



# Разработка подсистемы управления данными комплекса промежуточного программного обеспечения SPD Online Filter

Студент

Научный консультант

П.А. Коршунова

Д.А. Олейник

Москва 2025



# Эксперимент SPD



Схема ускорительного комплекса NICA

**Детектор SPD** будет использоваться для изучения спиновой структуры протона и дейтрона и других связанных со спином явлений

**Основная цель эксперимента SPD** – всестороннее изучение неполяризованной и поляризованной глюонной составляющей нуклона

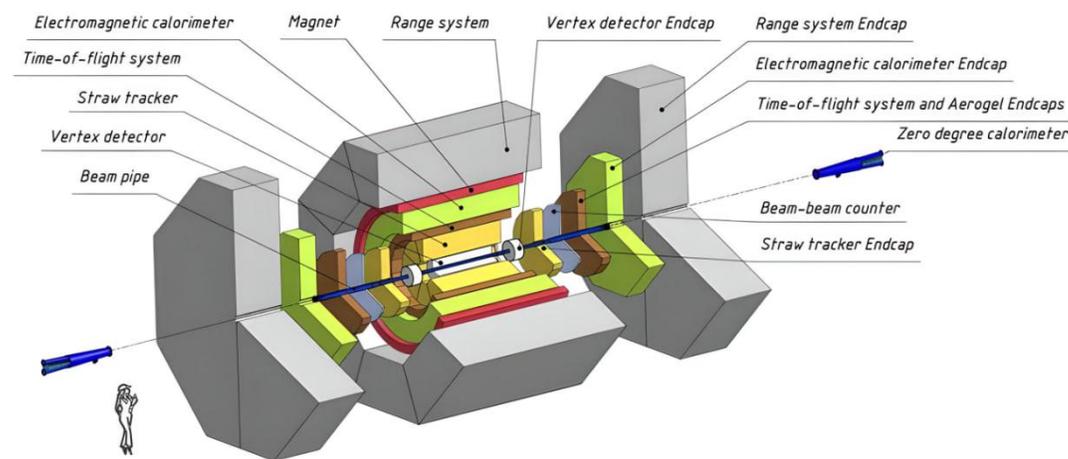


Схема детектора SPD

# Зачем нам SPD Online Filter?

**Сложность и широта изучаемых процессов**



**Невозможно использовать классический подход с использованием триггерной системы**



**Большой поток данных ~ 200 ПБ/год**



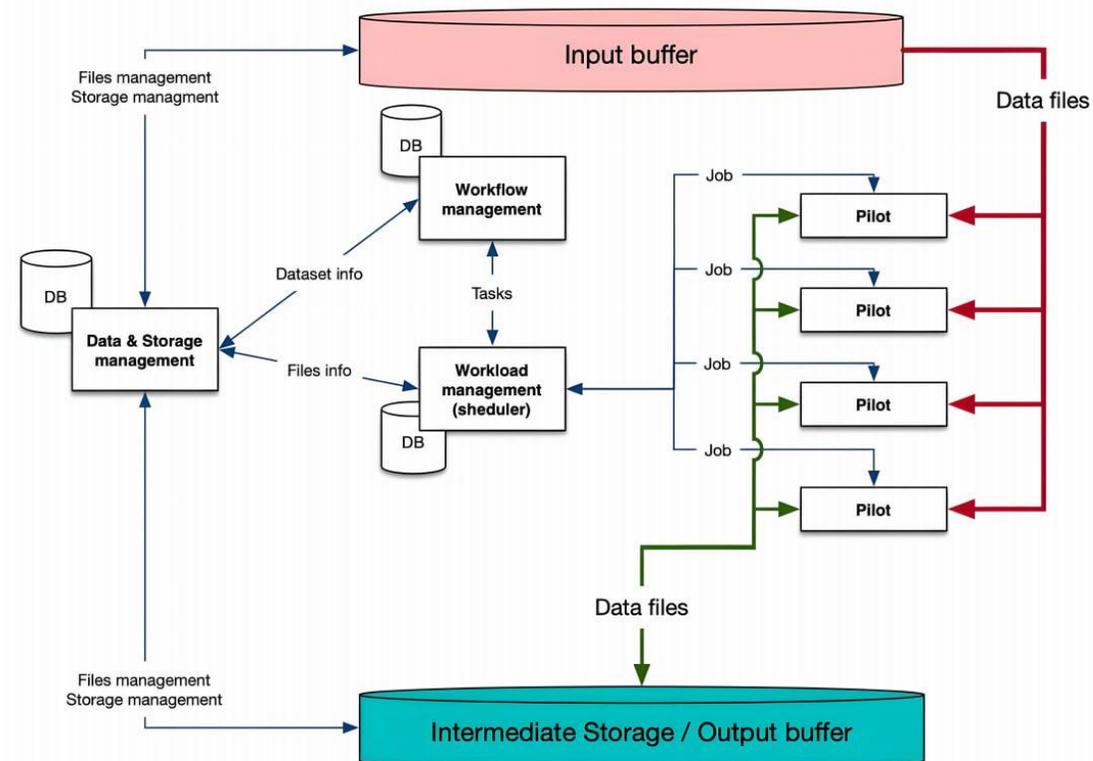
**Проблемы с обработкой и хранением данных**



**Необходимость разработки специализированной вычислительной системы с целью сокращения объема данных для долговременного хранения**

**SPD Online Filter** - это высокопроизводительная вычислительная система для высокопропускной обработки данных

Особенностью **высокопропускной обработки** данных является большой объем данных, как первичных, которые необходимо обработать, так и промежуточных, возникающих в процессе обработки



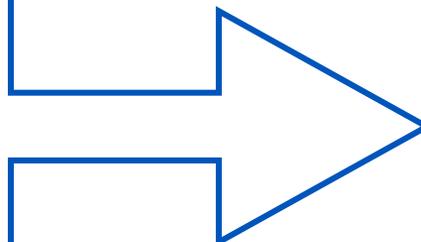
Архитектура SPD Online Filter

# Зачем нужна система управления данными?

Большой поток данных с  
детектора  
~ 200 ПБ/год

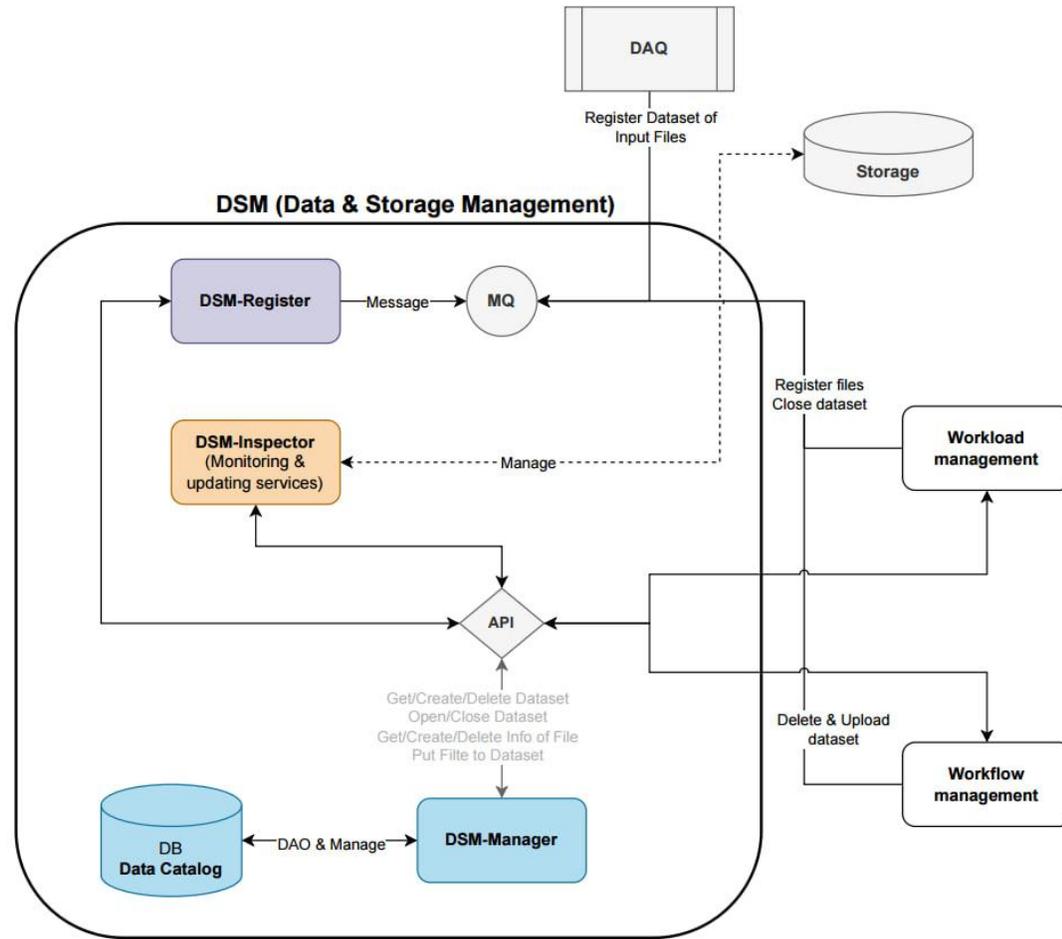


Большой объем  
промежуточных данных,  
полученных в процессе  
обработки



Необходимость  
разработки системы  
управления данными,  
которая должна  
обеспечивать контроль  
над хранением,  
организацией и  
целостностью данных

# Система управления данными



Архитектура DSM

## dsm-register

- сервис, принимающий в асинхронном режиме (через очередь сообщений) заявки на добавление/удаление данных в системе

## dsm-manager

- сервис, предоставляющий REST API к каталогу данных (размещение данных в каталоге, обращение к каталогу, изменение данных в каталоге)

## dsm-inspector

- набор фоновых сервисов для мониторинга и контроля состояния данных в хранилище

# dsm-register: шлюзы приема и отправки сообщений



Сервис должен слушать очередь сообщений и обрабатывать заявки на добавление/удаление данных в системе.

В качестве AMQP-брокера, который осуществляет маршрутизацию и подписку на нужные очереди, используется RabbitMQ.

Exchange	Routing Key	Назначение
	file.input	Приём информации о поступивших файлах на входной буфер
dsm.register (direct)	file.process	Приём информации о новых файлах, полученных в процессе обработки
	dataset.close	Приём заявки на закрытие набора файлов
	dataset.upload	Приём заявки на выгрузку файлов в наборе во внешнее хранилище
	dataset.delete	Приём заявки на удаление файлов в наборе на внутреннем хранилище

Exchange	Routing Key	Назначение
dsm.register (direct)	file.process.reply	Отправка информации о статусе регистрации файлов, полученных в процессе обработки
	dataset.input	Отправка информации о входных датасетах

Функциональность	URL	Контракт запроса	Контракт ответа
Внутреннее API			
<b>Управление информацией о наборах в каталоге</b>			
Создать набор	POST /dataset	Информация о наборе	ID набора
Получить набор	GET /dataset/<id>	ID набора	Информация о наборе
Изменить набор	PUT /dataset/<id>	ID набора и изменения	Обновленная информация о наборе
Удалить набор	DELETE /dataset/<id>	ID набора	-
Получить список наборов	GET /dataset	-	Информация о наборах
<b>Управление информацией о файлах в каталоге</b>			
Добавить файл	POST /file	Информация о файле и принадлежность к набору	ID файла
Получить файл	GET /file/<id>	ID файла	Информация о файле
Изменить файл	PUT /file/<id>	ID файла и изменения	Обновленная информация о файле
Удалить файл	DELETE /file/<id>	ID файла	-
Получить список файлов	GET /file	-	Информация о файлах
<b>Управление информацией о хранилище</b>			
Добавить хранилище	POST /storage	Информация о хранилище	ID хранилища
Получить хранилище	GET /storage/<id>	ID хранилища	Информация о хранилище
Изменить хранилище	PUT /storage/<id>	ID хранилища и изменения	Обновленная информация о хранилище
Удалить хранилище	DELETE /storage/<id>	ID хранилища	-
Получить список хранилищ	GET /storage	-	Информация о хранилищах
<b>Получение информации для мониторинга</b>			
Получение списка наборов, к которым принадлежит файл	GET /dataset/?file_id=<id>	ID файла	Информация о наборах
Список наборов в определенном статусе	GET /dataset/?status=<>	Статус набора	Информация о наборах
Список файлов в определенном статусе	GET /file/?status=<>	Статус файла	Информация о файлах
Поиск информации о файле по имени	GET /file/file_name/<name>	Имя файла	Информация о файле
Внешнее API			
<b>Взаимодействие с системой управления нагрузкой</b>			
Содержание набора	GET /file/?dataset_id=<id>	ID набора	Информация о файлах
<b>Взаимодействие с системой управления процессами обработки</b>			
Некоторые API управления наборами (создание выходного набора, получение статуса выходного набора)			

Сервис должен предоставлять REST API к каталогу данных.

В качестве веб-фреймворка используется асинхронный фреймворк FastAPI

Сервис состоит из набора **фоновых** задач:

- Удаление файлов на хранилищах
  - Контроль выгрузки файлов
  - Проверка целостности файлов
  - Контроль использования хранилища
- 
- Реализация в виде **демона**

# Доработка взаимодействия dsm-register с системой управления процессами обработки



## Было необходимо:

- ✓ Настроить механизм **отправки сообщений** с информацией о входных датасетах в новую очередь `dsm.register.dataset.input` (предварительно создать её)
- ✓ Реализовать прием заявок на удаление датасета:
  - ✓ Создать новый **consumer** для очереди `dsm.register.dataset.delete`
  - ✓ Написать обработчик сообщений, который будет запрашивать статус набора и, либо **отправлять** сообщение **обратно** в очередь (если статус OPEN), либо **устанавливать** набору статус `TO_DELETE`

# Механизм отправки сообщений о входных наборах



Exchange	dsm.register
Routing Key	file.input
Redelivered	o
Properties	
Payload 1406 bytes Encoding: string	{"meta":{"run_number":85},"files":[{"name":"input.test.e847e3c3-e463-455c-bda2-f2b9f937027a.raw","path":"input_18","size":50,"checkSum":"d144d83b5405dad60805ac13e52ac2a1"}]}

app-1 | 2024-09-26 11:15:00 INFO: Finish registering files. Dataset ID=ce931b56-86c7-4f11-bf81-f0c823f282fc CLOSED!

Получение сообщения с информацией о входных файлах из очереди dsm.register.file.input

Exchange	dsm.register
Routing Key	dataset.input
Redelivered	•
Properties	
Payload 109 bytes Encoding: string	{"id": "ce931b56-86c7-4f11-bf81-f0c823f282fc", "name": "input.test.cf3d9f65-1341-47aa-823a-3382502a85ab.raw"}

Проверка отправки сообщения с информацией о входном датасете в очередь dsm.register.dataset.input

# Прием и обработка заявок на удаление датасета в статусе CLOSED



	id [PK] uuid	name character varying (255)	meta_data json	status_code character varying (20)
1	05d4c366-6d5a-468b-9d80-b21ae2ba0ce0	input.test.c40a08fe-4a75-4157-a04d-92946d5c0967.raw.output...	{"task_id": 19}	OPEN
2	0859c70b-2c4d-45c8-93e2-a2e804d4f53c	input.test.c40a08fe-4a75-4157-a04d-92946d5c0967.raw.log.1	{"task_id": 19}	OPEN
3	14cd5201-7238-4b86-adb6-79c03866fbcf	input.test.c1471f0f-4b55-45d8-9d66-97c42ff7adff.raw	{"run_number": 96, "files": 10}	CLOSED

## Информация о наборе в статусе CLOSED

```
app-1 | 2025-01-18 16:26:06 INFO: [DSM-REGISTER-I004] [CONSUMER-HANDLER] Received message DatasetDeleteDto(id=UUID('14cd5201-7238-4b86-adb6-79c03866fbcf')) from queue '.dsm.register.dataset.delete' with delivery tag 56 [in /src/app/executor/rabbit/consumers/base/consumer_handler.py:50]
app-1 | 2025-01-18 16:26:06 INFO: Processing msg. [in /src/app/executor/services/processor/dataset_delete_processor.py:23]
app-1 | 2025-01-18 16:26:06 INFO: Dataset ID=14cd5201-7238-4b86-adb6-79c03866fbcf status changed to TO_DELETE. [in /src/app/executor/services/processor/dataset_delete_processor.py:35]
```

## Лог с информацией о получении сообщения и изменения статуса набора на TO\_DELETE.

	id [PK] uuid	name character varying (255)	meta_data json	status_code character varying (20)
1	05d4c366-6d5a-468b-9d80-b21ae2ba0ce0	input.test.c40a08fe-4a75-4157-a04d-92946d5c0967.raw.output...	{"task_id": 19}	OPEN
2	0859c70b-2c4d-45c8-93e2-a2e804d4f53c	input.test.c40a08fe-4a75-4157-a04d-92946d5c0967.raw.log.1	{"task_id": 19}	OPEN
3	14cd5201-7238-4b86-adb6-79c03866fbcf	input.test.c1471f0f-4b55-45d8-9d66-97c42ff7adff.raw	{"run_number": 96, "files": 10}	TO_DELETE

## Проверка изменения статуса набора в базе данных

**▼ Publish message**

Message will be published to the default exchange with routing key **dsm.register.dataset.delete**, routing it to this queue.

Delivery mode:

Headers: ?  =  String ▾

Properties: ?  =

Payload: 

```
{
  "id": "14cd5201-7238-4b86-adb6-79c03866fbcf"
}
```

Payload encoding:

## Заявка на удаление датасета в статусе CLOSED

# Прием и обработка заявок на удаление датасета в статусе OPEN

	id [PK] uuid	name character varying (255)	meta_data json	status_code character varying (20)
1	05d4c366-6d5a-468b-9d80-b21ae2ba0ce0	input.test.c40a08fe-4a75-4157-a04d-92946d5c0967.raw.output...	{"task_id": 19}	OPEN

## Информация о наборе в статусе OPEN

### ▼ Publish message

Message will be published to the default exchange with routing key **dsm.register.dataset.delete**, routing it to this queue.

Delivery mode: 1 - Non-persistent ▼

Headers: ? [input] = [input] String ▼

Properties: ? [input] = [input]

Payload: {  
 "id": "05d4c366-6d5a-468b-9d80-b21ae2ba0ce0"  
}

Payload encoding: String (default) ▼

Publish message

## Заявка на удаление датасета в статусе OPEN

```
app-1 | 2025-01-18 16:34:40 INFO: [DSM-REGISTER-I004] [CONSUMER-HANDLER] Received message DatasetDeleteDto(id=UUID('05d4c366-6d5a-468b-9d80-b21ae2ba0ce0')) from queue '.dsm.register.dataset.delete' with delivery tag 2547 [in /src/app/executer/rabbit/consumers/base/consumer_handler.py:50]
```

```
app-1 | 2025-01-18 16:34:40 INFO: Processing msg. [in /src/app/executer/services/processor/dataset_delete_processor.py:23]
```

```
app-1 | 2025-01-18 16:34:40 INFO: Dataset ID=05d4c366-6d5a-468b-9d80-b21ae2ba0ce0 OPEN. The message returned to the queue for deferred processing. [in /src/app/executer/services/processor/dataset_delete_processor.py:28]
```

Лог с информацией о получении сообщения и возвращении его в очередь для отложенной обработки.

## Алгоритм работы:

1. Получаем список наборов в статусе **CLOSED**
2. По каждому набору получаем его **содержимое** (информацию о файлах)
3. Формируем **список уникальных файлов** (по ID)
4. По каждому файлу:
  - проверяем наличие файла на хранилище
  - проверяем размер файла
  - проверяем контрольную сумму
  - если какая-то из проверок не пройдена, устанавливаем файлу статус **DAMAGED**
5. Если какой-либо файл в наборе **поврежден**, устанавливаем набору статус **FROZEN**

# Информация о файле и датасете



Полная информация о выбранном файле в БД

**Curl**

```
curl -X 'GET' \
'http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64defb5e6a17' \
-H 'accept: application/json'
```

**Request URL**

```
http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64defb5e6a17
```

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "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-5c02-4d17-87a9-64defb5e6a17" }</pre>

Информация о наборе, которому принадлежит выбранный файл

**Curl**

```
curl -X 'GET' \
'http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64defb5e6a17' \
-H 'accept: application/json'
```

**Request URL**

```
http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64defb5e6a17
```

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>[   {     "name": "input.test.b255570d-33bf-4dc3-9e6e-718df9a1a8ef.raw",     "metaData": {       "run_number": 49,       "files": 10     },     "statusCode": "CLOSED",     "id": "f61828be-64b5-44e8-9d18-a1a22068094d"   } ]</pre>

Меняем размер выбранного файла в каталоге

	id [PK] uuid	name character varying (255)	path character varying (255)	storage_id uuid	size integer
1	0fb1e1af-5c02-4d17-87a9-64defb5e6a17	input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw	input_40	b3307ad4-f2b3-4f3a-a390-4f4e2762c620	50
1	0fb1e1af-5c02-4d17-87a9-64defb5e6a17	input.test.a976a020-3de5-44e2-91ee-319e426eda2f.raw	input_40	b3307ad4-f2b3-4f3a-a390-4f4e2762c620	100

# Информация о файле и датасете после проверки



Лог с информацией о повреждении файла и изменении статуса набора на FROZEN

```
integrity-inspector-1 | 2025-01-19 13:31:47 INFO: File /data/SPDOF-buffers/input/input_40/input.test.a976a020-3de5-44e2-91ee-319e426eda2f.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_inspector/file_integrity_inspector.py:89]
```

## Информация о выбранном файле

Curl

```
curl -X 'GET' \
'http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64defb5e6a17' \
-H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/file/0fb1e1af-5c02-4d17-87a9-64defb5e6a17
```

Server response

Code	Details
200	<pre>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-5c02-4d17-87a9-64defb5e6a17" }</pre>

## Информация о наборе, которому принадлежит выбранный файл

Curl

```
curl -X 'GET' \
'http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64defb5e6a17' \
-H 'accept: application/json'
```

Request URL

```
http://10.220.16.177:8080/api/v1/dataset/?file_id=0fb1e1af-5c02-4d17-87a9-64defb5e6a17
```

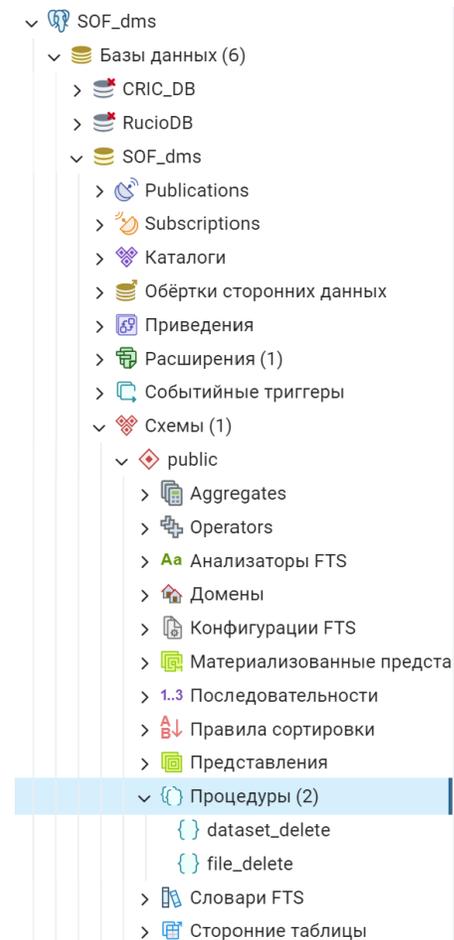
Server response

Code	Details
200	<pre>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"   } ]</pre>

# Удаление датасетов и файлов

## Состоит из:

1. Определения списка файлов, подлежащих удалению (храняя процедура в БД)
  - для каждого датасета со статусом `TO_DELETE` определяем **список файлов**
  - для каждого файла смотрим, **принадлежит ли он еще какому-нибудь датасету**
    - a. если не принадлежит, то устанавливаем ему статус `TO_DELETE`
    - b. если принадлежит, то отцепляем файл от данного датасета (удаляем соответствующую запись в таблице `dat_associations_file_dataset`)
2. Удаление файлов в датасетах
  - получаем **список файлов** со статусом `TO_DELETE`
  - **удаляем** каждый файл с хранилища и **помечаем** его в каталоге статусом `DELETED`
3. Удаление датасетов (храняя процедура в БД)
  - получаем список датасетов со статусом `TO_DELETE`
  - для каждого датасета смотрим, **все ли файлы** находятся **в статусе DELETED**
  - если да, то **помечаем датасет** статусом `DELETED`



# Работа процедуры file\_delete()



Запрос История запросов

```
1 SELECT dat_file.id, dat_file.name, dat_file.status_code, dat_file.path FROM dat_dataset
2 JOIN dat_association_file_dataset fd on fd.dataset_id = dat_dataset.id
3 JOIN dat_file on fd.file_id = dat_file.id
4 WHERE dat_dataset.status_code = 'TO_DELETE'
```

Data Output Сообщения Notifications

	id [PK] uuid	name character varying (255)	status_code character varying (20)	path character varying (255)
1	c81794fe-c57c-485c-89c7-0ace02f81883	input.test.41dad14a-e58c-4cd0-979a-16ee3e1cdb03.r...	CREATED	input_4
2	aa20e90d-6892-4d1a-aad1-1f3c5a7e4dc2	input.test.08850bb9-ae56-460e-875e-486f166c7533.r...	CREATED	input_4
3	b41ef691-3a5d-4d17-90d9-715a01e109a7	input.test.297eb0ea-c6e2-4261-acdf-b0d6f1c87482.ra...	CREATED	input_4
4	baa79005-1115-4bef-a82c-3b342c2213f0	input.test.8e607ad1-c88a-4e95-ba61-b1d539282662.r...	CREATED	input_4
5	6fa4d091-f983-4e21-92b8-ce68ea020732	input.test.d8a5c878-1ebe-4828-84b7-b37056bf6ac5.r...	CREATED	input_4
6	4255668b-26f2-4be7-98d0-64d82832c3ac	input.test.faa4e27f-ae2a-49d8-8934-5bfaa3102709.raw	CREATED	input_4
7	ec5ba1bf-b30b-47dc-8bc7-c76689bb58e1	input.test.8c04f7d8-2f26-4cc3-a81e-1b0dc796239e.ra...	CREATED	input_4
8	1e87276a-0462-4964-803e-7b6f96ea018d	input.test.3ea2562e-aa73-4c5e-8911-576db19d7fc5.r...	CREATED	input_4
9	8e238086-b493-459e-a567-627e00ab08e4	input.test.0aec3382-efe0-4986-8237-f07bc545248c.ra...	CREATED	input_4
10	760ecdb0-6f41-4515-b880-c7c21890ed81	input.test.451393ef-5ecb-4ed8-afd4-454acecc92d6.raw	CREATED	input_4

До запуска процедуры

После запуска процедуры

Запрос История запросов

```
1 SELECT dat_file.id, dat_file.name, dat_file.status_code, dat_file.path FROM dat_dataset
2 JOIN dat_association_file_dataset fd on fd.dataset_id = dat_dataset.id
3 JOIN dat_file on fd.file_id = dat_file.id
4 WHERE dat_dataset.status_code = 'TO_DELETE'
```

Data Output Сообщения Notifications

	id [PK] uuid	name character varying (255)	status_code character varying (20)	path character varying (255)
1	c81794fe-c57c-485c-89c7-0ace02f81883	input.test.41dad14a-e58c-4cd0-979a-16ee3e1cdb03.r...	TO_DELETE	input_4
2	aa20e90d-6892-4d1a-aad1-1f3c5a7e4dc2	input.test.08850bb9-ae56-460e-875e-486f166c7533.r...	TO_DELETE	input_4
3	b41ef691-3a5d-4d17-90d9-715a01e109a7	input.test.297eb0ea-c6e2-4261-acdf-b0d6f1c87482.ra...	TO_DELETE	input_4
4	baa79005-1115-4bef-a82c-3b342c2213f0	input.test.8e607ad1-c88a-4e95-ba61-b1d539282662.r...	TO_DELETE	input_4
5	6fa4d091-f983-4e21-92b8-ce68ea020732	input.test.d8a5c878-1ebe-4828-84b7-b37056bf6ac5.r...	TO_DELETE	input_4
6	4255668b-26f2-4be7-98d0-64d82832c3ac	input.test.faa4e27f-ae2a-49d8-8934-5bfaa3102709.raw	TO_DELETE	input_4
7	ec5ba1bf-b30b-47dc-8bc7-c76689bb58e1	input.test.8c04f7d8-2f26-4cc3-a81e-1b0dc796239e.ra...	TO_DELETE	input_4
8	1e87276a-0462-4964-803e-7b6f96ea018d	input.test.3ea2562e-aa73-4c5e-8911-576db19d7fc5.r...	TO_DELETE	input_4
9	8e238086-b493-459e-a567-627e00ab08e4	input.test.0aec3382-efe0-4986-8237-f07bc545248c.ra...	TO_DELETE	input_4
10	760ecdb0-6f41-4515-b880-c7c21890ed81	input.test.451393ef-5ecb-4ed8-afd4-454acecc92d6.raw	TO_DELETE	input_4

# Работа сервиса по удалению файлов

```

deleting-inspector-1 | 2025-01-19 14:17:13 INFO: Foarm file list with status TO_DELETE [in /src/files_deleting_inspector/file_delete_inspector.py:32
deleting-inspector-1 | 2025-01-19 14:17:13 INFO: HTTP Request: GET http://app:8080/api/v1/file/?file_status=TO_DELETE "HTTP/1.1 200 OK" [in /src/.venv/lib/python3.11/site-packages/httpx/_client.py:1038
deleting-inspector-1 | 2025-01-19 14:17:13 INFO: HTTP Request: GET http://app:8080/api/v1/storage/b3307ad4-f2b3-4f3a-a390-4f4e2762c620 "HTTP/1.1 200 OK" [in /src/.venv/lib/python3.11/site-packages/httpx/_client.py:1038
deleting-inspector-1 | 2025-01-19 14:17:13 INFO: File /data/SPDOF-buffers/input/input_4/input.test.8c04f7d8-2f26-4cc3-a81e-1b0dc796239e.raw DELETED! [in /src/files_deleting_inspector/file_delete_inspector.py:44
  
```

## Лог с информацией об удалении файлов, находящихся в статусе TO\_DELETE

Запрос История запросов

```

1 SELECT dat_file.id, dat_file.name, dat_file.status_code, dat_file.path FROM dat_dataset
2 JOIN dat_association_file_dataset fd on fd.dataset_id = dat_dataset.id
3 JOIN dat_file on fd.file_id = dat_file.id
4 WHERE dat_dataset.status_code = 'TO_DELETE'
  
```

Data Output Сообщения Notifications

	id [PK] uuid	name character varying (255)	status_code character varying (20)	path character varying (255)
1	c81794fe-c57c-485c-89c7-0ace02f81883	input.test.41dad14a-e58c-4cd0-979a-16ee3e1cdb03.r...	DELETED	input_4
2	aa20e90d-6892-4d1a-aad1-1f3c5a7e4dc2	input.test.08850bb9-ae56-460e-875e-486f166c7533.r...	DELETED	input_4
3	b41ef691-3a5d-4d17-90d9-715a01e109a7	input.test.297eb0ea-c6e2-4261-acdf-b0d6f1c87482.ra...	DELETED	input_4
4	baa79005-1115-4bef-a82c-3b342c2213f0	input.test.8e607ad1-c88a-4e95-ba61-b1d539282662.r...	DELETED	input_4
5	6fa4d091-f983-4e21-92b8-ce68ea020732	input.test.d8a5c878-1ebe-4828-84b7-b37056bf6ac5.r...	DELETED	input_4
6	4255668b-26f2-4be7-98d0-64d82832c3ac	input.test.faa4e27f-ae2a-49d8-8934-5bfaa3102709.raw	DELETED	input_4
7	ec5ba1bf-b30b-47dc-8bc7-c76689bb58e1	input.test.8c04f7d8-2f26-4cc3-a81e-1b0dc796239e.ra...	DELETED	input_4
8	1e87276a-0462-4964-803e-7b6f96ea018d	input.test.3ea2562e-aa73-4c5e-8911-576db19d7fc5.r...	DELETED	input_4
9	8e238086-b493-459e-a567-627e00ab08e4	input.test.0aec3382-efe0-4986-8237-f07bc545248c.ra...	DELETED	input_4
10	760ecdb0-6f41-4515-b880-c7c21890ed81	input.test.451393ef-5ecb-4ed8-afd4-454acecc92d6.raw	DELETED	input_4

## Измененная информация об удаленных файлах

# Работа процедуры dataset\_delete()

До запуска процедуры

	<b>id</b> [PK] uuid	<b>name</b> character varying (255)	<b>meta_data</b> json	<b>status_code</b> character varying (20)
1	bf04dfe1-db61-4195-b584-45de6ea8f04d	input.test.d95edab2-8f40-4b67-9fd7-01565512228f.raw	{"run_number": 3, "files": 10}	TO_DELETE

После запуска процедуры

	<b>id</b> [PK] uuid	<b>name</b> character varying (255)	<b>meta_data</b> json	<b>status_code</b> character varying (20)
1	bf04dfe1-db61-4195-b584-45de6ea8f04d	input.test.d95edab2-8f40-4b67-9fd7-01565512228f.raw	{"run_number": 3, "files": 10}	DELETED

# Заключение

В работе было изучено, что такое эксперимент [SPD](#) и система [Online Filter](#), а также зачем нужна и как устроена [система управления данными](#).

Был добавлен [дополнительный функционал](#) для взаимодействия сервиса [dsm-register](#) с системой управления процессами обработки и был [частично реализован сервис dsm-inspector](#). Также по ходу выполнения данных задач были добавлены дополнительные возможности в сервис [dsm-manager](#).

## Дальнейшие планы

### [dsm-inspector:](#)

Реализовать фоновые сервисы для:

- Контроля выгрузки файлов
- Контроля использования хранилища

### [dsm-register:](#)

Реализовать обработку сообщений из очередей:

- `dsm.register.dataset.closed`
- `dsm.register.dataset.upload`



**Спасибо за  
внимание!!!**



## **Дополнительные слады**



# Промежуточное ПО

## Система управления данными

- регистрация, каталогизация, контроль целостности

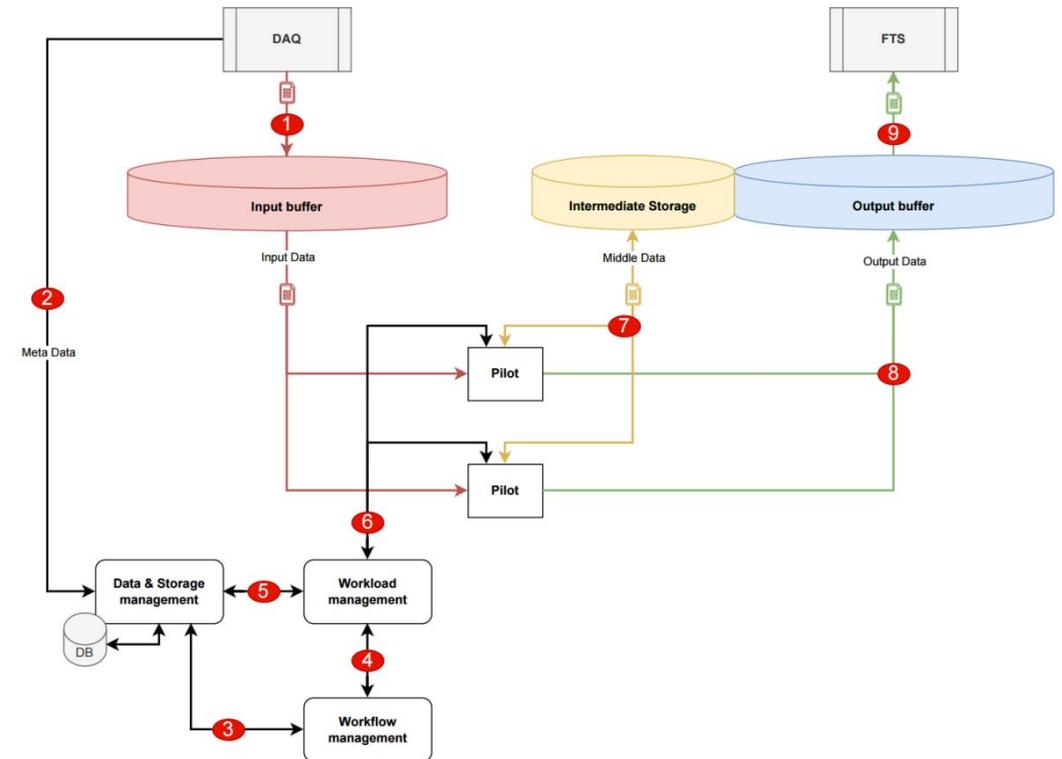
## Система управления процессами обработки

- формирование и контроль исполнения этапов обработки данных

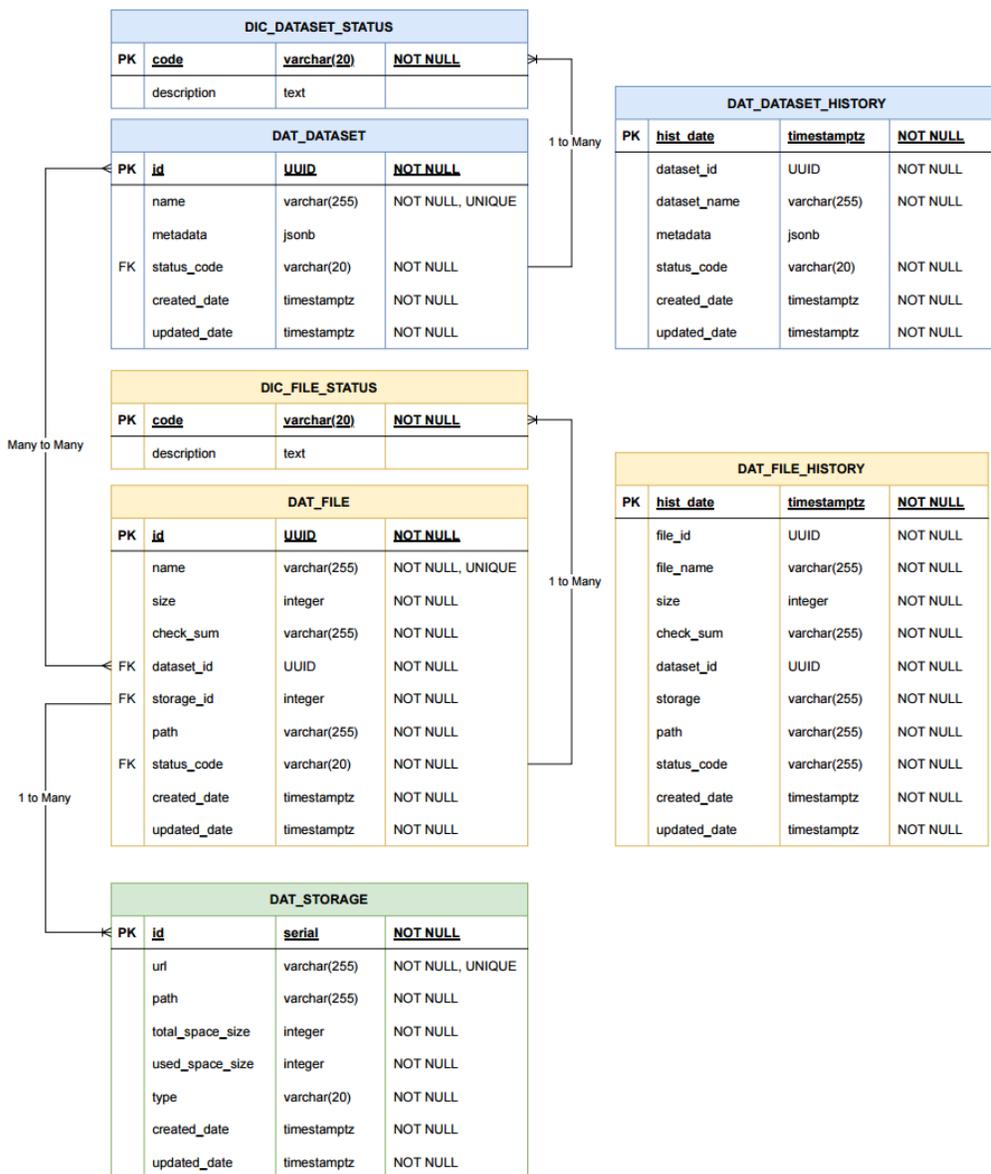
## Система управления нагрузкой

- реализация этапов обработки (формирование задач, отправка задач пилотам)

**Pilot** – приложение, работающее на вычислительном узле и исполняющее задачи



# Структура БД



Код статуса	Описание
CREATED	Файл добавлен в систему
DAMAGED	Файл повреждён
TO_DELETE	Файл с пометкой «На удалении»
UPLOADING	Файл выгружается
DELETED	Файл удалён из системы

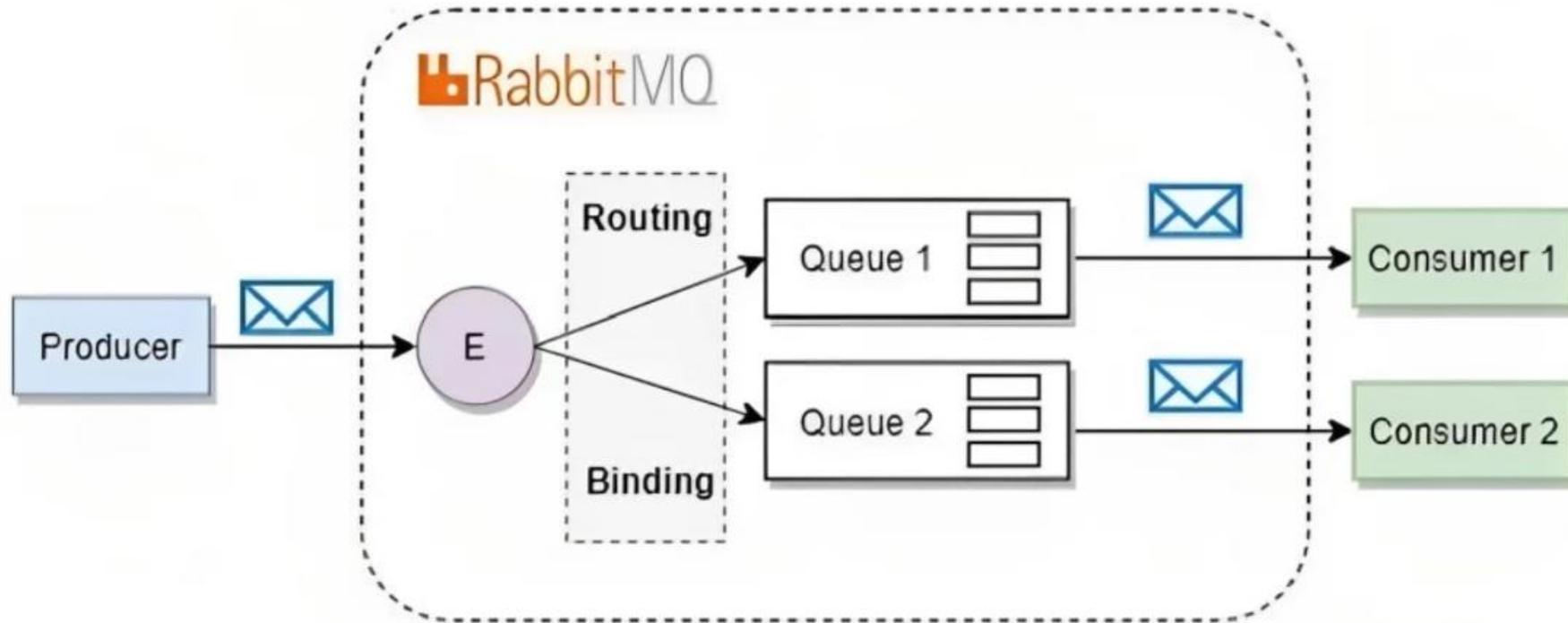
Справочник статусов файлов

Код статуса	Описание
OPEN	Набор открыт
CLOSED	Набор закрыт
FROZEN	Набор временно «заморожен»
TO_UPLOAD	Набор с пометкой «На выгрузку»
UPLOADING	Набор выгружается
TO_DELETE	Набор с пометкой «На удалении»
DELETED	Набор удалён из системы

Справочник статусов датасетов

# RabbitMQ

- брокер сообщений, который служит посредником для обмена информацией между различными системами



Основные компоненты RabbitMQ

# Сконфигурированные очереди RabbitMQ

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