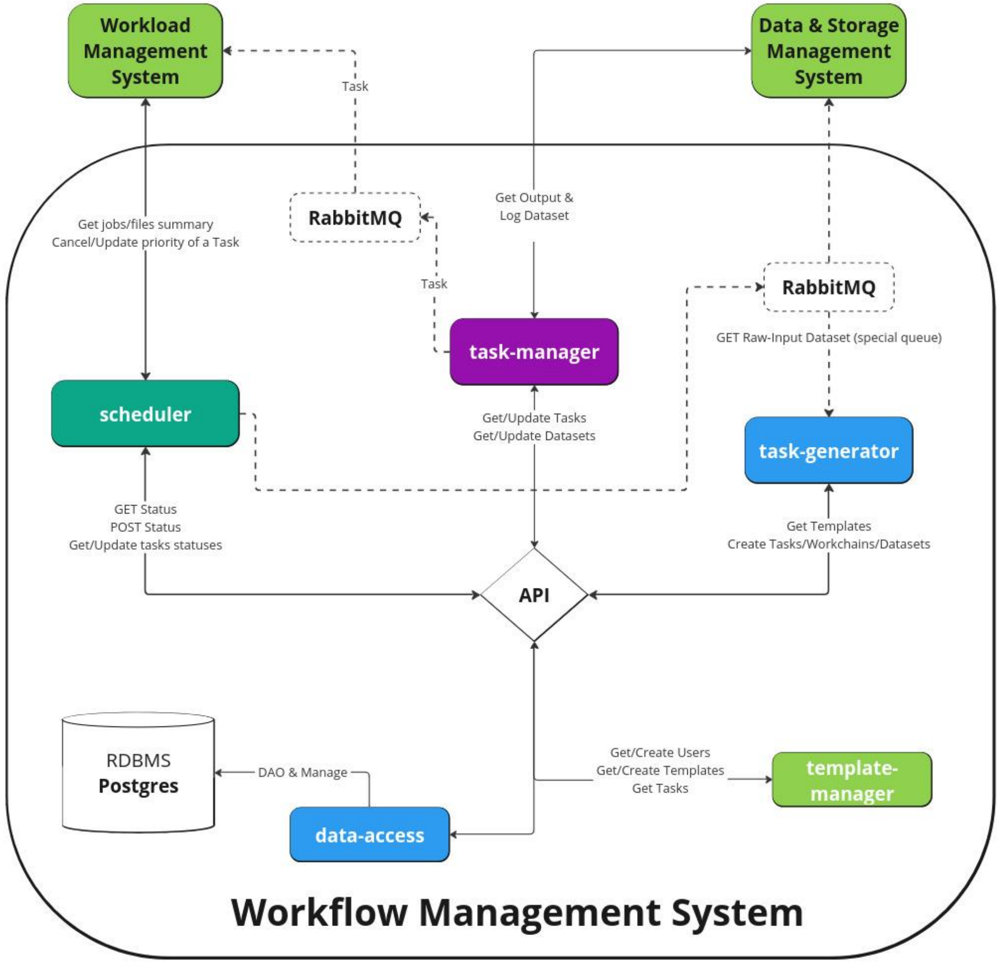
⇓

To	Routing key	Arguments	
dsm.register.dataset.close	dataset.close		Unbind
dsm.register.dataset.delete	dataset.delete		Unbind
dsm.register.dataset.input	dataset.input		Unbind
dsm.register.dataset.upload	dataset.upload		Unbind
dsm.register.file.input	file.input		Unbind
dsm.register.file.process	file.process		Unbind
dsm.register.file.process.reply	file.process.reply		Unbind

Exchange	Routing Key	Appointment
dsm.register (direct)	file.input	Receiving information about incoming files to the input buffer
	file.process	Receiving information about new files, received during processing
	dataset.close	Accepting a request to close a dataset
	dataset.upload	Accepting an application to upload files in a dataset to an external storage
	dataset.delete	Accepting a request to delete files in a dataset on the internal storage

Workflow Management System

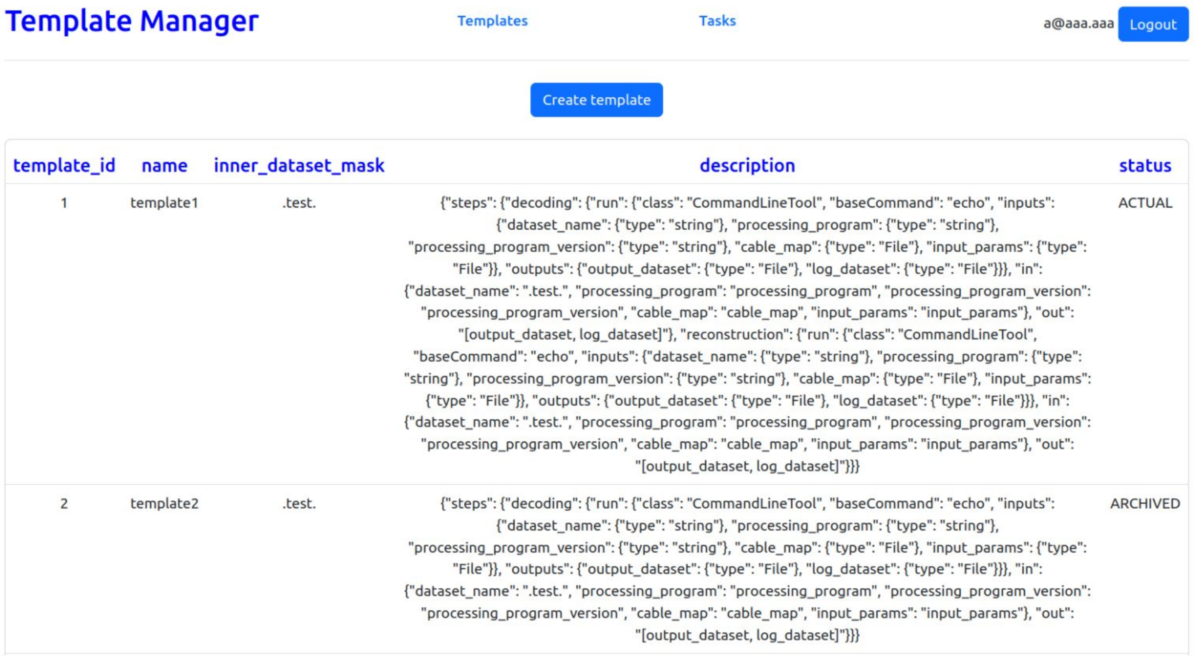
- ➢ **task-manager** – a service that requests the last dataset created in the previous step of the workflow chain, populates it, and sends the next task to the WMS.
- ➢ **task-generator** – responsible for starting the workflows based on the available templates**.**
- ➢ **template-manager** – service for interaction with the data processing operator/user**.**
- ➢ **data access** – a service that encapsulates direct database access, provides a RESTful API's through endpoints**.**
- ➢ **scheduler** – a services responsible for making decision on when to close datasets, cancel or change a priority of a task.



Workflow Management System High-Level Architecture

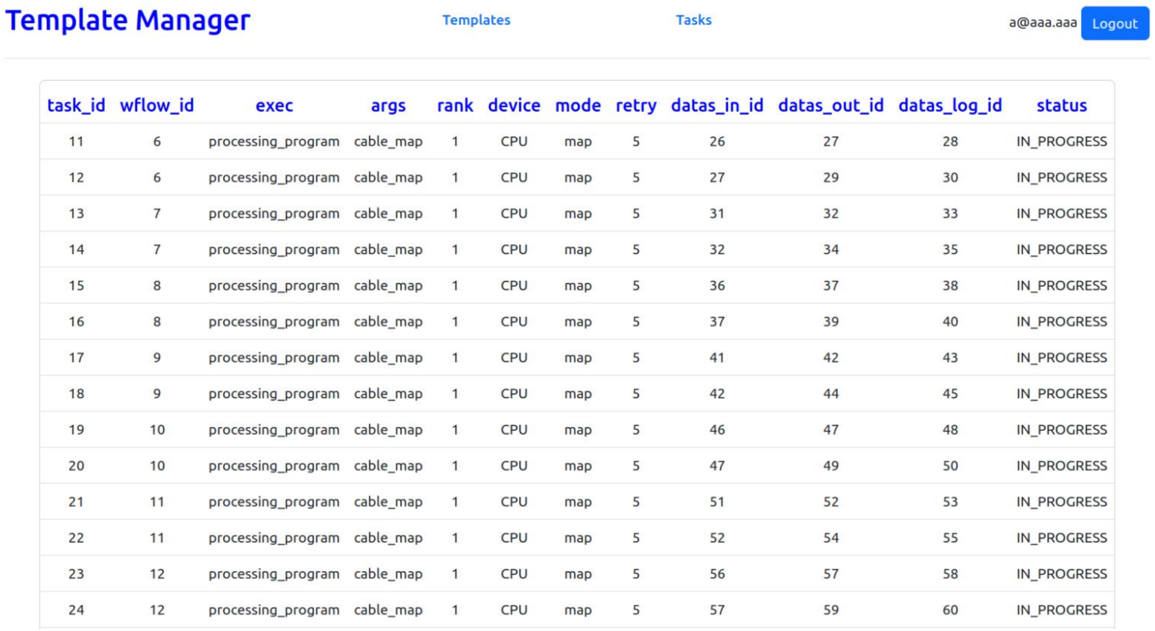
Examples of Templates and Tasks

- Viewing templates and tasks is available to all users who have completed the authorization process;
- Template creation is only available to superusers;



template_id	name	inner_dataset_mask	description	status
1	template1	.test.	{"steps": {"decoding": {"run": {"class": "CommandLineTool", "baseCommand": "echo", "inputs": {"dataset_name": {"type": "string"}, "processing_program": {"type": "string"}, "processing_program_version": {"type": "string"}, "cable_map": {"type": "File"}, "input_params": {"type": "File"}}, "outputs": {"output_dataset": {"type": "File"}, "log_dataset": {"type": "File"}}}, "in": {"dataset_name": ".test.", "processing_program": "processing_program", "processing_program_version": "processing_program_version", "cable_map": "cable_map", "input_params": "input_params"}, "out": "[output_dataset, log_dataset]"}, "reconstruction": {"run": {"class": "CommandLineTool", "baseCommand": "echo", "inputs": {"dataset_name": {"type": "string"}, "processing_program": {"type": "string"}, "processing_program_version": {"type": "string"}, "cable_map": {"type": "File"}, "input_params": {"type": "File"}}, "outputs": {"output_dataset": {"type": "File"}, "log_dataset": {"type": "File"}}}, "in": {"dataset_name": ".test.", "processing_program": "processing_program", "processing_program_version": "processing_program_version", "cable_map": "cable_map", "input_params": "input_params"}, "out": "[output_dataset, log_dataset]"}}}	ACTUAL
2	template2	.test.	{"steps": {"decoding": {"run": {"class": "CommandLineTool", "baseCommand": "echo", "inputs": {"dataset_name": {"type": "string"}, "processing_program": {"type": "string"}, "processing_program_version": {"type": "string"}, "cable_map": {"type": "File"}, "input_params": {"type": "File"}}, "outputs": {"output_dataset": {"type": "File"}, "log_dataset": {"type": "File"}}}, "in": {"dataset_name": ".test.", "processing_program": "processing_program", "processing_program_version": "processing_program_version", "cable_map": "cable_map", "input_params": "input_params"}, "out": "[output_dataset, log_dataset]"}}}	ARCHIVED

Created template



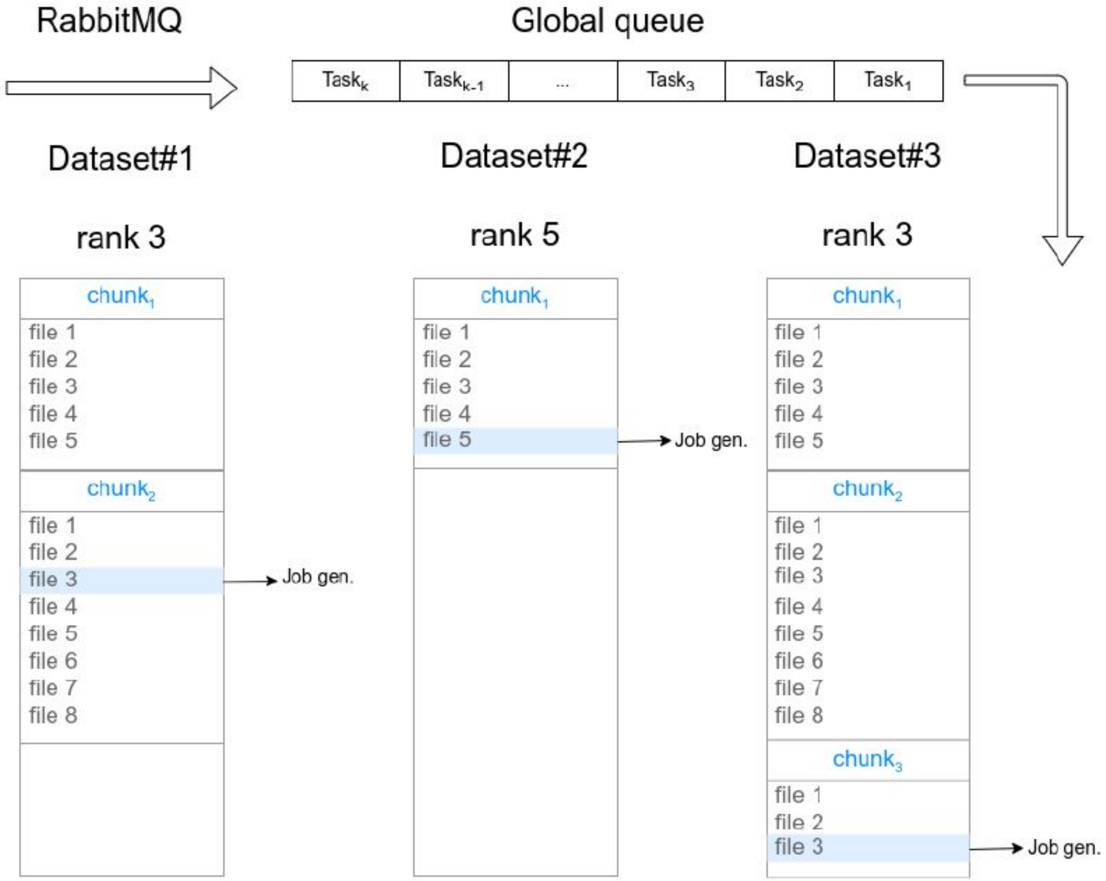
task_id	wflow_id	exec	args	rank	device	mode	retry	datas_in_id	datas_out_id	datas_log_id	status
11	6	processing_program	cable_map	1	CPU	map	5	26	27	28	IN_PROGRESS
12	6	processing_program	cable_map	1	CPU	map	5	27	29	30	IN_PROGRESS
13	7	processing_program	cable_map	1	CPU	map	5	31	32	33	IN_PROGRESS
14	7	processing_program	cable_map	1	CPU	map	5	32	34	35	IN_PROGRESS
15	8	processing_program	cable_map	1	CPU	map	5	36	37	38	IN_PROGRESS
16	8	processing_program	cable_map	1	CPU	map	5	37	39	40	IN_PROGRESS
17	9	processing_program	cable_map	1	CPU	map	5	41	42	43	IN_PROGRESS
18	9	processing_program	cable_map	1	CPU	map	5	42	44	45	IN_PROGRESS
19	10	processing_program	cable_map	1	CPU	map	5	46	47	48	IN_PROGRESS
20	10	processing_program	cable_map	1	CPU	map	5	47	49	50	IN_PROGRESS
21	11	processing_program	cable_map	1	CPU	map	5	51	52	53	IN_PROGRESS
22	11	processing_program	cable_map	1	CPU	map	5	52	54	55	IN_PROGRESS
23	12	processing_program	cable_map	1	CPU	map	5	56	57	58	IN_PROGRESS
24	12	processing_program	cable_map	1	CPU	map	5	57	59	60	IN_PROGRESS

WfMS task description

Workload management system requirements - reminder

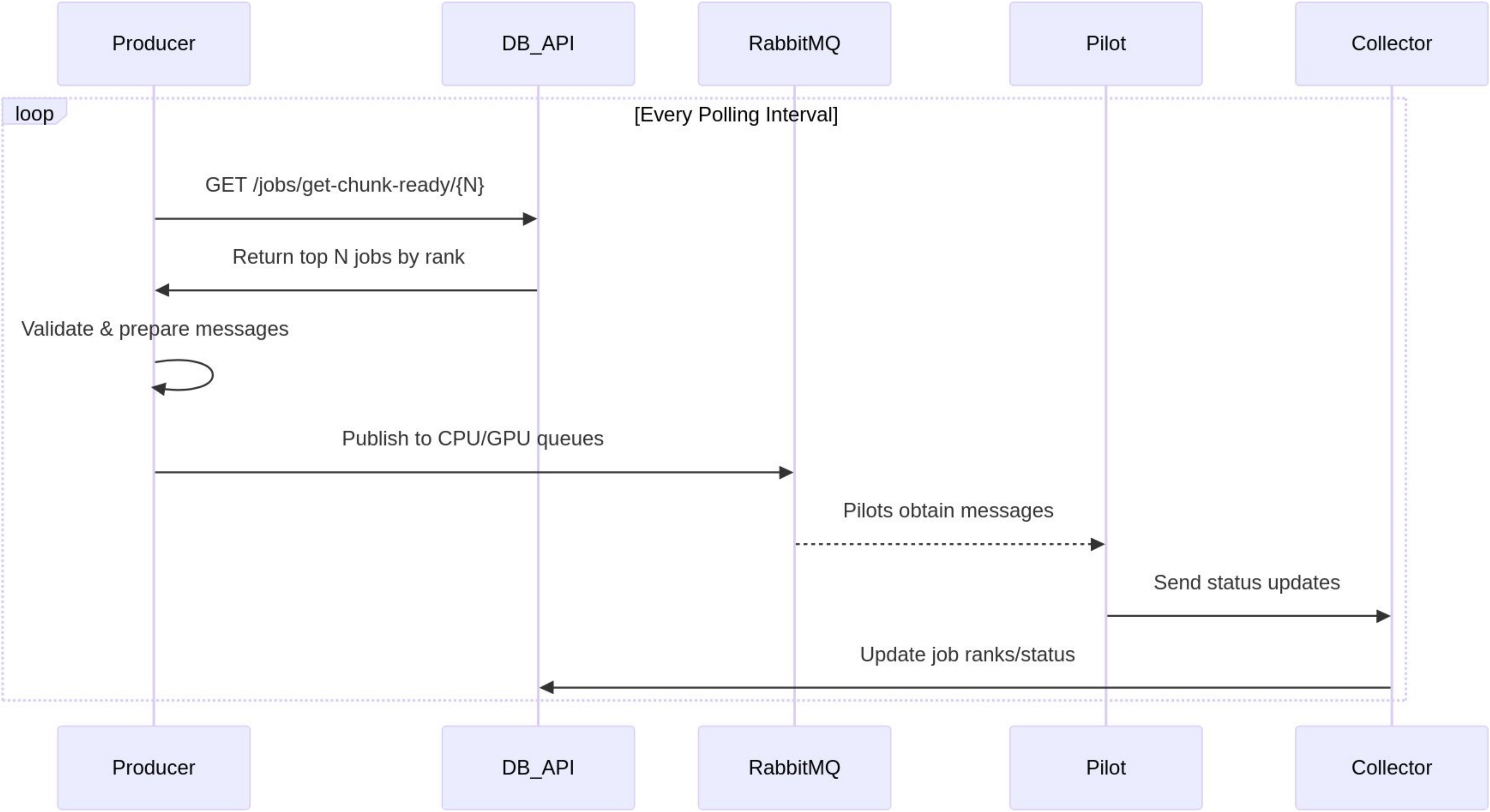
The key requirement - systems must meet the **high-throughput paradigm**.

- ❑ **Task registration**: formalized task description, including job options and required metadata registration;
- ❑ **Jobs definition**: generation of required number of jobs to perform task by controlled loading of available computing resources;
- ❑ **Jobs execution management**: continuous job state monitoring by communication with pilot, job retries in case of failures, job execution termination;
- ❑ **Consistency control**: control of the consistency of information in relation to the tasks, files and jobs;
- ❑ **Scheduling:** implementing a scheduling principle for task/job distribution;



Forming jobs based on dataset contents, one file per one job

Workload Management System - Pilot Agent



Task-executor (Scheduler)

Continuous-Time Domain

The original aging term is the following:

$$\alpha \ln(x(t)+1)$$

With lead-lag compensation, should be

$$\alpha_{\text{adj}}(t) = \mathcal{L}^{-1}\left[\frac{1+\tau_1 s}{1+\tau_2 s} \cdot \mathcal{L}(\alpha \ln(x(t)+1))\right]$$

Heaviside step function H(t–t_k) introduces instantaneous jumps at retry times t_k

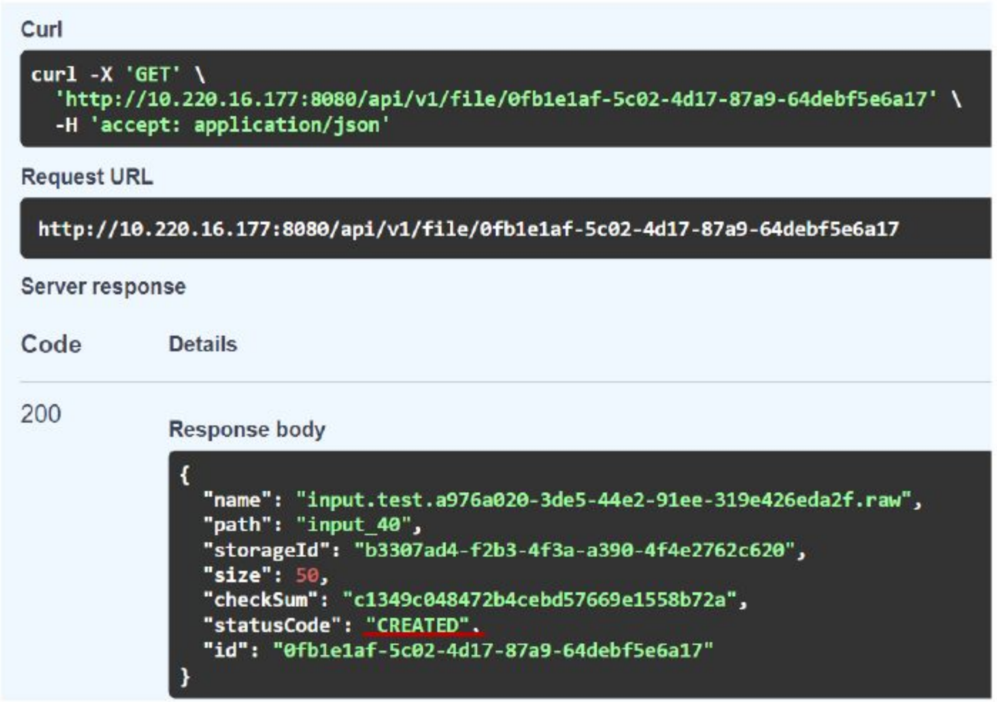
$$y(t) = \sum_{k=1}^{N_{\text{retry}}} H(t-t_k)$$

And retry penalty depends on past events (retry history), making the system state depend on its history, so we have a **delay differential equation**, which models the “physics” of retry-driven rank adjustments of our jobs

$$\frac{dr}{dt} = (\gamma-1)r(t) + \alpha_{\text{adj}}(t) - \beta 2^{y(t)} + \delta(1-L)$$

	Discrete Simulation	Continuous Solution
Retries	Exact event times	Requires Dirac delta
Implementation	Matches real code	Theoretical analysis
Stability	Bounded by design	Must prove convergence?
Visualization	Step changes	Smooth curves (Runge-Kutta Solver?)

Data consistency



Curl

```
curl -X 'GET' \
  'http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64debf5e6a17' \
  -H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64debf5e6a17
```

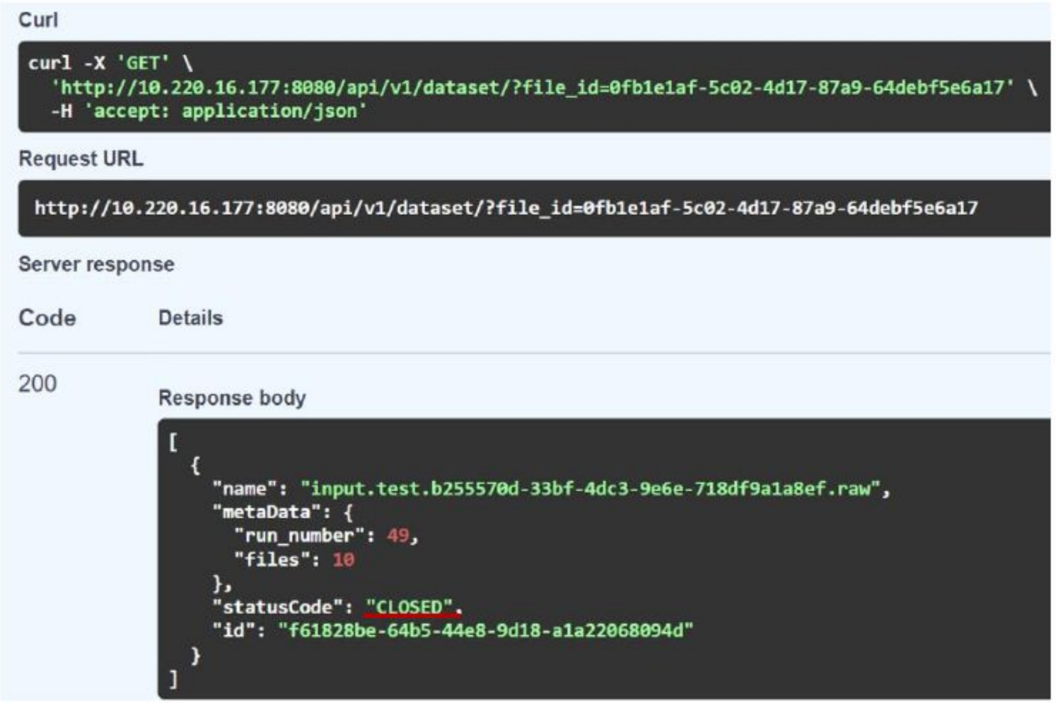
Server response

Code Details

200

Response body

```
{
  "name": "input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw",
  "path": "input_40",
  "storageId": "b3307ad4-f2b3-4f3a-a390-4f4e2762c620",
  "size": 50,
  "checkSum": "c1349c048472b4cebd57669e1558b72a",
  "statusCode": "CREATED",
  "id": "0fb1e1af-5c02-4d17-87a9-64debf5e6a17"
}
```



Curl

```
curl -X 'GET' \
  'http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64debf5e6a17' \
  -H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64debf5e6a17
```

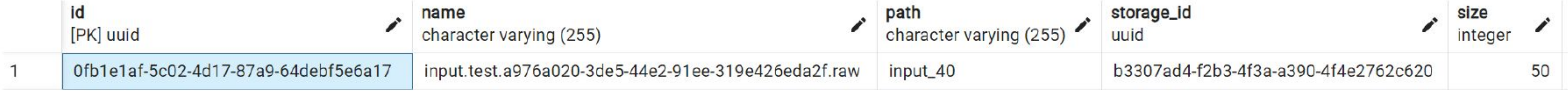
Server response

Code Details

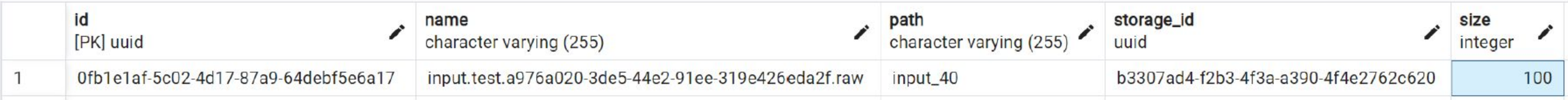
200

Response body

```
[
  {
    "name": "input.test.b255570d-33bf-4dc3-9e6e-718df9a1a8ef.raw",
    "metaData": {
      "run_number": 49,
      "files": 10
    },
    "statusCode": "CLOSED",
    "id": "f61828be-64b5-44e8-9d18-a1a22068094d"
  }
]
```



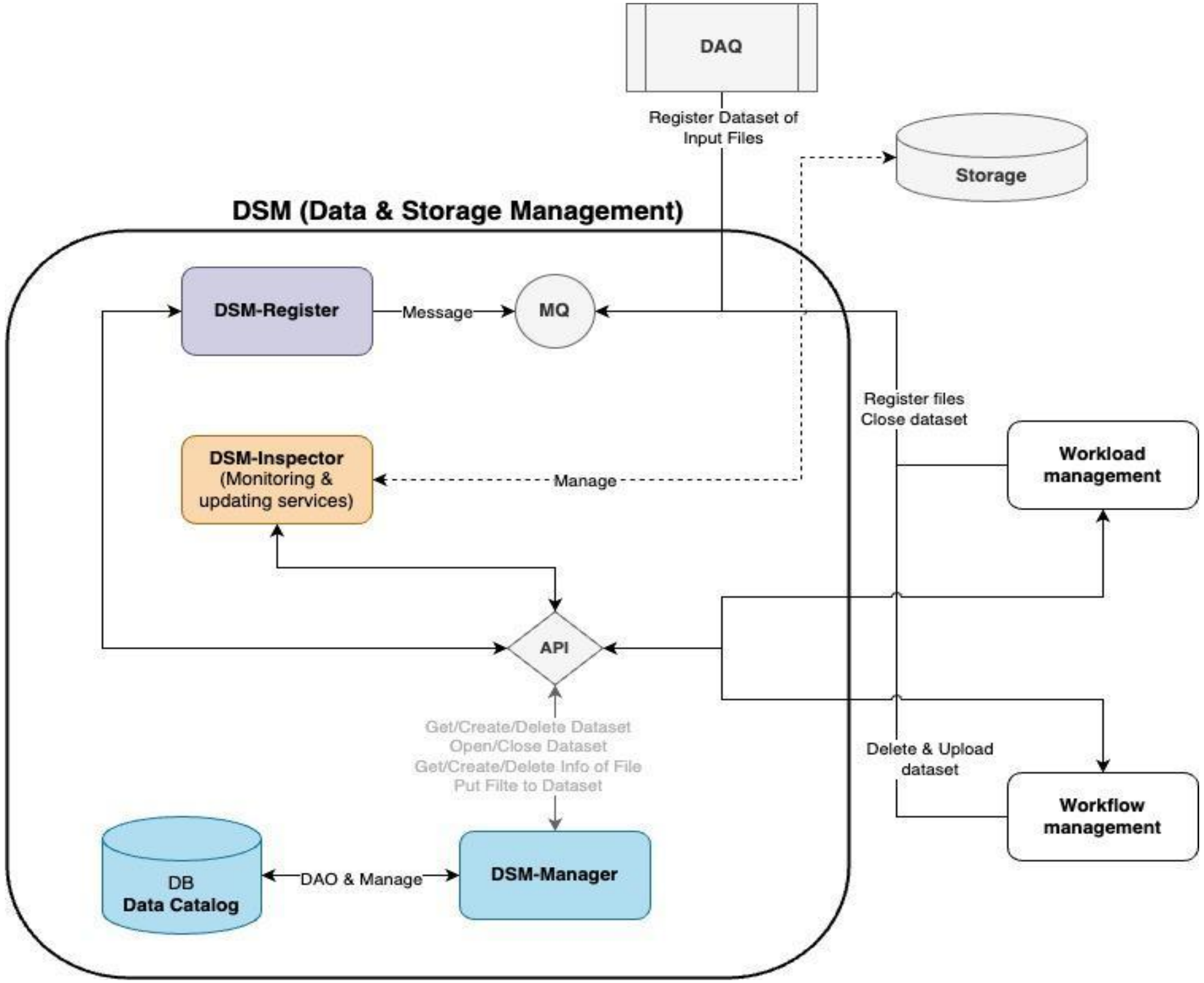
	id [PK] uuid	name character varying (255)	path character varying (255)	storage_id uuid	size integer
1	0fb1e1af-5c02-4d17-87a9-64debf5e6a17	input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw	input_40	b3307ad4-f2b3-4f3a-a390-4f4e2762c620	50



	id [PK] uuid	name character varying (255)	path character varying (255)	storage_id uuid	size integer
1	0fb1e1af-5c02-4d17-87a9-64debf5e6a17	input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw	input_40	b3307ad4-f2b3-4f3a-a390-4f4e2762c620	100

Data & Storage Management Update

1. **DSM-Register (Data Registration)**:
    a. Create a new consumer for the queue dsm.register.dataset.delete
    b. Write a correspondent message handler
2. **DSM-Manager (REST API of data catalog):**
    a. Getting the list of files/datasets by status
    b. Searching for a file by name
3. **DSM-Inspector (Daemon tasks)**:
    a. Storage monitoring service for dark files
    b. Checking file integrity
    c. Deleting files and datasets



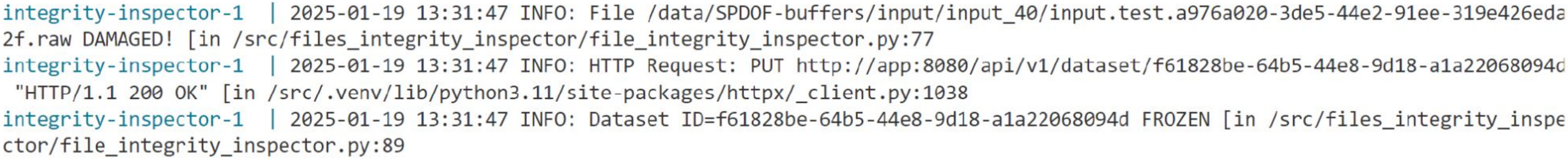
Architecture of Data Management System

Data & Storage Management

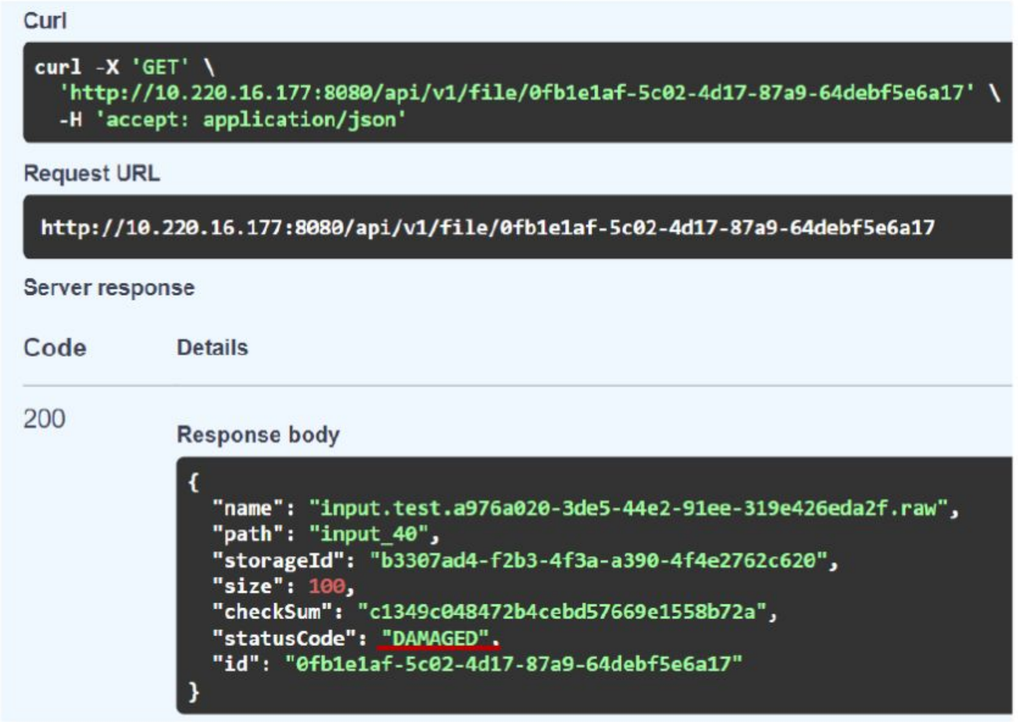
Next steps:

1. **dsm-inspector:**
    a. Implement background services for
        i. Control file uploads
        ii. Control storage utilization
2. **dsm-register**
    a. Implement message processing from the following queues:
        i. dsm.register.dataset.closed - Accepting request to close a dataset
        ii. dsm.register.dataset.upload - To upload files in a dataset to an external storage

Data consistency



```
integrity-inspector-1  | 2025-01-19 13:31:47 INFO: File /data/SPDOF-buffers/input/input_40/input.test.a976a020-3de5-44e2-91ee-319e426eda
2f.raw DAMAGED! [in /src/files_integrity_inspector/file_integrity_inspector.py:77
integrity-inspector-1  | 2025-01-19 13:31:47 INFO: HTTP Request: PUT http://app:8080/api/v1/dataset/f61828be-64b5-44e8-9d18-a1a22068094d
 "HTTP/1.1 200 OK" [in /src/.venv/lib/python3.11/site-packages/httpx/_client.py:1038
integrity-inspector-1  | 2025-01-19 13:31:47 INFO: Dataset ID=f61828be-64b5-44e8-9d18-a1a22068094d FROZEN [in /src/files_integrity_inspe
ctor/file_integrity_inspector.py:89
```



Curl

```
curl -X 'GET' \
  'http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64debf5e6a17' \
  -H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64debf5e6a17
```

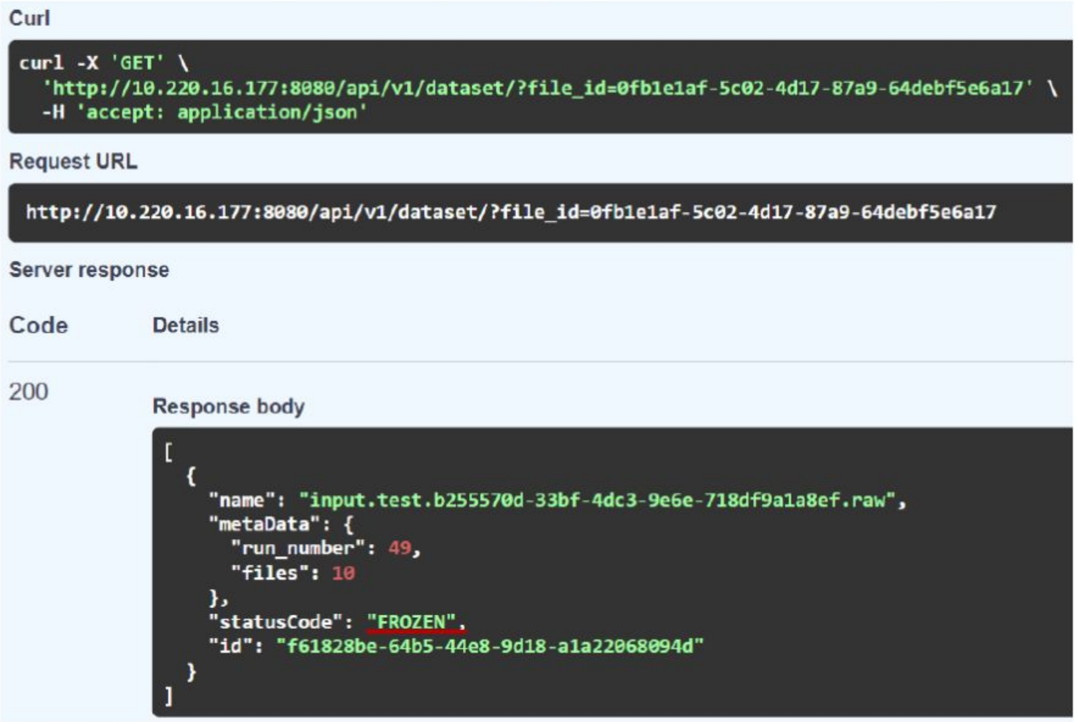
Server response

Code Details

200

Response body

```
{
  "name": "input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw",
  "path": "input_40",
  "storageId": "b3307ad4-f2b3-4f3a-a390-4f4e2762c620",
  "size": 100,
  "checkSum": "c1349c048472b4cebd57669e1558b72a",
  "statusCode": "DAMAGED",
  "id": "0fb1e1af-5c02-4d17-87a9-64debf5e6a17"
}
```



Curl

```
curl -X 'GET' \
  'http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64debf5e6a17' \
  -H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64debf5e6a17
```

Server response

Code Details

200

Response body

```
[
  {
    "name": "input.test.b255570d-33bf-4dc3-9e6e-718df9a1a8ef.raw",
    "metaData": {
      "run_number": 49,
      "files": 10
    },
    "statusCode": "FROZEN",
    "id": "f61828be-64b5-44e8-9d18-a1a22068094d"
  }
]
```