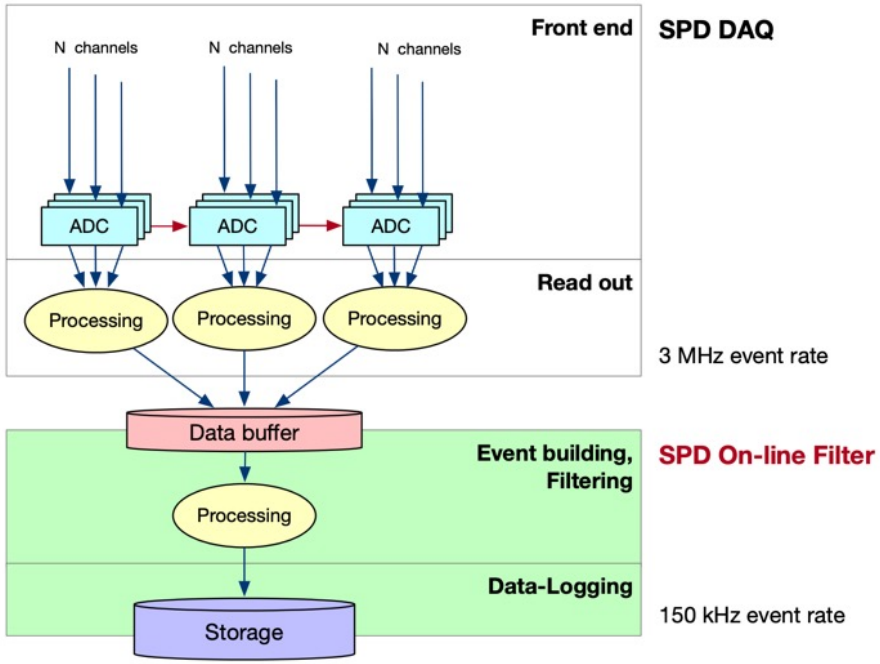
Система управления данными для SPD Online Filter

Выполнил: Терещенко Дмитрий Владиславович
Руководитель от ОИЯИ: Олейник Данила Анатольевич

Источники данных в SPD OnLine Filter

Data-acquisition - Первичные данные



SPD DAQ обрабатывает сигналы, генерируемые детекторами, и передаёт данные на входной буфер SPD Online Filter. **Объём данных**:

- Скорость потока до 20 Гб/с (в зависимости от режима работы коллайдера)
- Данные поступают в несколько потоков, кажды которых пишет файлы в согласованную структуру директорий.
- По окончанию записи директории DAQ Информирует SPD OnLine Filter

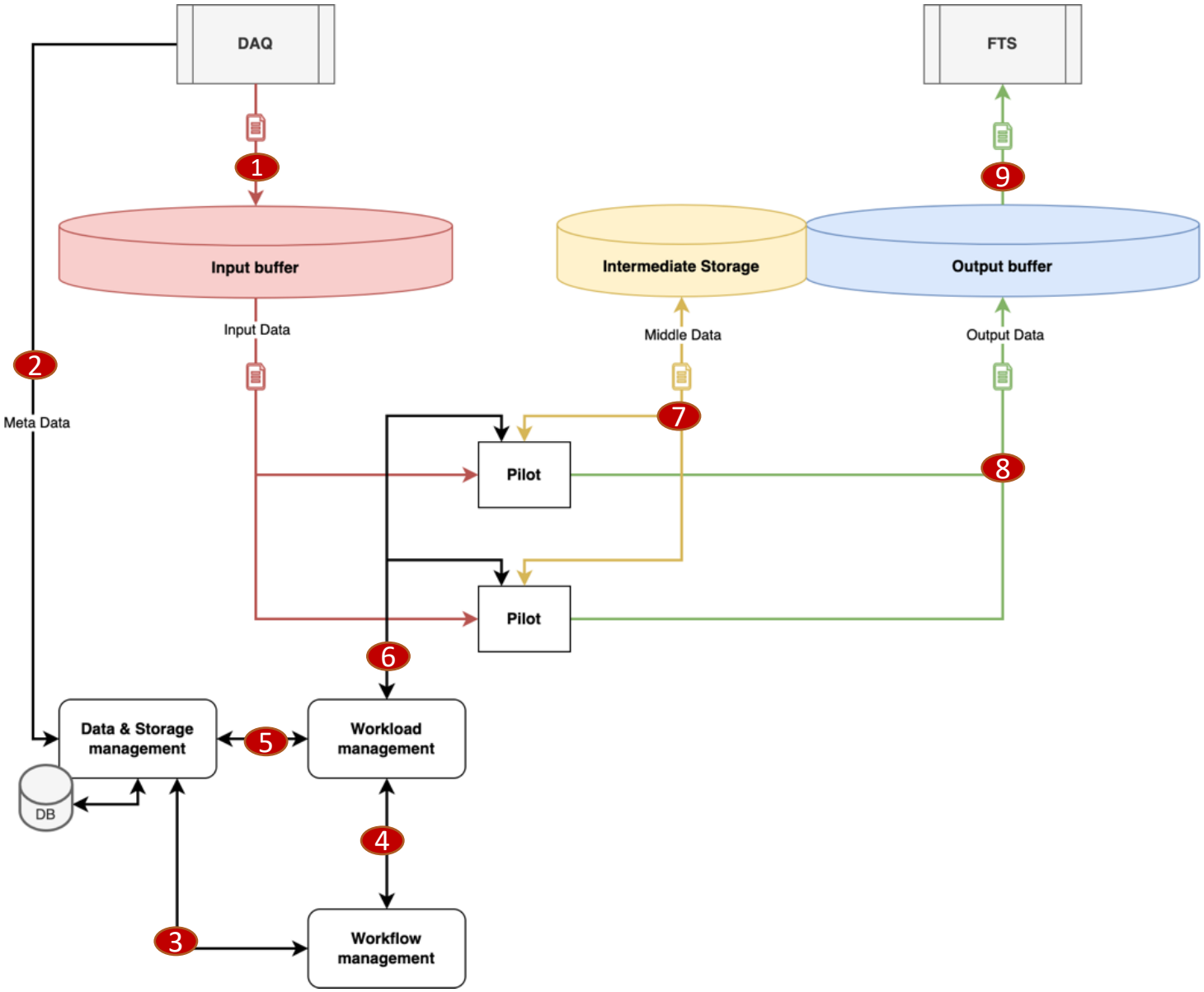
На каждом шаге обработки данных возникают вторичные данные.

Система управления процессом обработки создает у удаляет промежуточные и конечные наборы данных (датасеты). Объединение логическое, не релевантное к уровню физического хранения.

Система управления нагрузкой “заполняет” наборы файлами.

Необходима специализированная система осуществляющая функции каталогизации данных и реализующее управление данными на хранилищах

Потоки данных в SPD Online Filter

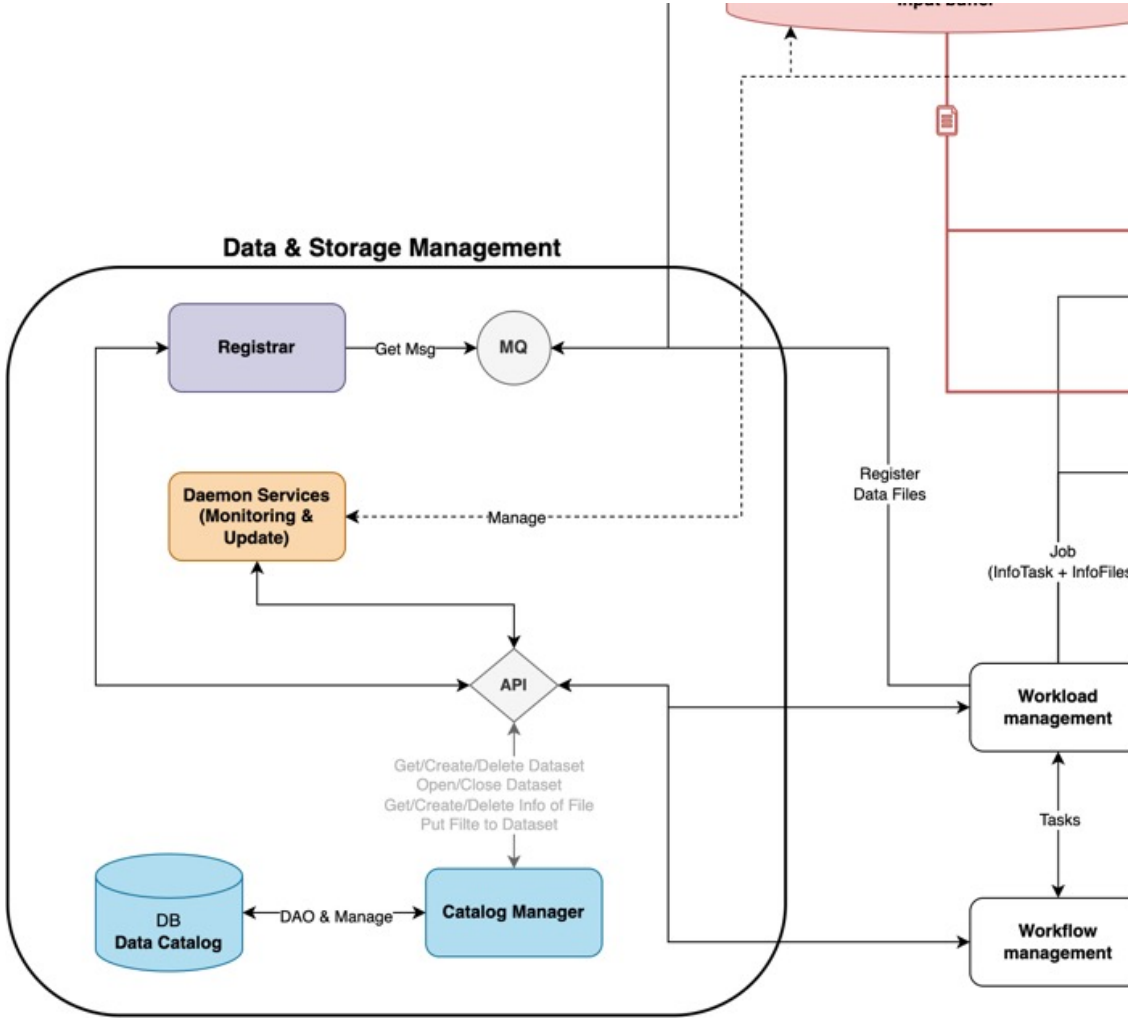


1) Набор входных данных (файлов) с DAQ
2) Мета-информация о первичном наборе
3) Мета-информация о наборе файлов
4) Мета-информация о наборе файлов + описание задач + результат задач
5) Мета-информация о файлах в наборе
6) Мета-информация о файле + описание задачи
7) Промежуточные данные (файлы)
8) Итоговые данные (файлы)
9) Данные, выгружаемые в FTS

Архитектура и функциональность DSM

Data and storage management system for SPD OnLine Filter

- **DSM-Register (Регистрация данных):** размещение мета-информации о файле (имя, физический путь к нему и т.п.), загруженном в систему + его наборе
- **DSM-Manager:**
  - **Каталогизация файлов:** предоставление мета-информации об организации загруженных данных в систему
  - **Управление наборами (dataset-ми):** создать dataset, добавить файл в dataset, закрыть dataset; удалить dataset; дать информацию о содержимом dataset (файлах в датасете)
- **DSM-Inspector (Сервисы):** удаление файлов на хранилищах, проверка целостности файлов, контроль использования хранилища («темные» данные)



Модель метаданных

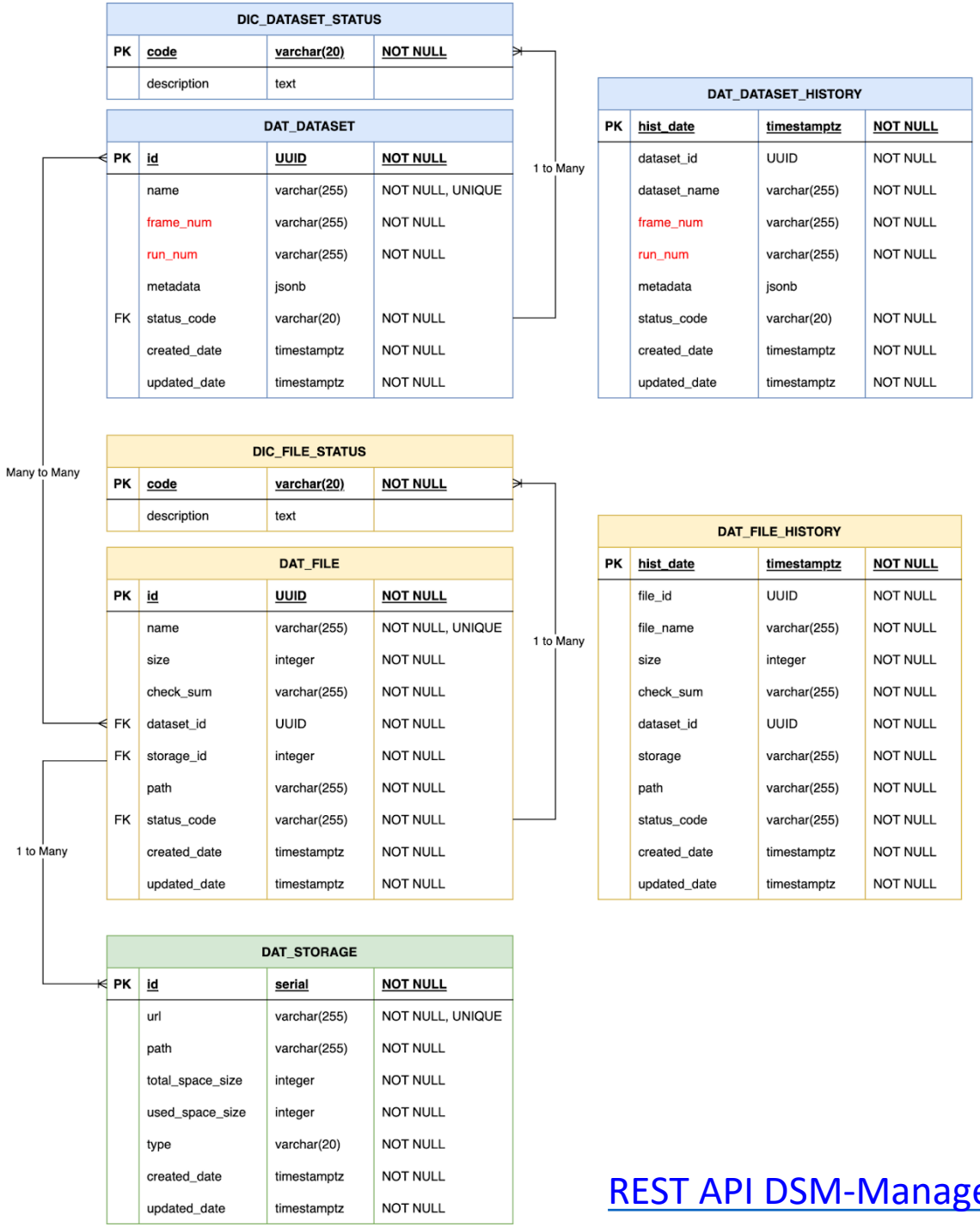
*Для того, чтобы отделить информацию об организации данных от самих данных, в системе отдельно будут хранится **мета-данные**.*

Таблицы:

- **DAT_FILE** - каталог файлов, обрабатываемых системой
- **DAT_DATASET** - каталог наборов файлов, созданных системой при построении вычислений
- **DAT_FILE_HISTORY и DAT_DATASET_HISTORY** - архивные таблицы по файлам и наборам
- **DIC_FILE_STATUS и DIC_DATASET_STATUS** - справочники, хранящие возможные статусы по файлам и наборам соответственно
- **DAT_STORAGE** - информация о хранилищах

Предполагаемые доп. механизмы:

- Партиционирование таблиц DAT_FILE и DAT_DATASET с ключём status для архивирования записей со статусом DELETED
- Партиционирование таблиц DAT_FILE_HISTORY и DAT_DATASET_HISTORY по ключу hist_date. Размер партиции = около месяца.
- Триггер на DAT_FILE и DAT_DATASET (на insert и update) для записи в таблицы DAT_FILE_HISTORY и DAT_DATASET_HISTORY



[REST API DSM-Manager]

DSM-Register

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

Exchange	Routing Key	Msg	Алгоритм
dsm.register (direct)	add.input	Info of files in Input Frame	Создаём набор по номеру frame-а со статусом OPEN. Добавляём файлы с привязкой к этому набору, устанавливаем первичный статус CREATE.
	add.process	Info of Dataset + Info of files	Создаём набор (если его нет) в статусе OPEN, добавляем файлы в набор, устанавливаем статус CREATE.
	close	Info of Dataset	Закрываем набор для регистрации в нём файлов (статус CLOSED)
	upload	Info of Dataset	Помечаем набор на выгрузку (статус TO_UPLOAD)
	delete	Info of Dataset	Помечаем набор на удаление (статус TO_DELETE)

Сервисы DSM-Inspector (status система)

- **Удаление файлов на хранилищах**
  - Одним процессом получаем наборы файлов со статусом TO_DELETE. По каждому файлу в наборе проверяем статус.
    - Если есть не DELETED, то ставим статус TO_DELETE.
    - Иначе устанавливаем для набора статус DELETED.
  - Вторым процессом получаем файлы со статусом TO_DELETE. Проверяем, что все его наборы находятся в статусе TO_DELETE. Удаляем файл на хранилище. Устанавливаем статус DELETED.
- **Контроль выгрузки данных во внешнюю систему**
- **Проверка целостности файлов**
- **Контроль использования хранилища**

Технологический стек DSM

DSM-Register	DSM-Manager	DSM-Inspector
• Python 3.11 • Docker • Pika (RabbitMQ) • Aiohttp • …	• Python 3.11 • Docker • FastAPI + Unicorn • Pydantic Schemas • Asyncio + Aiopg • SQLAlchemy (PostgreSQL) • Alembic (Migration) • …	• Python 3.11 • Docker • Celery Cron Jobs • Aiofiles • Aiohttp • …