INTERNATIONAL
INTERGOVERNMENTAL
ORGANIZATION

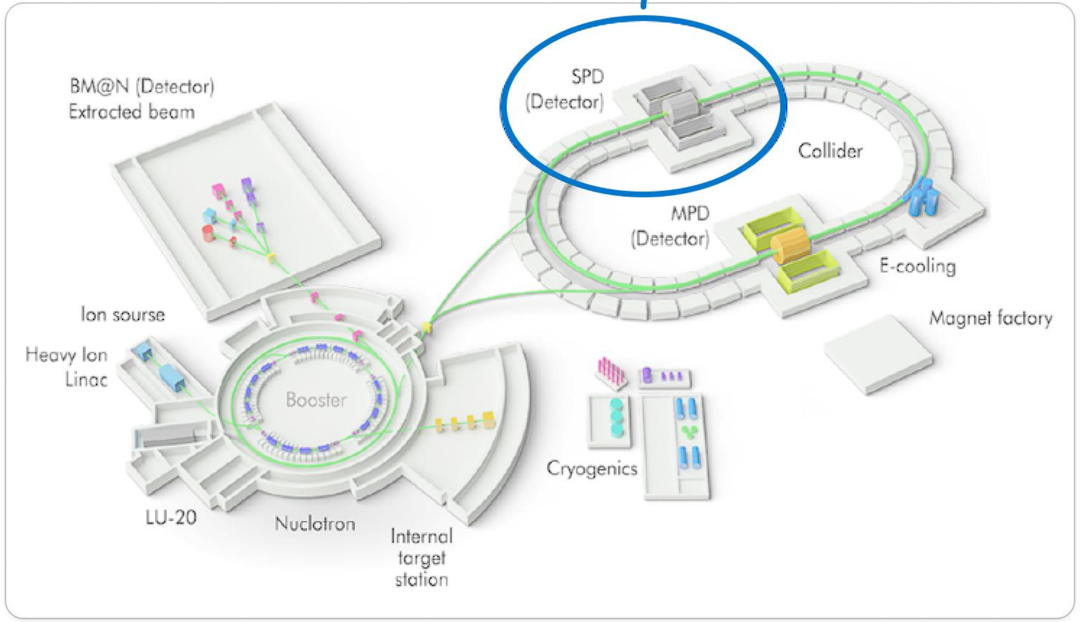
JOINT INSTITUTE
FOR NUCLEAR RESEARCH

SPD NICA

Workflow Management System for SPD Online Filter

Presenter: Artem Plotnikov

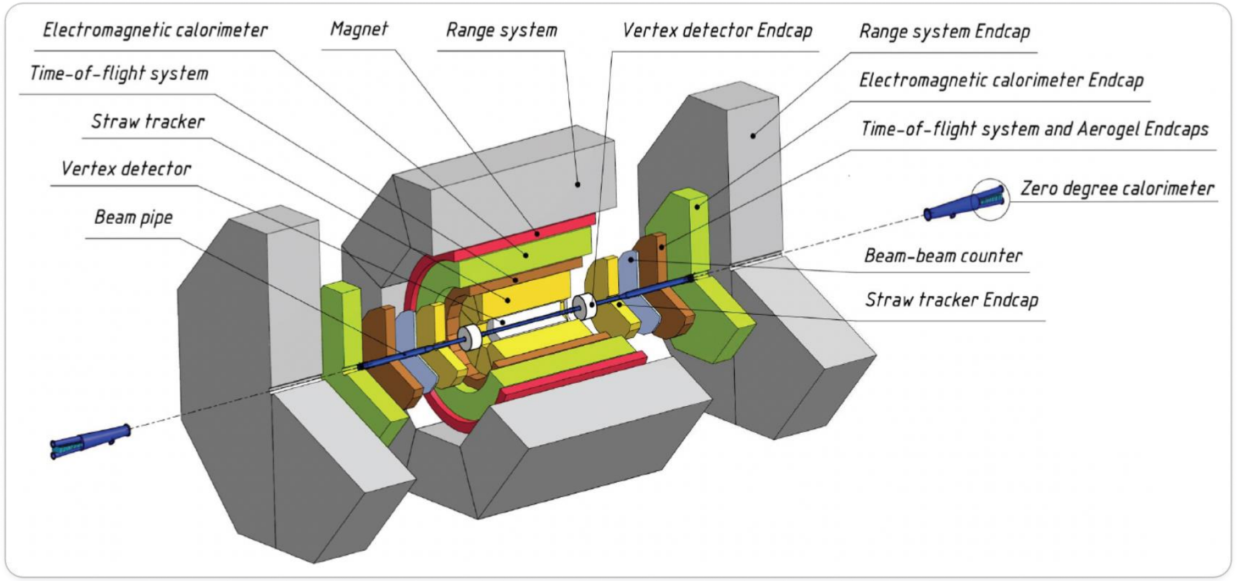
Spin Physics Detector



NICA (Nuclotron-based Ion Collider fAcility)
Dubna, Russia

- The expected **event rate** of the SPD experiment is about 3 MHz.
- This is equivalent to a **raw data rate** of 20 GB/s (or ~ 200 PB/year).

Objective: to study the spin structure of the proton (p) and deuteron (d) and other spin-related phenomena in polarized p-p, d-d and p-d collisions.



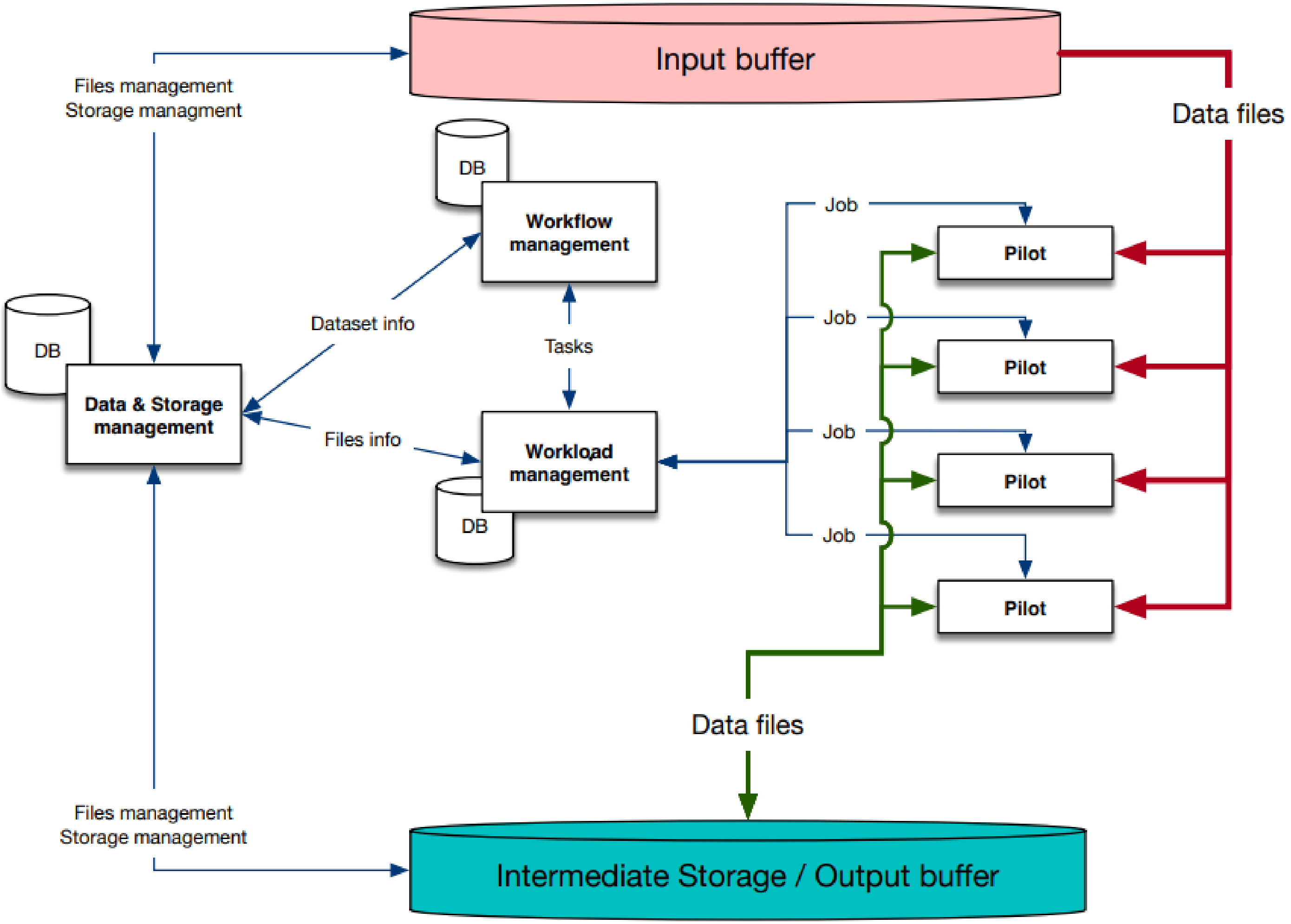
Detector diagram

SPD Online Filter

SPD Online Filter is a specialized hardware and software system designed for preliminary processing of SPD experiment data.

The system implements a **multistage**, high-throughput processing method.

The main, but not the only, goal of processing is to significantly **reduce the volume of data** for subsequent analysis and long-term storage.

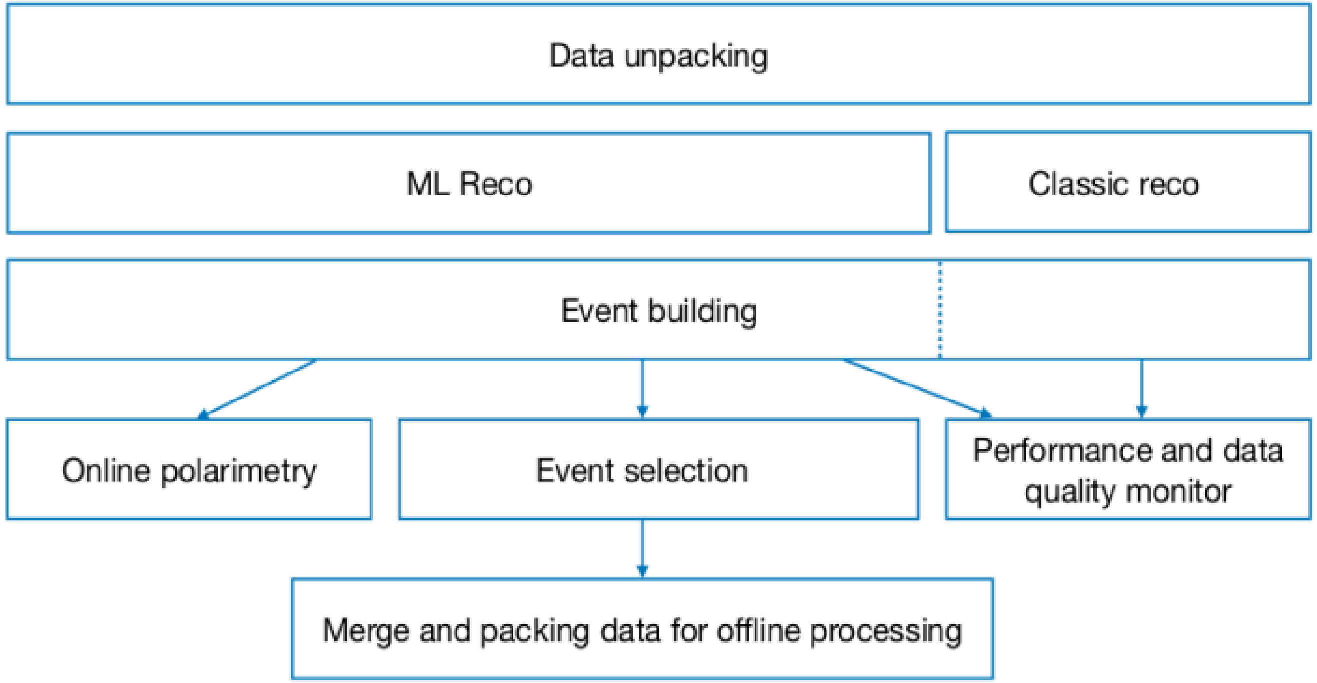


Middleware complex "SPD Online filter"

Workflow management

Multistage processing is a set of sequential stages of data processing. At each stage, a fairly large amount of data can be processed.

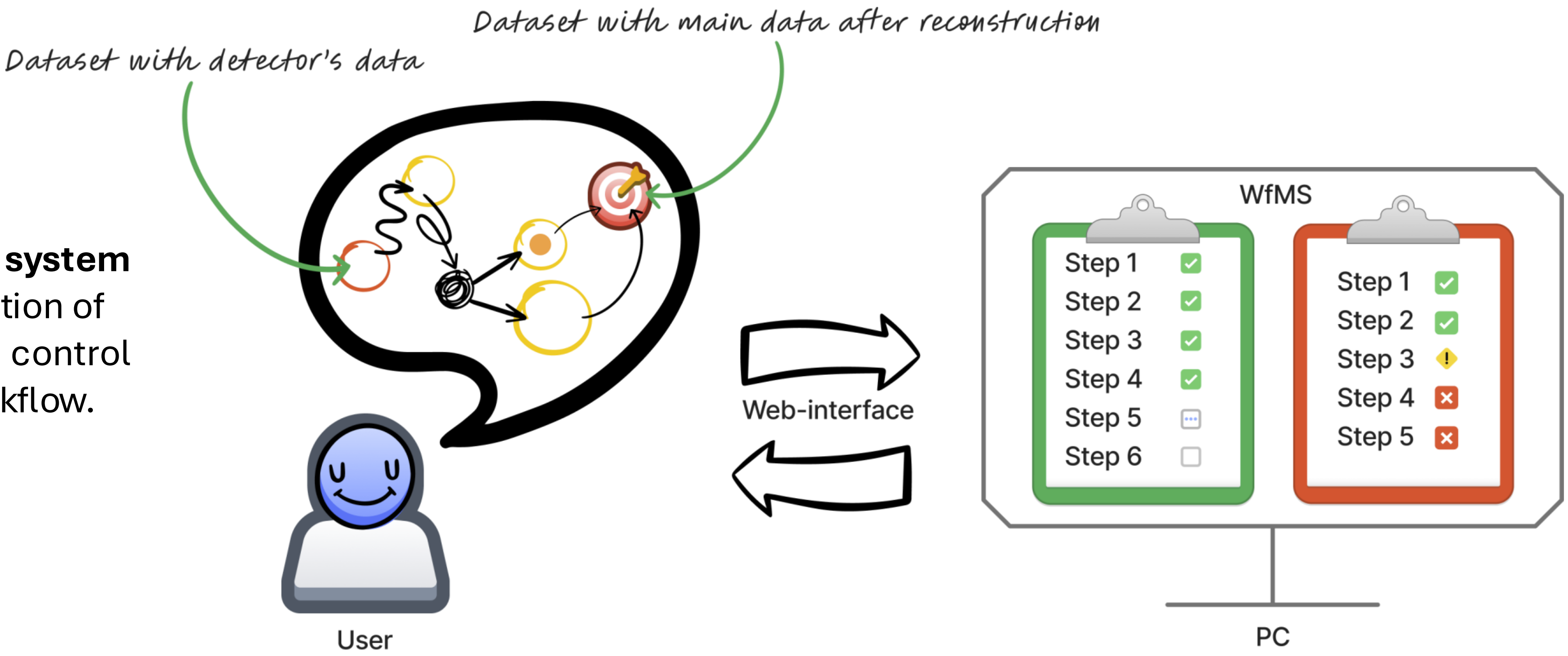
Each step, except the first, accepts the data processed at the previous stage and passes the results to the next, ensuring the **transformation of information from one representation to another**.


Data unpacking
ML Reco
Classic reco
Event building
Online polarimetry
Event selection
Performance and data quality monitor
Merge and packing data for offline processing

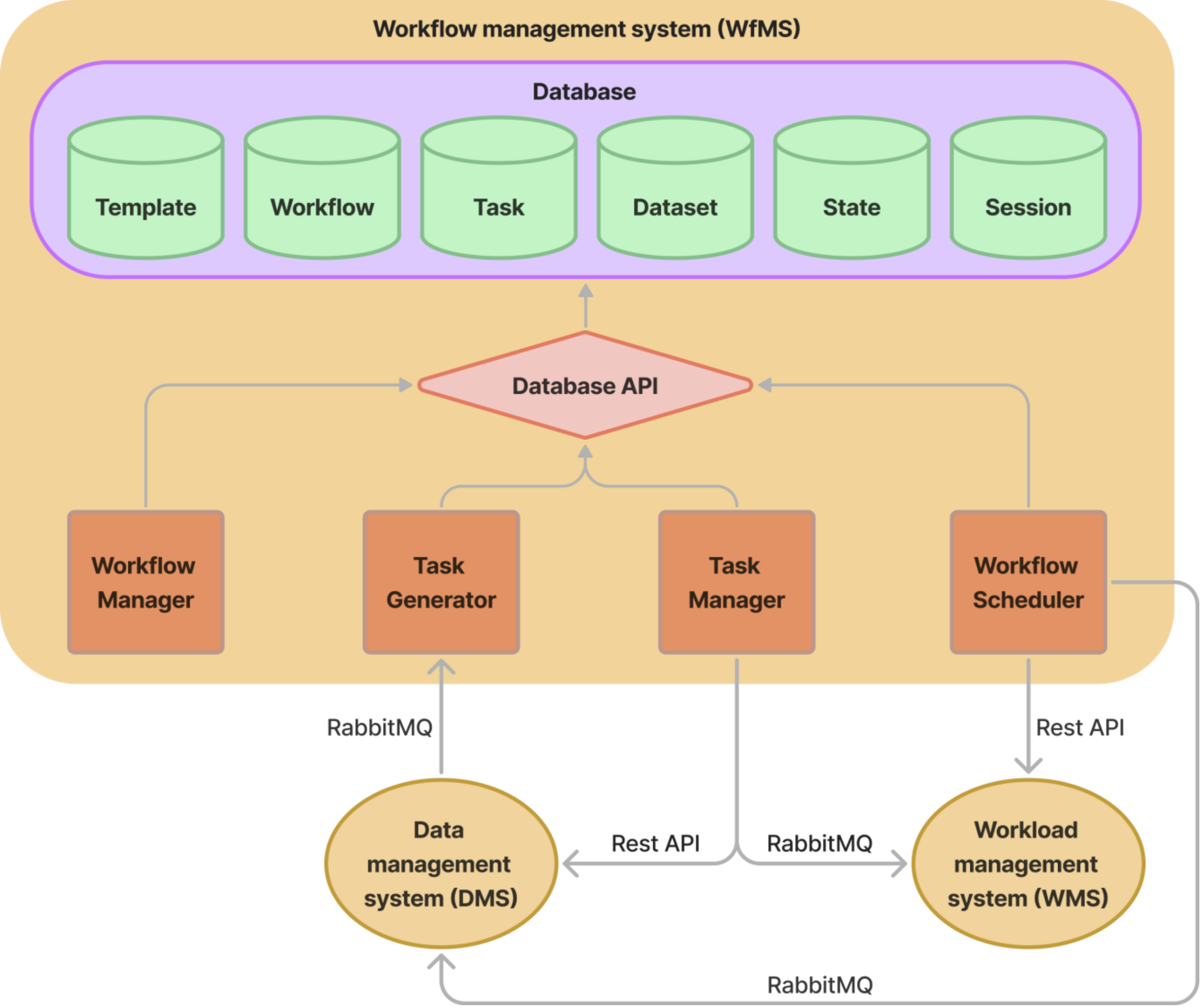
Example of sequential processing steps

Workflow management system (WfMS)

Workflow management system responsible for the definition of processing pipelines and control of processing stages workflow.



WfMS's microservices



WfMS architecture

Workflow Manager – service for interacting with the operator, which defines processing pipeline templates.

Task Generator – service for generating a chain of tasks based on specified templates and data.

Task Manager – data management system (DMS) polling service for retrieving the datasets and sending them for processing to workload management system (WMS).

Workflow Scheduler – workload management system (WMS) polling service that monitors the status of task execution and controls the life cycle of intermediate datasets by sending requests for their deletion.

Tech stack

python

FastAPI

RabbitMQ

COMMON
WORKFLOW
LANGUAGE

SQLA

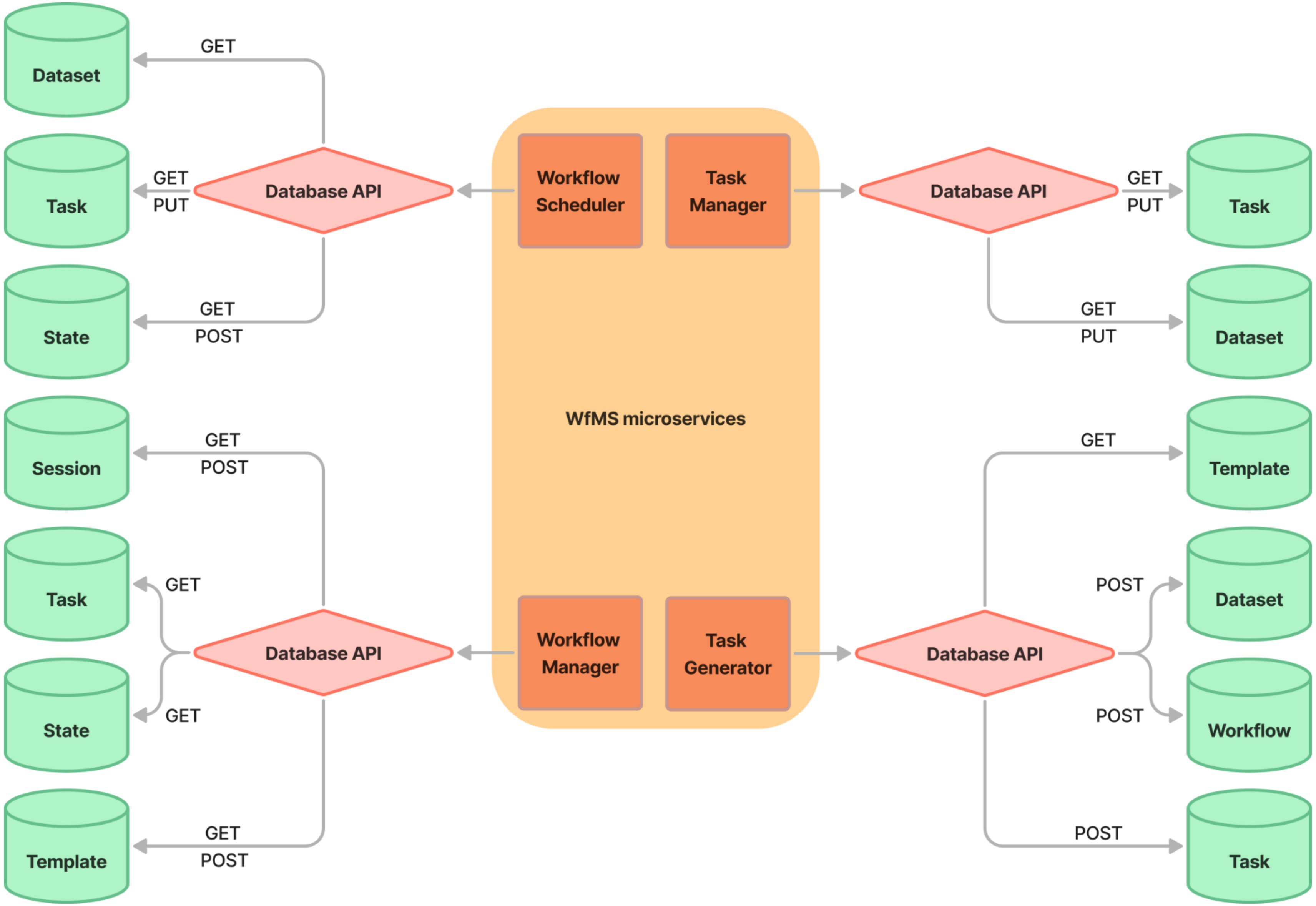
PostgreSQL

Bootstrap

神社 Jinja

docker

Interaction of microservices and database



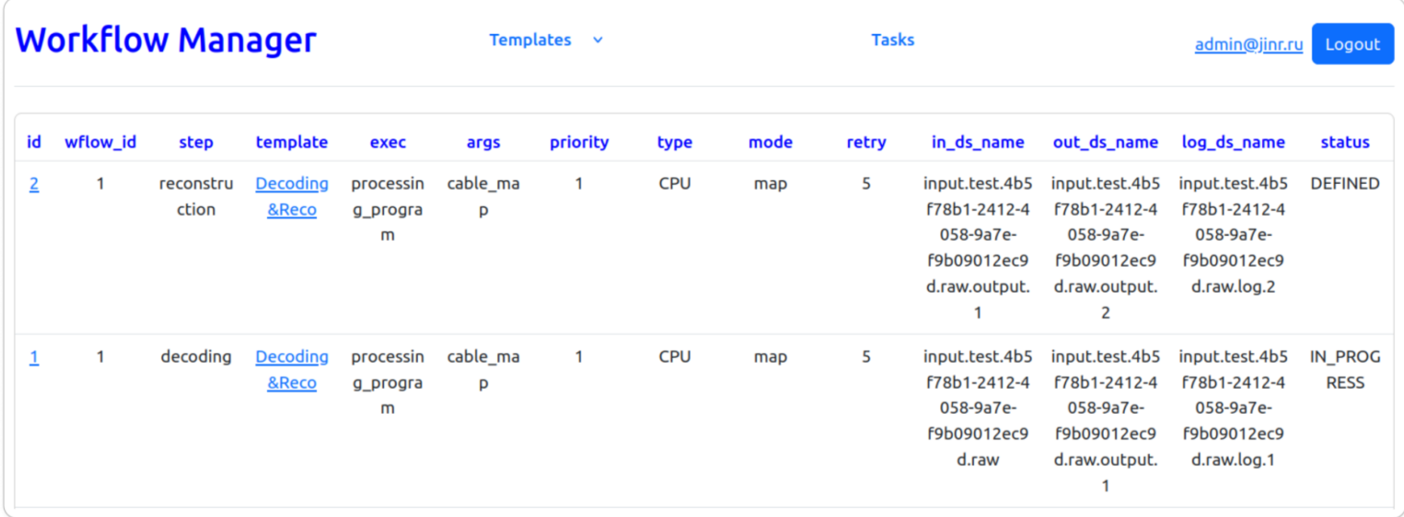
- **PostgreSQL** is deployed on a separate virtual machine.
- Interaction with the database is carried out via **REST API**.

SQLAlchemy ORM
+
Asynchronous sessions and asyncpg
+
Alembic migrations

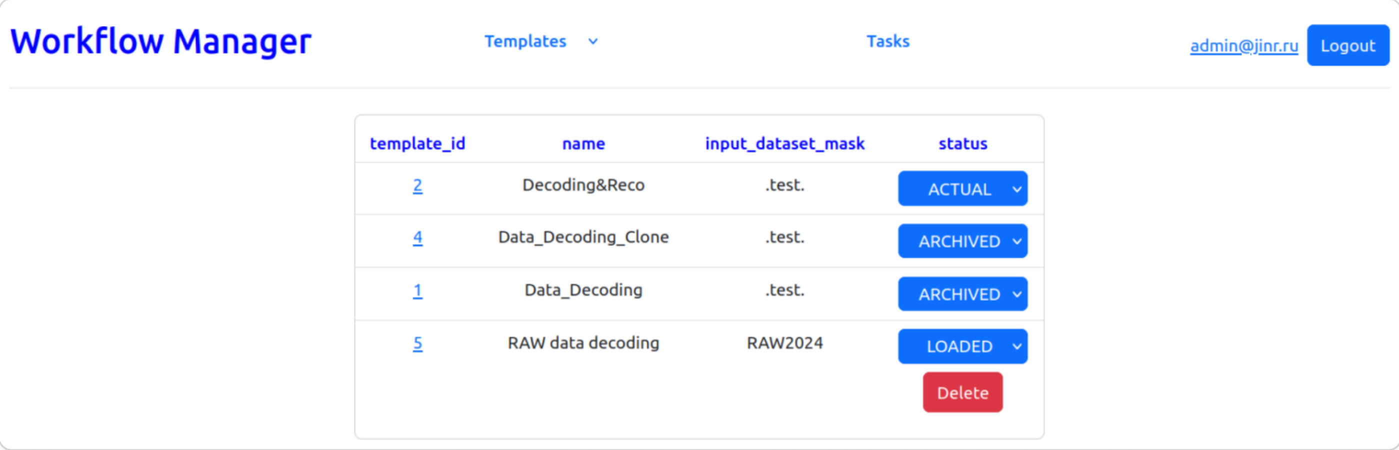
Workflow Manager

1. User authorization using SPD-IAM;
2. Output of CWL templates;
3. Output of tasks;
4. Creation of CWL templates by superusers (entering into a special field / loading from a file / cloning existing templates);
5. Preliminary validation and recording in the CWL templates DB;
6. Changing the template statuses by superusers to “ACTUAL” and “ARCHIVED”;
7. Deleting templates with the “LOADED” status by superusers.

- Backend: **FastAPI**
- Template engine: **Jinja2**
- Frontend: **Bootstrap** (HTML + CSS + JS)



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

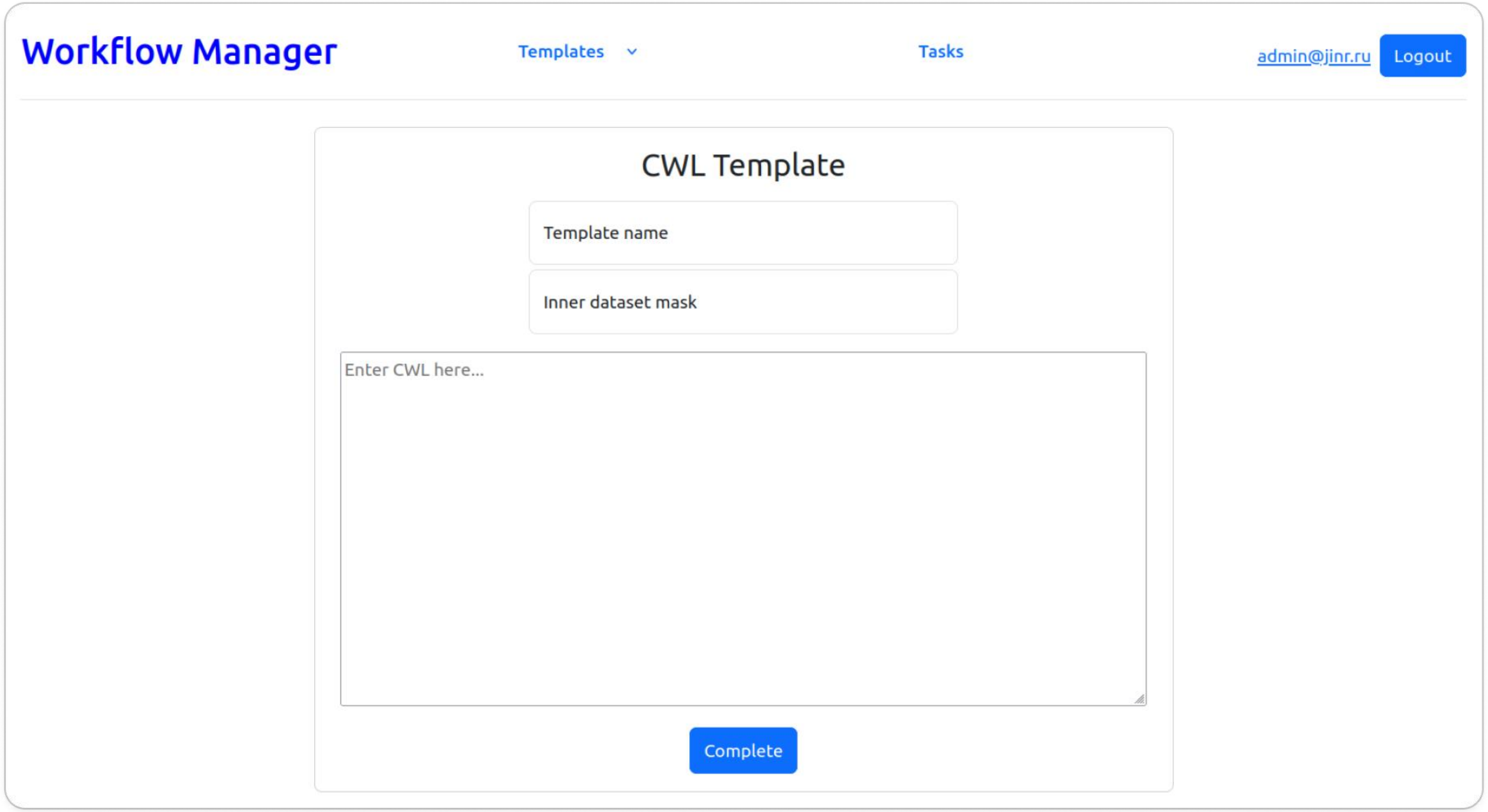


template_id	name	input_dataset_mask	status
2	Decoding&Reco	.test.	ACTUAL
4	Data_Decoding_Clone	.test.	ARCHIVED
1	Data_Decoding	.test.	ARCHIVED
5	RAW data decoding	RAW2024	LOADED Delete

User interface

Templates creation

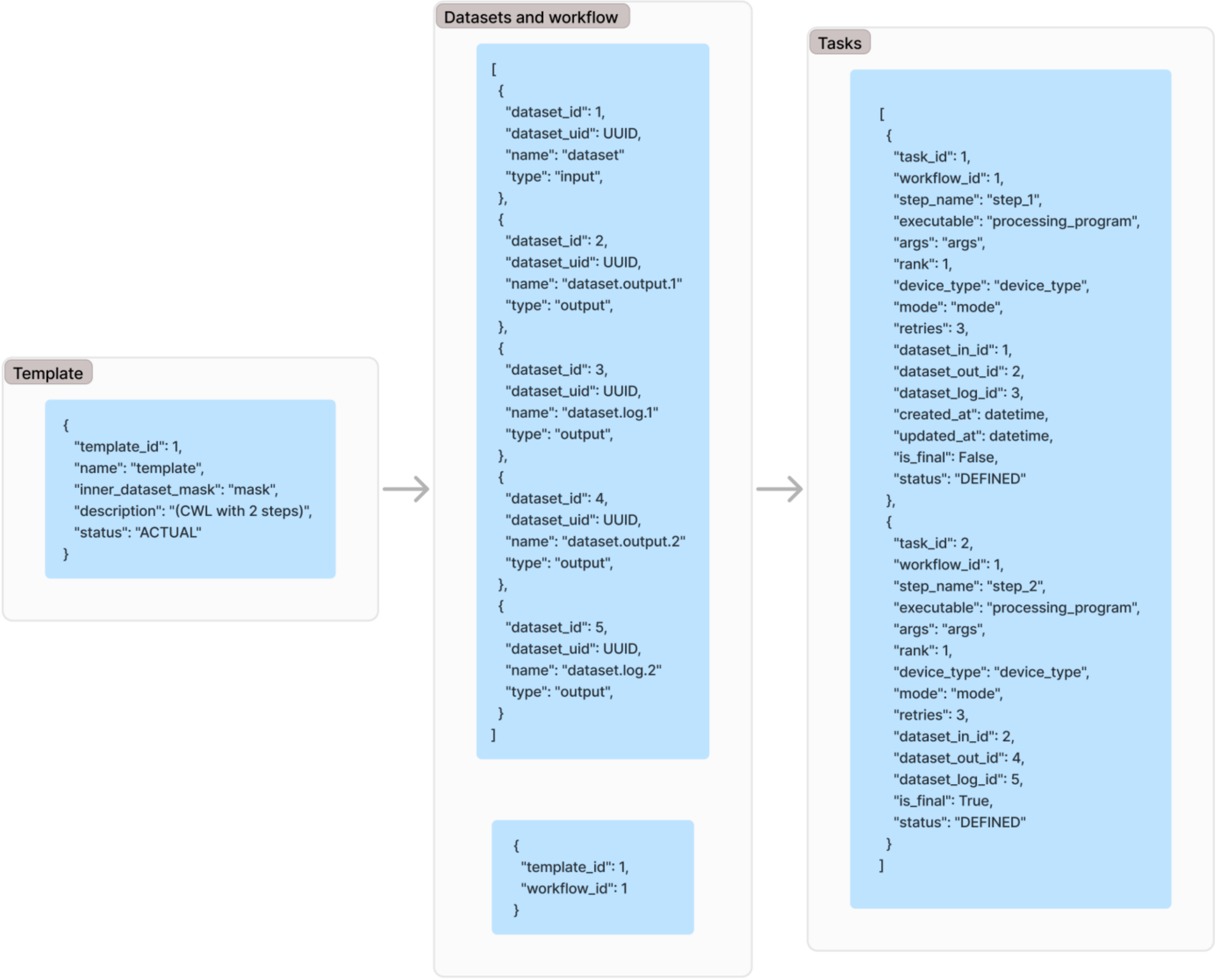
- **Common Workflow Language (CWL)**
- Template creation is **only available to superusers** (in particular, there is no button in the interface for a regular user).
- **Pre-validation** of templates using cwltools.
- **Saving** a template to the database.



Template creation page

Task Generator

1. Receiving registered datasets from DMS from RabbitMQ;
2. Matching a dataset by name mask with the required template;
3. Registering an input dataset in the system;
4. Creating a processing process by template;
5. Creating an output dataset and a log dataset in the system;
6. Creating tasks.



```
{
  "template_id": 1,
  "name": "template",
  "inner_dataset_mask": "mask",
  "description": "(CWL with 2 steps)",
  "status": "ACTUAL"
}
```

```
[
 {
  "dataset_id": 1,
  "dataset_uid": UUID,
  "name": "dataset"
  "type": "input",
 },
 {
  "dataset_id": 2,
  "dataset_uid": UUID,
  "name": "dataset.output.1"
  "type": "output",
 },
 {
  "dataset_id": 3,
  "dataset_uid": UUID,
  "name": "dataset.log.1"
  "type": "output",
 },
 {
  "dataset_id": 4,
  "dataset_uid": UUID,
  "name": "dataset.output.2"
  "type": "output",
 },
 {
  "dataset_id": 5,
  "dataset_uid": UUID,
  "name": "dataset.log.2"
  "type": "output",
 }
]
```

```
{
 "template_id": 1,
 "workflow_id": 1
}
```

```
[
 {
  "task_id": 1,
  "workflow_id": 1,
  "step_name": "step_1",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 1,
  "dataset_out_id": 2,
  "dataset_log_id": 3,
  "created_at": datetime,
  "updated_at": datetime,
  "is_final": False,
  "status": "DEFINED"
 },
 {
  "task_id": 2,
  "workflow_id": 1,
  "step_name": "step_2",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 2,
  "dataset_out_id": 4,
  "dataset_log_id": 5,
  "is_final": True,
  "status": "DEFINED"
 }
]
```

From template to task

Task Manager

Tasks before

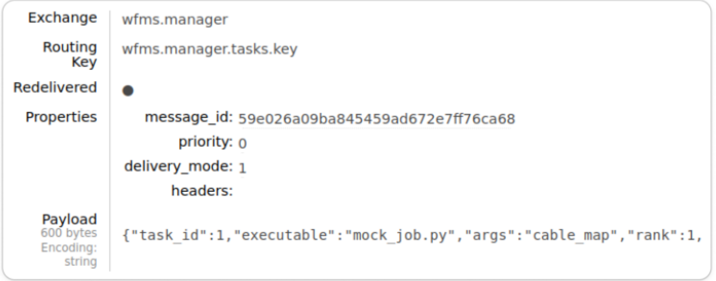
```
[
 {
  "task_id": 1,
  "workflow_id": 1,
  "step_name": "step_1",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 1,
  "dataset_out_id": 2,
  "dataset_log_id": 3,
  "created_at": datetime,
  "updated_at": datetime,
  "is_final": False,
  "status": "DEFINED"
 },
 {
  "task_id": 2,
  "workflow_id": 1,
  "step_name": "step_2",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 2,
  "dataset_out_id": 4,
  "dataset_log_id": 5,
  "is_final": True,
  "status": "DEFINED"
 }
]
```

Formed task

```
{
 "task_id": 1,
 "executable": "processing_program",
 "args": "cable_map",
 "rank": 1,
 "device_type": "device_type",
 "mode": "mode",
 "retries": 3,
 "dataset_in": [
  {
   "name": "dataset",
   "metaData": {
     "run_number": 1,
     "files": 50
   },
   "statusCode": "CLOSED",
   "id": UUID
  }
 ],
 "dataset_out": [
  {
   "name": "dataset.output.1",
   "metaData": {
     "task_id": 1
   },
   "statusCode": "OPEN",
   "id": UUID
  }
 ],
 "dataset_log": {
  "name": "dataset.log.1",
  "metaData": {
    "task_id":1
   },
  "statusCode": "OPEN",
  "id": UUID
 }
}
```

1. Iterating over tasks in the "DEFINED" status;
2. Querying DMS about the input dataset status ("CLOSED");
3. Creating output and log datasets in DMS;
4. Sending a task to RabbitMQ for subsequent processing in WMS;
5. Changing the task status to "RUNNING".

RabbitMQ



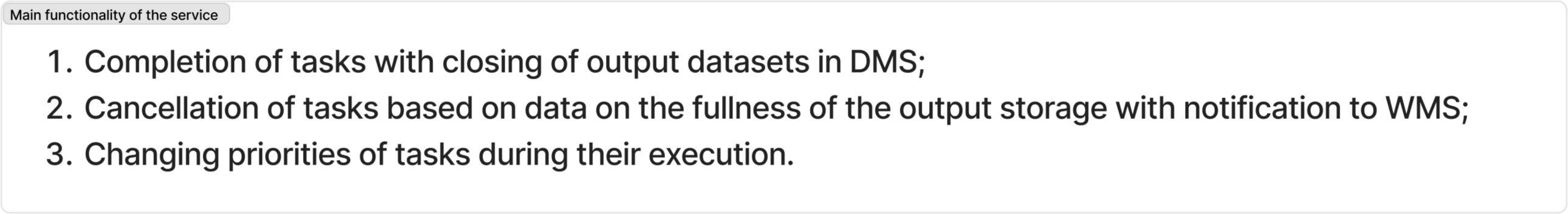
```
Exchange     wfms.manager
Routing Key  wfms.manager.tasks.key
Redelivered  ●
Properties       message_id: 59e026a09ba845459ad672e7ff76ca68
                   priority: 0
              delivery_mode: 1
                    headers:
Payload
600 bytes
Encoding:
string       {"task_id":1,"executable":"mock_job.py","args":"cable_map","rank":1,
```

Submitting a task for processing

Task after

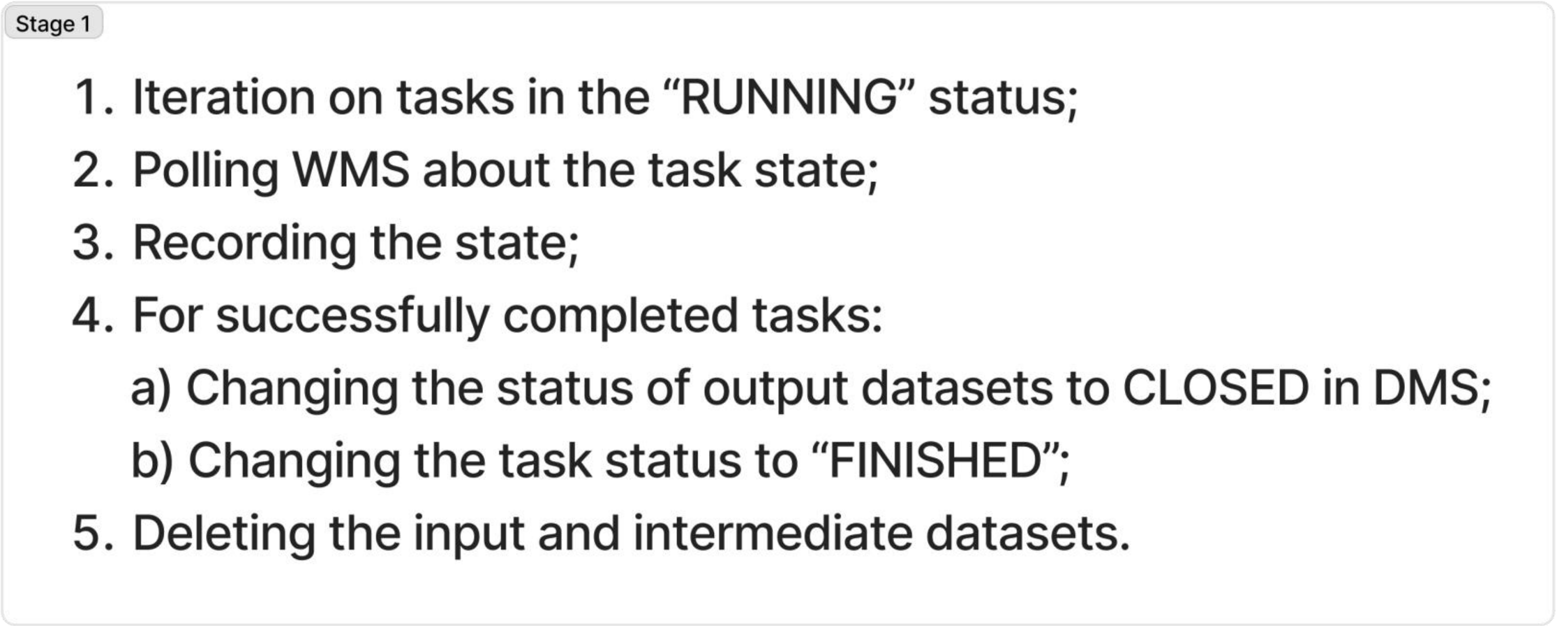
```
[
 {
  "task_id": 1,
  "workflow_id": 1,
  "step_name": "step_1",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 1,
  "dataset_out_id": 2,
  "dataset_log_id": 3,
  "created_at": datetime,
  "updated_at": datetime,
  "is_final": False,
  "status": "RUNNING"
 },
 {
  "task_id": 2,
  "workflow_id": 1,
  "step_name": "step_2",
  "executable": "processing_program",
  "args": "args",
  "rank": 1,
  "device_type": "device_type",
  "mode": "mode",
  "retries": 3,
  "dataset_in_id": 2,
  "dataset_out_id": 4,
  "dataset_log_id": 5,
  "is_final": True,
  "status": "DEFINED"
 }
]
```

Workflow Scheduler



Main functionality of the service

1. Completion of tasks with closing of output datasets in DMS;
2. Cancellation of tasks based on data on the fullness of the output storage with notification to WMS;
3. Changing priorities of tasks during their execution.

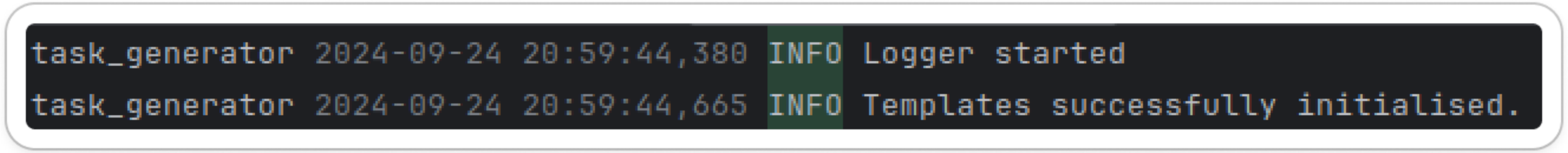


Stage 1

1. Iteration on tasks in the “RUNNING” status;
2. Polling WMS about the task state;
3. Recording the state;
4. For successfully completed tasks:
   a) Changing the status of output datasets to CLOSED in DMS;
   b) Changing the task status to “FINISHED”;
5. Deleting the input and intermediate datasets.

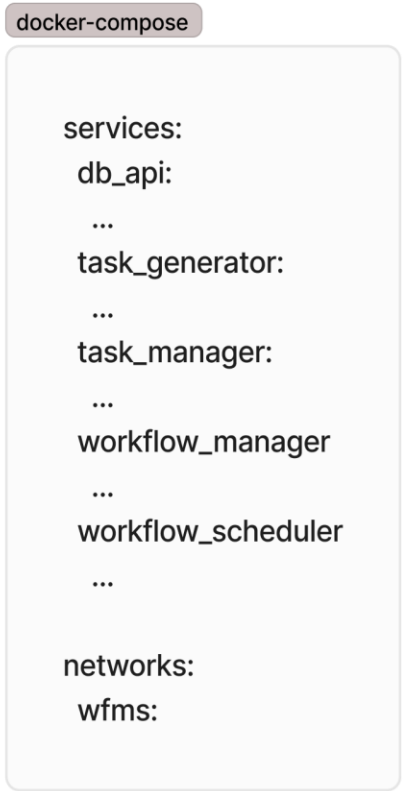
Logging, containerization and orchestration

- Logging: **custom logger**



```
task_generator 2024-09-24 20:59:44,380 INFO Logger started
task_generator 2024-09-24 20:59:44,665 INFO Templates successfully initialised.
```

- Containerization and orchestration via **Docker** and **Docker Compose**

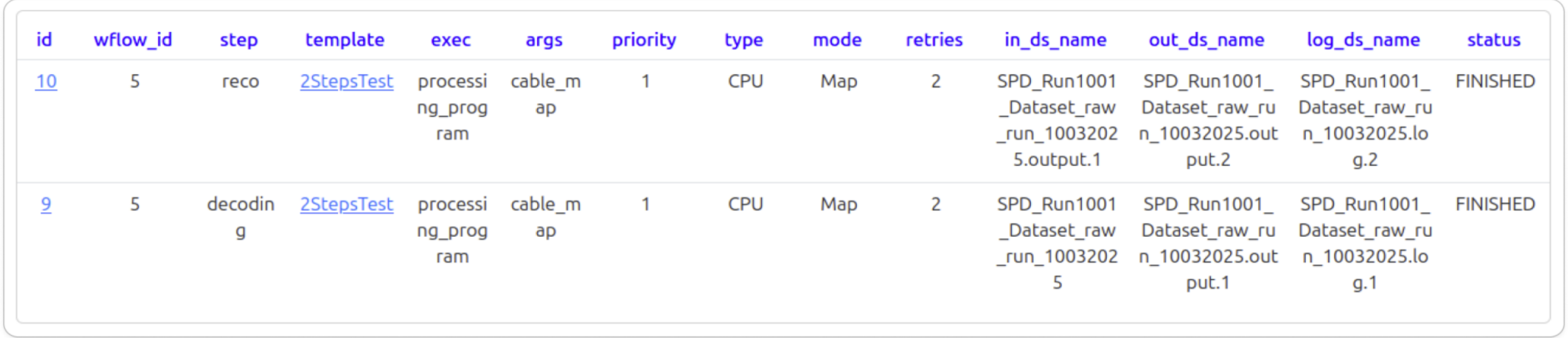


docker-compose

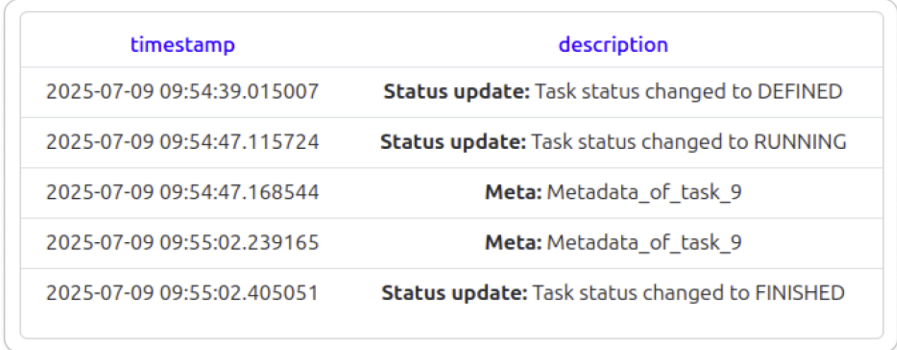
```
services:
  db_api:
    ...
  task_generator:
    ...
  task_manager:
    ...
  workflow_manager
    ...
  workflow_scheduler
    ...

networks:
  wfms:
```

Testing

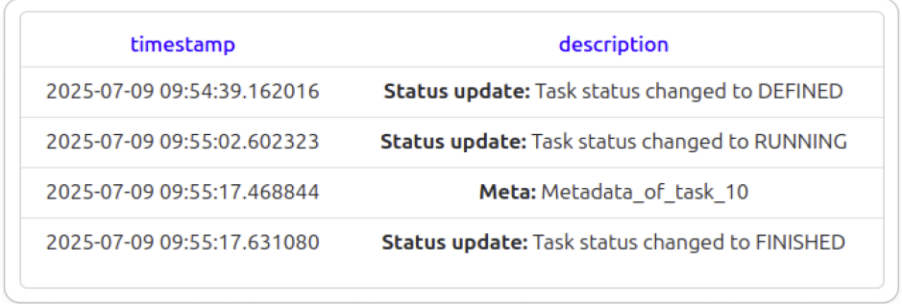
- **Users** are granted the correct permissions
- **Templates** are validated and saved
- **Tasks** are generated and submitted based on input
- **Workflows** are correctly tracked
- **States** are created successfully
- **Datasets** are created and deleted in right way



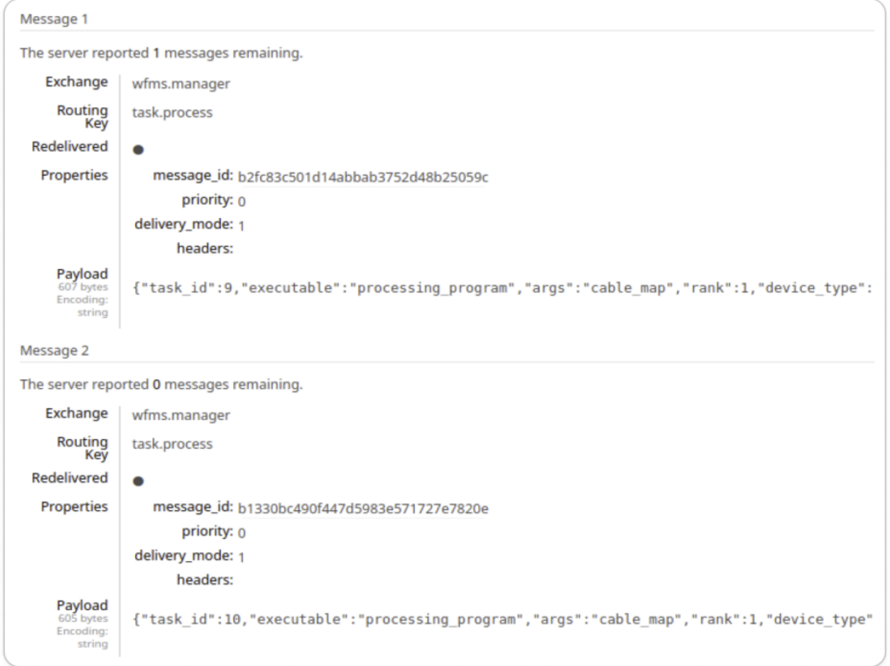
id	wflow_id	step	template	exec	args	priority	type	mode	retries	in_ds_name	out_ds_name	log_ds_name	status
10	5	reco	2StepsTest	processing_program	cable_map	1	CPU	Map	2	SPD_Run1001_Dataset_raw_run_10032025.output.1	SPD_Run1001_Dataset_raw_run_10032025.output.2	SPD_Run1001_Dataset_raw_run_10032025.log.2	FINISHED
9	5	decoding	2StepsTest	processing_program	cable_map	1	CPU	Map	2	SPD_Run1001_Dataset_raw_run_10032025	SPD_Run1001_Dataset_raw_run_10032025.output.1	SPD_Run1001_Dataset_raw_run_10032025.log.1	FINISHED



timestamp	description
2025-07-09 09:54:39.015007	**Status update:** Task status changed to DEFINED
2025-07-09 09:54:47.115724	**Status update:** Task status changed to RUNNING
2025-07-09 09:54:47.168544	**Meta:** Metadata_of_task_9
2025-07-09 09:55:02.239165	**Meta:** Metadata_of_task_9
2025-07-09 09:55:02.405051	**Status update:** Task status changed to FINISHED



timestamp	description
2025-07-09 09:54:39.162016	**Status update:** Task status changed to DEFINED
2025-07-09 09:55:02.602323	**Status update:** Task status changed to RUNNING
2025-07-09 09:55:17.468844	**Meta:** Metadata_of_task_10
2025-07-09 09:55:17.631080	**Status update:** Task status changed to FINISHED



Message 1

The server reported **1** messages remaining.

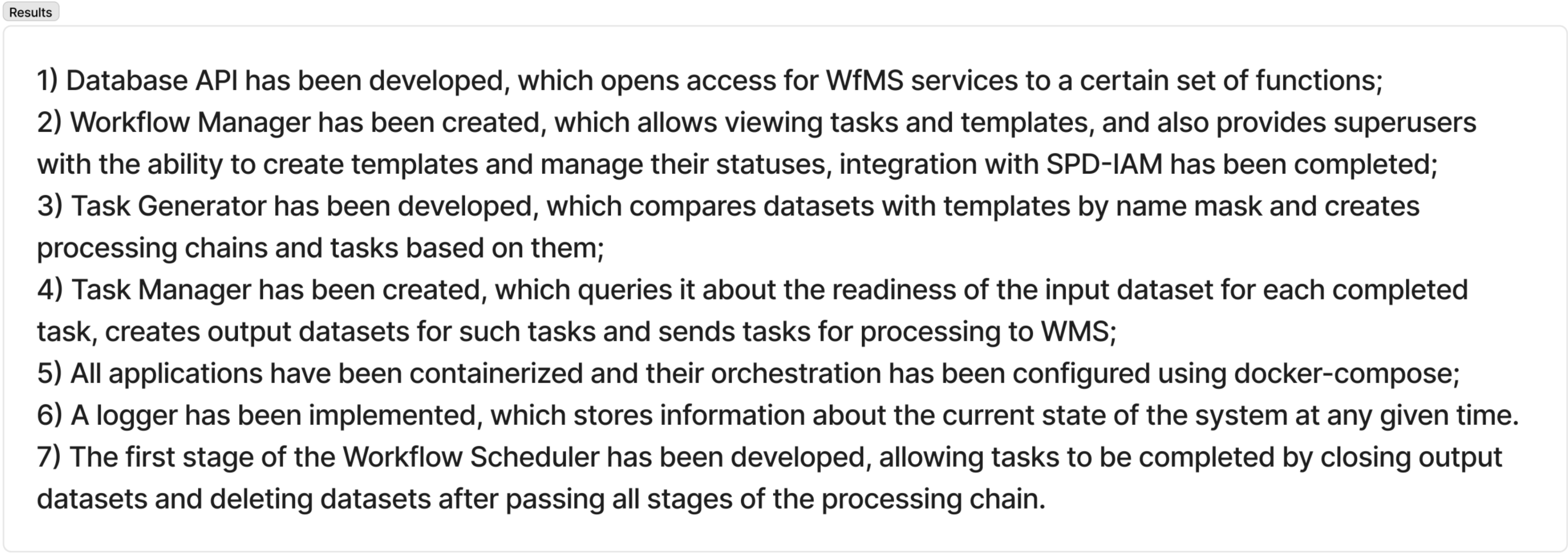
Exchange	wfms.manager
Routing Key	task.process
Redelivered	●
Properties	**message_id:** b2fc83c501d14abbab3752d48b25059c **priority:** 0 **delivery_mode:** 1 **headers:**
Payload 607 bytes Encoding: string	`{"task_id":9,"executable":"processing_program","args":"cable_map","rank":1,"device_type":`

Message 2

The server reported **0** messages remaining.

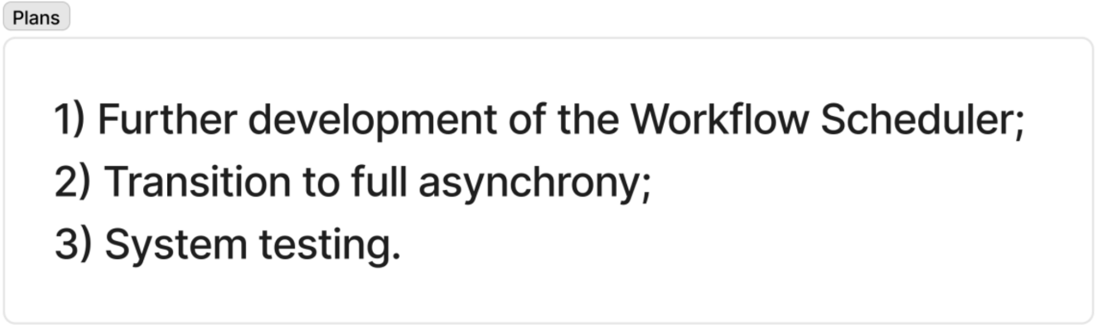
Exchange	wfms.manager
Routing Key	task.process
Redelivered	●
Properties	**message_id:** b1330bc490f447d5983e571727e7820e **priority:** 0 **delivery_mode:** 1 **headers:**
Payload 605 bytes Encoding: string	`{"task_id":10,"executable":"processing_program","args":"cable_map","rank":1,"device_type"`

Results and plans for the future



Results

1) Database API has been developed, which opens access for WfMS services to a certain set of functions;
2) Workflow Manager has been created, which allows viewing tasks and templates, and also provides superusers with the ability to create templates and manage their statuses, integration with SPD-IAM has been completed;
3) Task Generator has been developed, which compares datasets with templates by name mask and creates processing chains and tasks based on them;
4) Task Manager has been created, which queries it about the readiness of the input dataset for each completed task, creates output datasets for such tasks and sends tasks for processing to WMS;
5) All applications have been containerized and their orchestration has been configured using docker-compose;
6) A logger has been implemented, which stores information about the current state of the system at any given time.
7) The first stage of the Workflow Scheduler has been developed, allowing tasks to be completed by closing output datasets and deleting datasets after passing all stages of the processing chain.



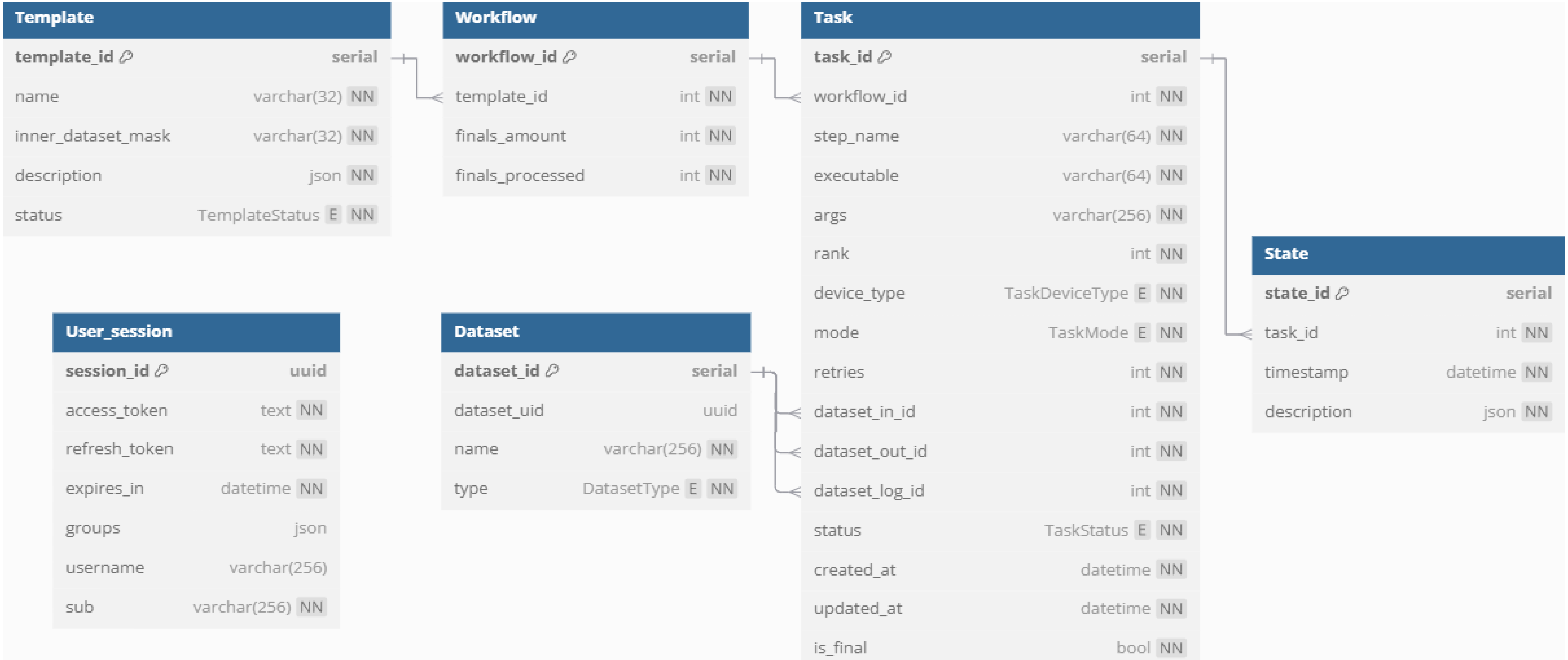
Plans

1) Further development of the Workflow Scheduler;
2) Transition to full asynchrony;
3) System testing.

Thank you for your attention!

Additional slides

Database structure



Template		
template_id 🔑	serial	
name	varchar(32)	NN
inner_dataset_mask	varchar(32)	NN
description	json	NN
status	TemplateStatus E	NN

Workflow		
workflow_id 🔑	serial	
template_id	int	NN
finals_amount	int	NN
finals_processed	int	NN

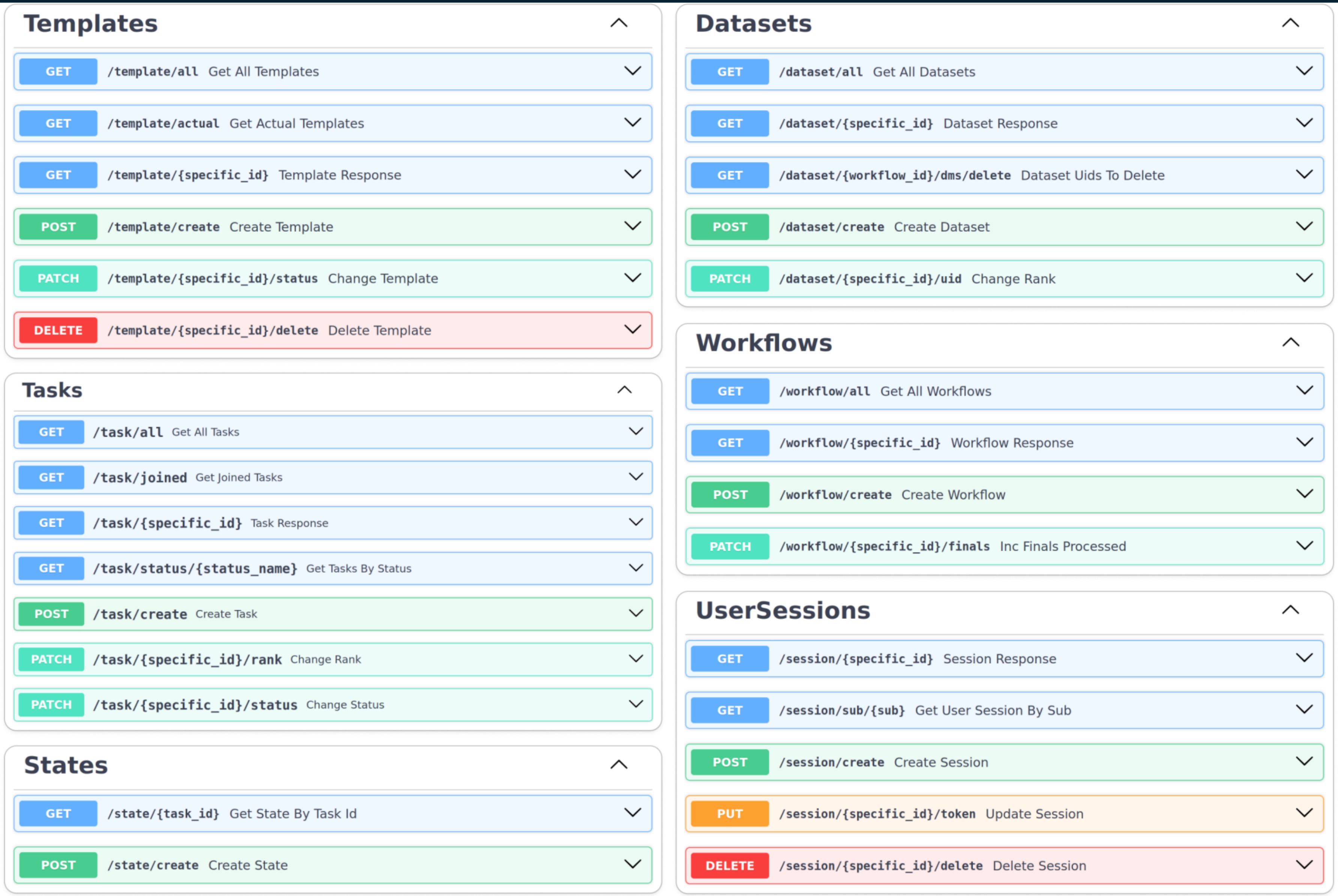
Task		
task_id 🔑	serial	
workflow_id	int	NN
step_name	varchar(64)	NN
executable	varchar(64)	NN
args	varchar(256)	NN
rank	int	NN
device_type	TaskDeviceType E	NN
mode	TaskMode E	NN
retries	int	NN
dataset_in_id	int	NN
dataset_out_id	int	NN
dataset_log_id	int	NN
status	TaskStatus E	NN
created_at	datetime	NN
updated_at	datetime	NN
is_final	bool	NN

State		
state_id 🔑	serial	
task_id	int	NN
timestamp	datetime	NN
description	json	NN

User_session		
session_id 🔑	uuid	
access_token	text	NN
refresh_token	text	NN
expires_in	datetime	NN
groups	json	
username	varchar(256)	
sub	varchar(256)	NN

Dataset		
dataset_id 🔑	serial	
dataset_uid	uuid	
name	varchar(256)	NN
type	DatasetType E	NN

Database API



Templates

GET /template/all Get All Templates

GET /template/actual Get Actual Templates

GET /template/{specific_id} Template Response

POST /template/create Create Template

PATCH /template/{specific_id}/status Change Template

DELETE /template/{specific_id}/delete Delete Template

Tasks

GET /task/all Get All Tasks

GET /task/joined Get Joined Tasks

GET /task/{specific_id} Task Response

GET /task/status/{status_name} Get Tasks By Status

POST /task/create Create Task

PATCH /task/{specific_id}/rank Change Rank

PATCH /task/{specific_id}/status Change Status

States

GET /state/{task_id} Get State By Task Id

POST /state/create Create State

Datasets

GET /dataset/all Get All Datasets

GET /dataset/{specific_id} Dataset Response

GET /dataset/{workflow_id}/dms/delete Dataset Uids To Delete

POST /dataset/create Create Dataset

PATCH /dataset/{specific_id}/uid Change Rank

Workflows

GET /workflow/all Get All Workflows

GET /workflow/{specific_id} Workflow Response

POST /workflow/create Create Workflow

PATCH /workflow/{specific_id}/finals Inc Finals Processed

UserSessions

GET /session/{specific_id} Session Response

GET /session/sub/{sub} Get User Session By Sub

POST /session/create Create Session

PUT /session/{specific_id}/token Update Session

DELETE /session/{specific_id}/delete Delete Session

- **FastAPI**
- Documentation and testing via **Swagger**