



Национальный исследовательский ядерный университет
«МИФИ»

Кафедра №40 «Физика элементарных частиц»



Выпускная квалификационная работа магистра

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ УПРАВЛЕНИЯ ПРОЦЕССАМИ ОБРАБОТКИ ПЕРВИЧНЫХ ДАННЫХ ЭКСПЕРИМЕНТА SPD

Студент

А. В. Плотников

Научный руководитель

Е. Ю. Солдатов

к.ф.-м.н.

Научный консультант

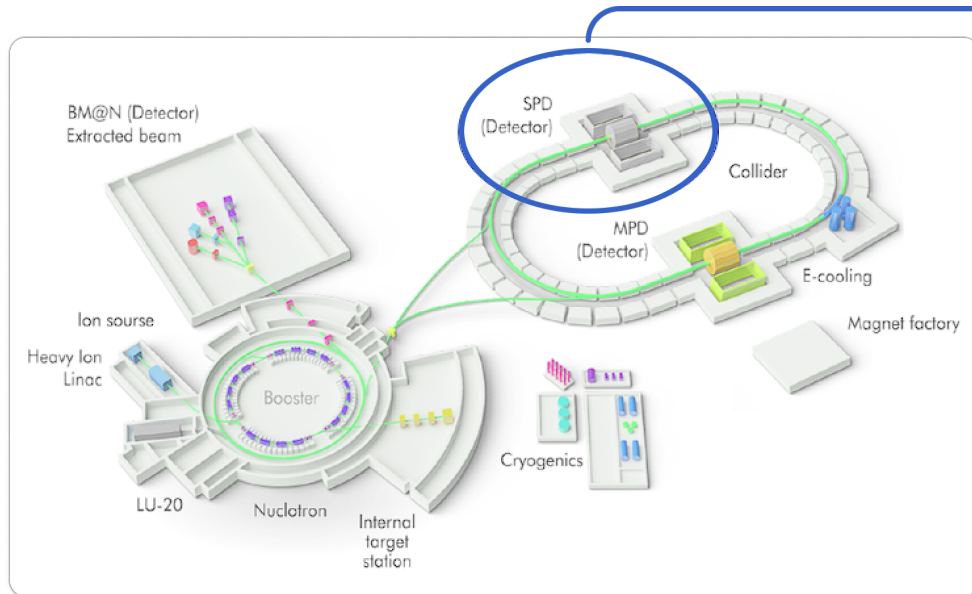
Д. А. Олейник

к.т.н

г. Москва, 2025



Spin Physics Detector



Комплекс NICA, Дубна

- Пересечение пучков с частотой 13 МГц.
- Предполагаемая частота регистрируемых событий $\sim 3 \text{ MHz}$ (при проектной светимости $10^{32} \text{ cm}^{-2} \text{ s}^{-1}$)
- Из приблизительной оценки получаем 20 ГБ/с (или 200 ПБ/год), что соответствует $\sim 3 \cdot 10^{13}$ событий/год.

Цель: всестороннее изучение спиновой структуры протона и дейтрона в поляризованных p-p, d-d и p-d-столкновениях

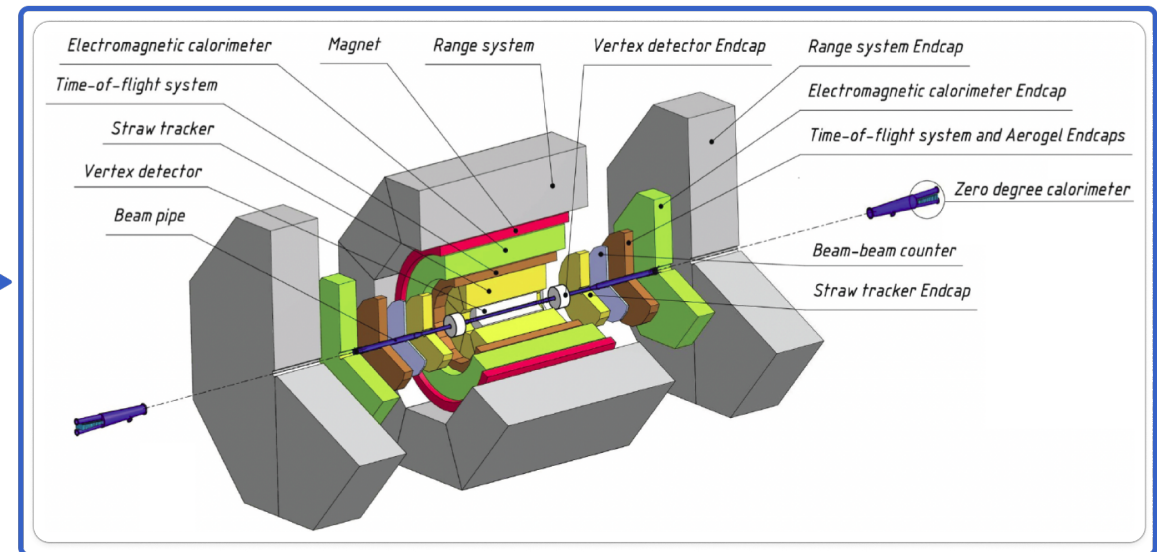


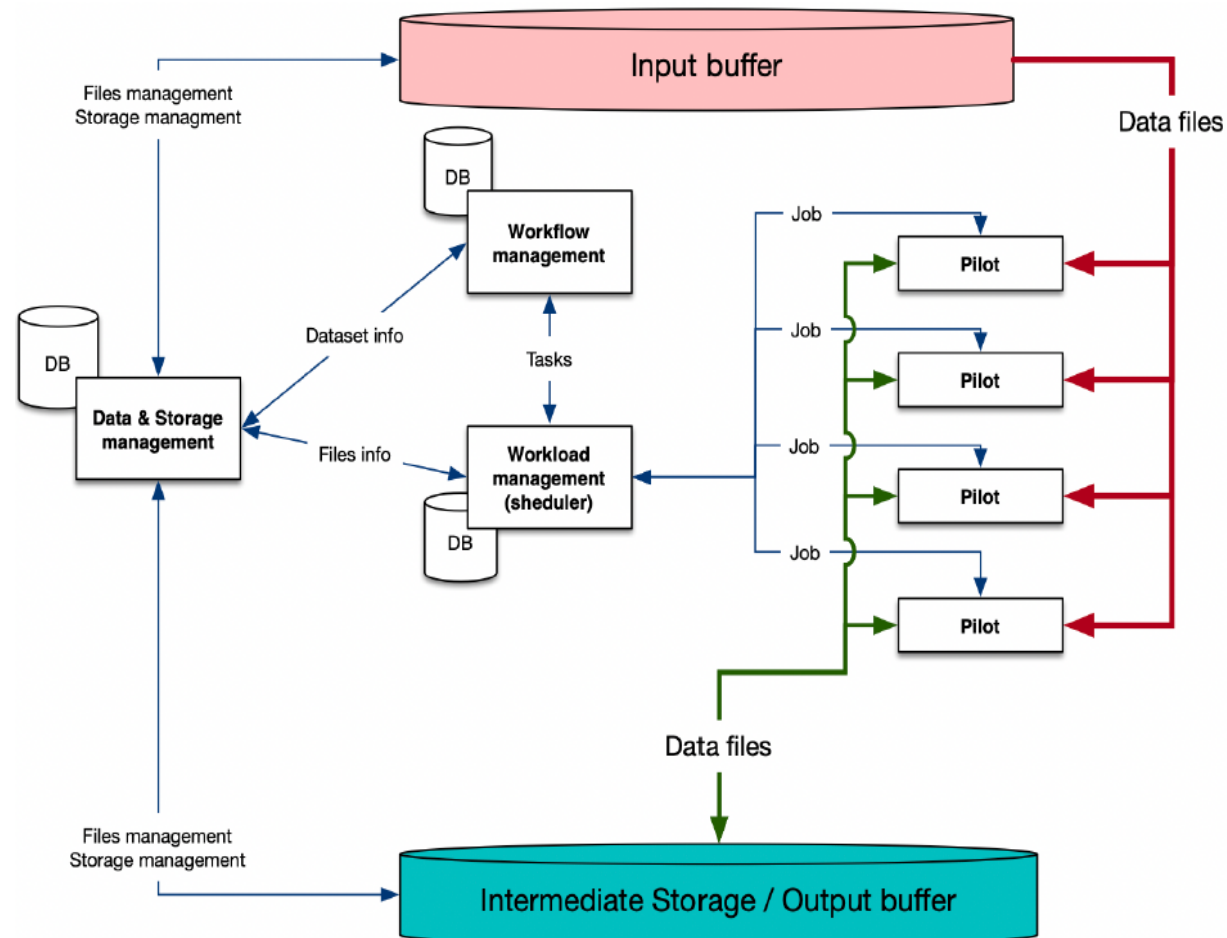
Схема детектора

SPD Online Filter

SPD Online Filter — специализированная программно-аппаратная система, предназначенная для предварительной обработки данных эксперимента SPD.

Система реализует многоступенчатую, высокопропускную методику обработки.

Основная, но не единственная, **цель обработки**: существенно уменьшить объём данных для последующего анализа и долговременного хранения.

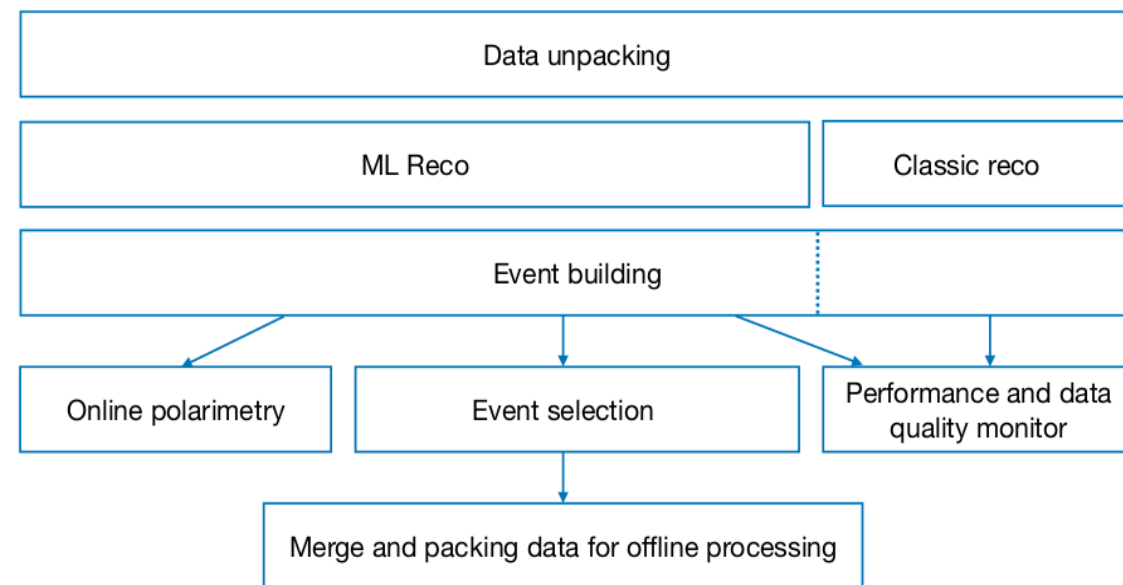


Архитектура Online Filter

Управление процессами обработки данных

Многоступенчатая обработка — набор последовательных этапов обработки данных. При этом на каждом этапе может обрабатываться достаточно большой объем данных.

Каждый шаг, кроме первого, принимает данные, обработанные на предыдущем этапе, и передает результаты следующему, обеспечивая преобразование информации из одного представления в другое.



Пример последовательных шагов обработки

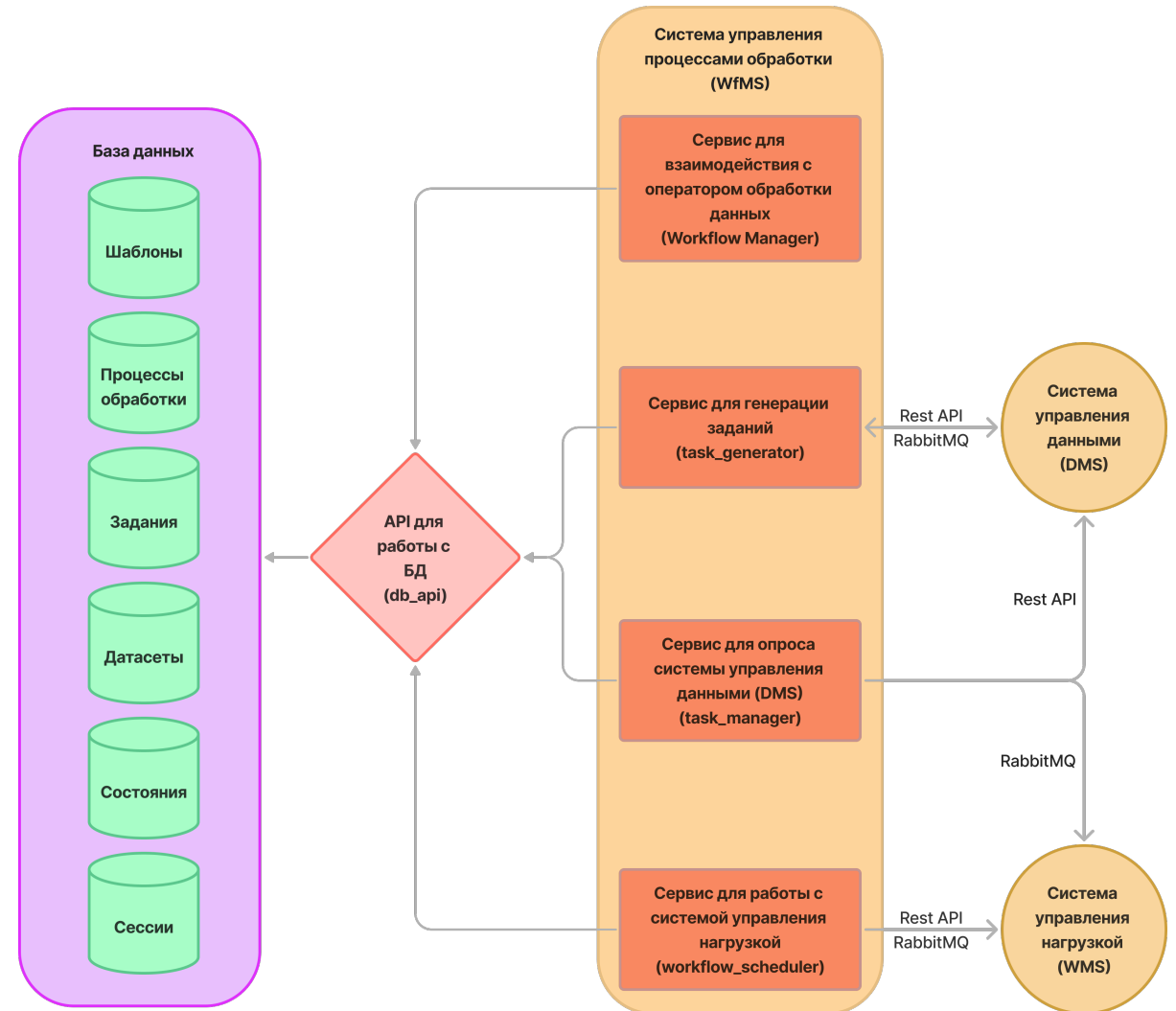
Система управления процессами обработки (WfMS)

Основная цель системы:

параллельное и наиболее результативное управление большим количеством процессов обработки данных и контроль статусов выполнения цепочек обработки для физического эксперимента SPD.

Основные задачи системы:

- Создание формального описания процесса;
- Контроль за выполнением обработки.



Взаимодействия микросервисов WfMS с БД

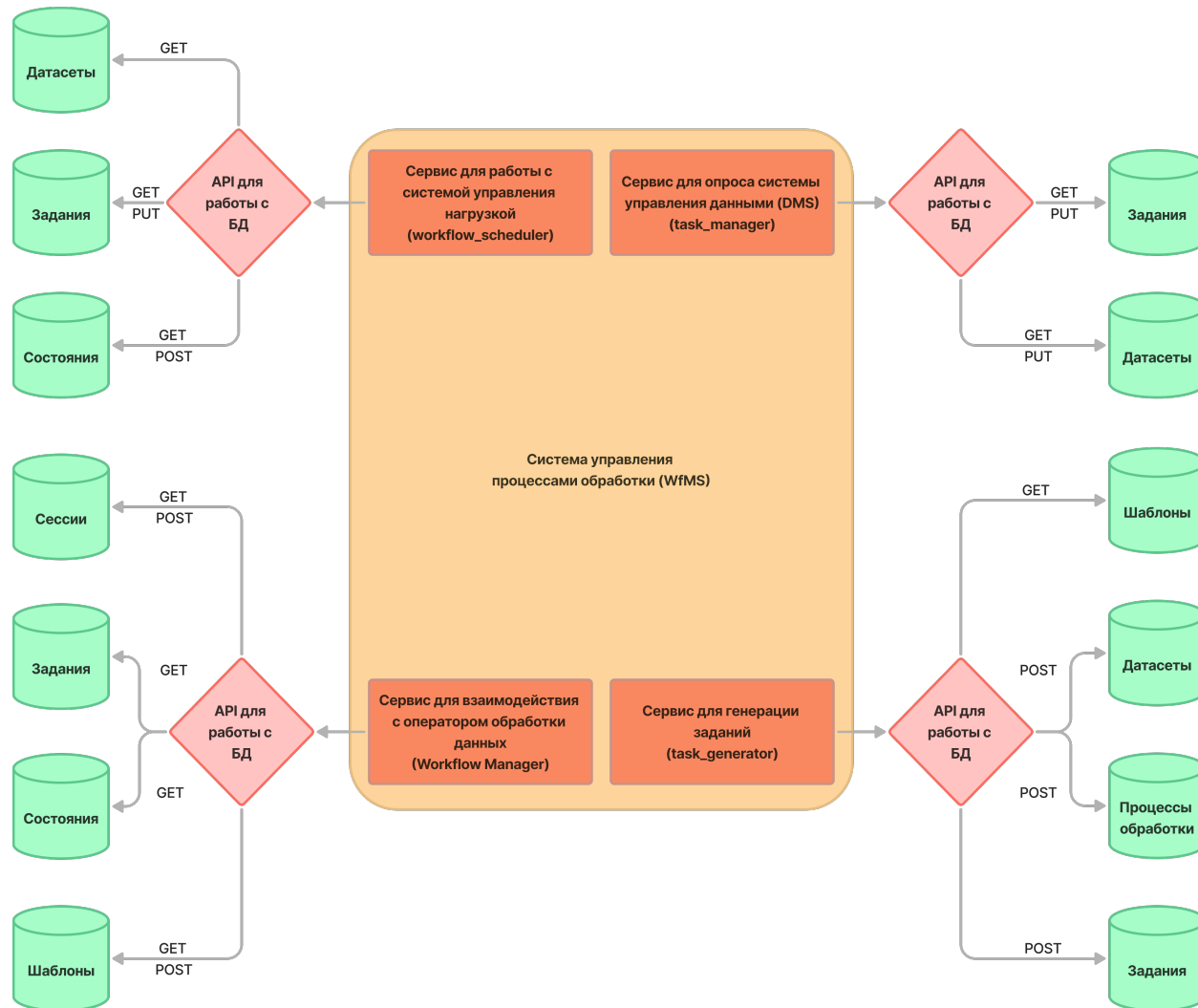
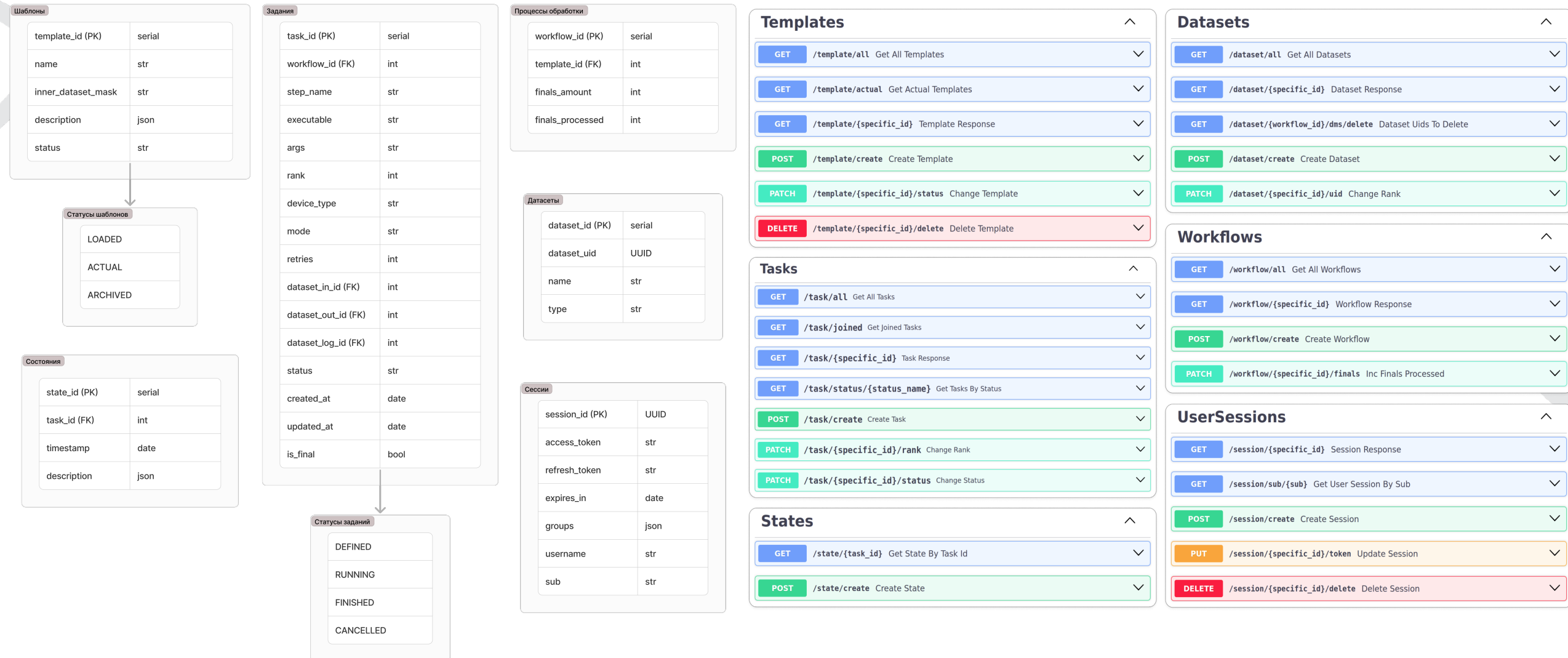


Схема взаимодействия микросервисов с БД

- База данных PostgreSQL развернута на отдельной виртуальной машине.
- Взаимодействие с базой данных осуществляется через REST API.

SQLAlchemy ORM
+
асинхронные сессии и asynpg
+
миграции Alembic

Структура базы данных и API к ней



Сервис для взаимодействия с оператором обработки данных

Основной функционал сервиса

1. Авторизация пользователей с помощью SPD-IAM;
2. Вывод CWL-шаблонов;
3. Вывод заданий;
4. Создание CWL-шаблонов суперпользователями (ввод в специальное поле / загрузка из файла / клонирование существующих шаблонов);
5. Предварительная валидация и запись в БД CWL-шаблонов;
6. Изменение статусов шаблонов суперпользователями на "ACTUAL" и "ARCHIVED";
7. Удаление суперпользователями шаблонов в статусе "LOADED".

- Пользовательский интерфейс
- Backend: FastAPI
- Шаблонизатор: Jinja2
- Frontend: Bootstrap (HTML + CSS + JS)

Workflow Manager

Templates ▾

Tasks

admin@jinr.ru

Logout

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	callback	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	callback	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

Workflow Manager

Templates ▾

Tasks

admin@jinr.ru

Logout

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

Создание шаблонов

- Создание шаблонов доступно только суперпользователям (в частности, отсутствует кнопка в интерфейсе)
- Предварительная валидация шаблонов с помощью инструментов cwltools
- Сохранение шаблона в БД

Workflow Manager

Templates ▾

Tasks

[admin@jinr.ru](#)

Logout

CWL Template

Template name

Inner dataset mask

Enter CWL here...

Complete

Сервис для генерации заданий

1. Получение зарегистрированных датасетов от DMS из RabbitMQ;
2. Сопоставление датасета по маске имени с нужным шаблоном;
3. Регистрация входного датасета в системе;
4. Создание процесса обработки по шаблону;
5. Создание выходного датасета и датасета логов в системе;
6. Создание заданий.

Шаблон

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

Датасеты и процесс обработки

```
[
  {
    "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"
  }
]
```

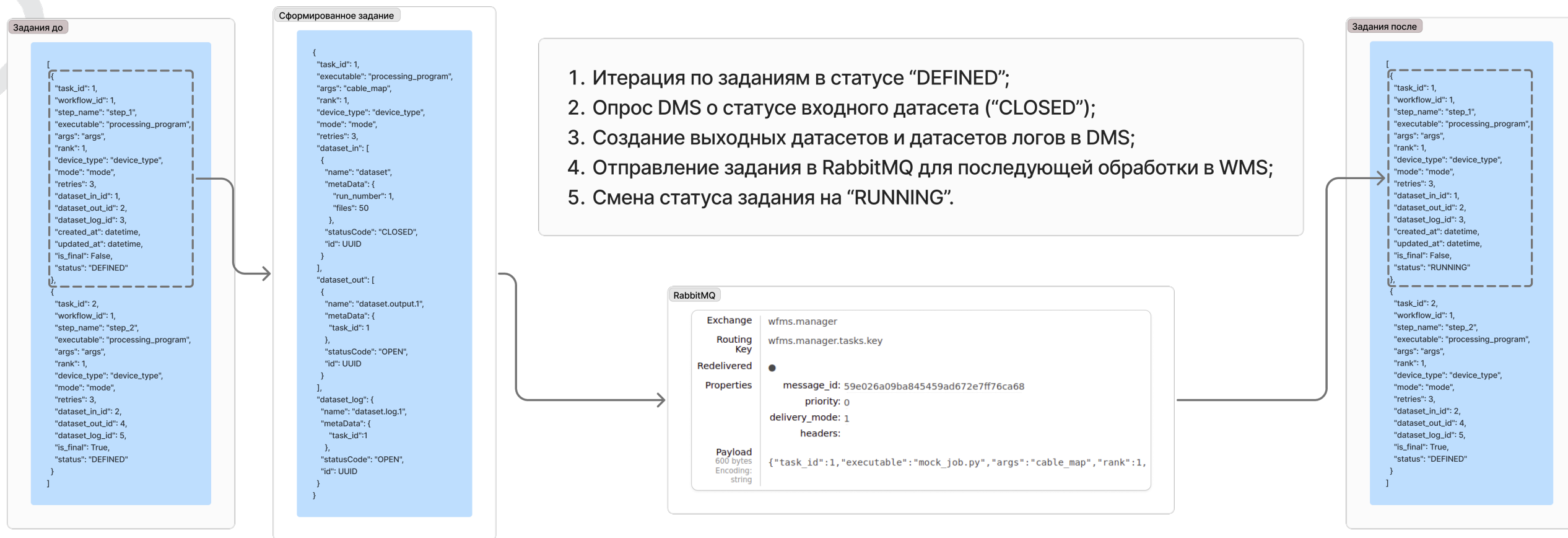
```
{
  "template_id": 1,
  "workflow_id": 1,
  "finals_amount": 1,
  "finals_processed": 0
}
```

Задания

```
[
  {
    "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"
  }
]
```

От шаблона к заданиям

Сервис для опроса системы управления данными (DMS)



Отправление заданий на обработку

Сервис для работы с системой управления нагрузкой (WMS)

Основной функционал сервиса

1. Завершение заданий с закрытием обработанных датасетов в DMS;
2. Отмена заданий на основе данных о заполненности выходного хранилища с оповещением WMS;
3. Изменение приоритетов заданий в процессе их выполнения.

1 этап

1. Итерация по заданиям в статусе "IN_PROGRESS";
2. Опрос WMS о состоянии задания;
3. Запись состояния;
4. Для успешно завершенных заданий:
 - а) Смена статуса выходных датасетов на CLOSED в DMS;
 - б) Смена статуса задания на "FINISHED";
5. Удаление входного и промежуточных датасетов.

Логирование, контейнеризация и оркестрация

- Логирование: 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.
```

- Контейнеризация и оркестрация:
Docker и Docker Compose

docker-compose

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

networks:
  wfms:
```

Результаты и дальнейшие планы

Результаты

- 1) разработан API для работы с базой данных, открывающий доступ для сервисов WfMS к определенному набору функций;
- 2) создан сервис для работы с оператором обработки данных, позволяющий просматривать задания и шаблоны, а также предоставляющий возможность суперпользователям создавать шаблоны и управлять их статусами, произведена интеграция с SPD-IAM;
- 3) разработан сервис для генерации заданий, сопоставляющий датасеты с шаблонами по маске имени и создающий по ним цепочки обработки и задания;
- 4) создан сервис для опроса DMS, опрашивающий его о готовности входного датасета для каждого готового задания, создающий выходные датасеты для таких заданий и отправляющий задания на обработку в WMS;
- 5) произведена контейнеризация всех приложений и настроена их оркестрация с помощью docker-compose;
- 6) внедрен logger, сохраняющий информацию об актуальном состоянии системы в каждый момент времени.
- 7) разработан первый этап сервиса для взаимодействия с WMS, позволяющий завершать задания с закрытием выходных датасетов и удалять датасеты после прохождения всех этапов цепочки обработки.

Дальнейшие планы

- 1) Последующая доработка сервиса для взаимодействия с WMS;
- 2) Переход к полной асинхронности;
- 3) Тестирование системы.



**Спасибо за
внимание!**