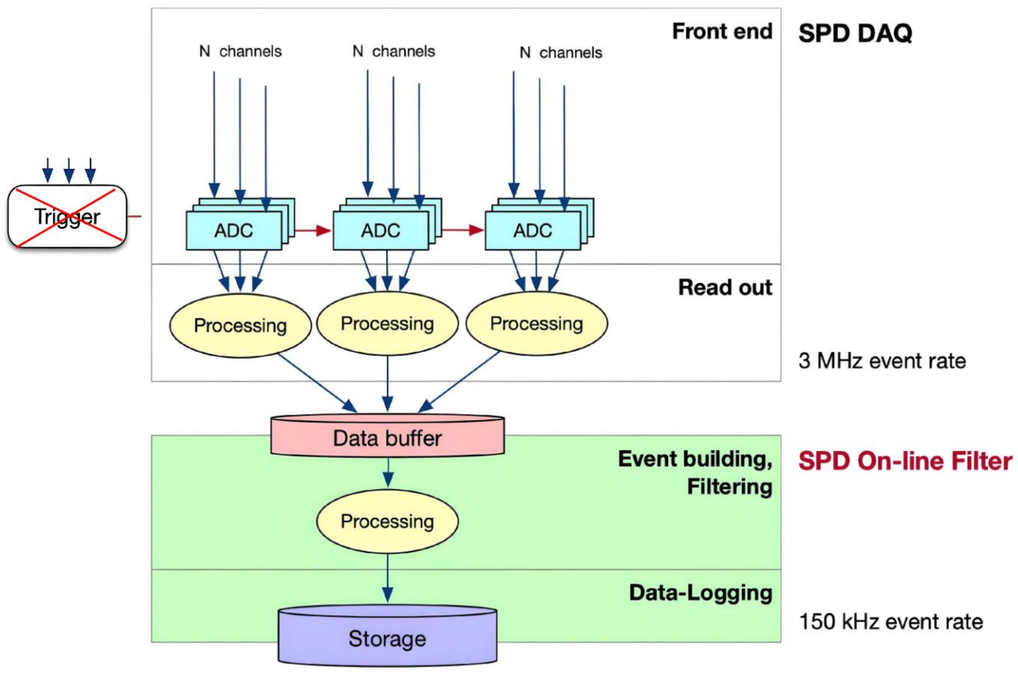
SPD Online Filter Middleware

Romanychev L.R., Greben N.V., Oleynik D.A., Korshunova P.A., Plotnikov A.V.

«SPD Online Filter» is a hardware-software system designed for multi-stage, high-throughput processing of data from the SPD detector. Its main task is the primary processing of data, in order to reduce its volume for long-term storage and subsequent full processing. The «SPD Online Filter» comprises a dedicated compute cluster, a middleware software, and a set of application-level services. The middleware layer consists of three microservice-based systems that communicate via lightweight API gateways for request routing and a message broker to decouple producers and consumers. Together, they form a configurable, fault-tolerant, and scalable data-processing pipeline.

TRIGGERLESS DATAFLOW IN SPD



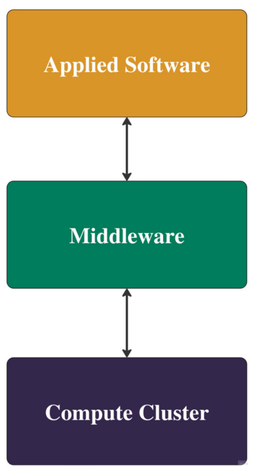
Triggerless DAQ means that the output of the system is not a set of raw events, but a set of signals from sub-detectors organized into time slices. **DAQ** provide data organized in time frames which placed in files with reasonable size (a few GB). Each of these file may be processed independently as a part of top-level workflow chain. No needs to exchange of any information during handling of each initial file, but results of may be used as input for next step of processing.

MIDDLEWARE

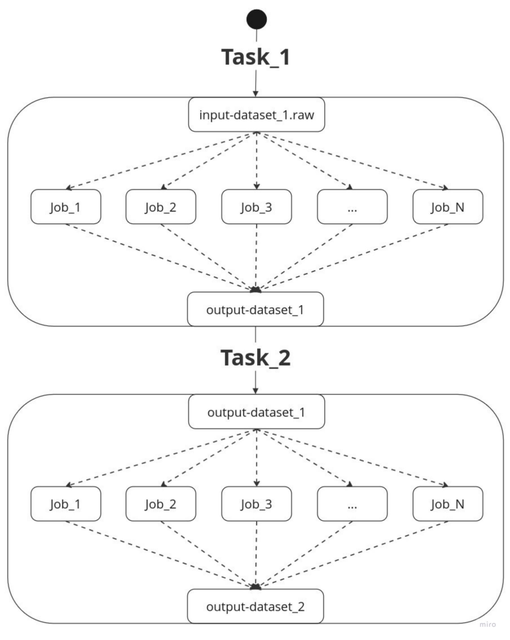
An intermediate software layer that connects hardware resources and application services. Primary purpose is to abstract the complexity of the compute cluster and provide a unified interface for application software.

Key Functions
- Data management
- Coordination of multi-stage workflows
- Efficient workload management, usage of computing resources.

Role in SPD Online Filter: bridges the dedicated compute cluster and applied software, enabling a configurable and scalable data-processing pipeline.



HIGH-THROUGHPUT COMPUTING



- **HTC** is defined as a type of computing that simultaneously executes numerous simple and computationally independent jobs to perform a data processing task.
- Since each data element can be processed simultaneously, this can be applied to data aggregated by a data acquisition system (DAQ).
- To ensure efficient utilization of computational resources, data processing should be multi-stage:
- One stage of processing → **task**
- Processing a block of data(file) → **job**

TECHNOLOGICAL STACK

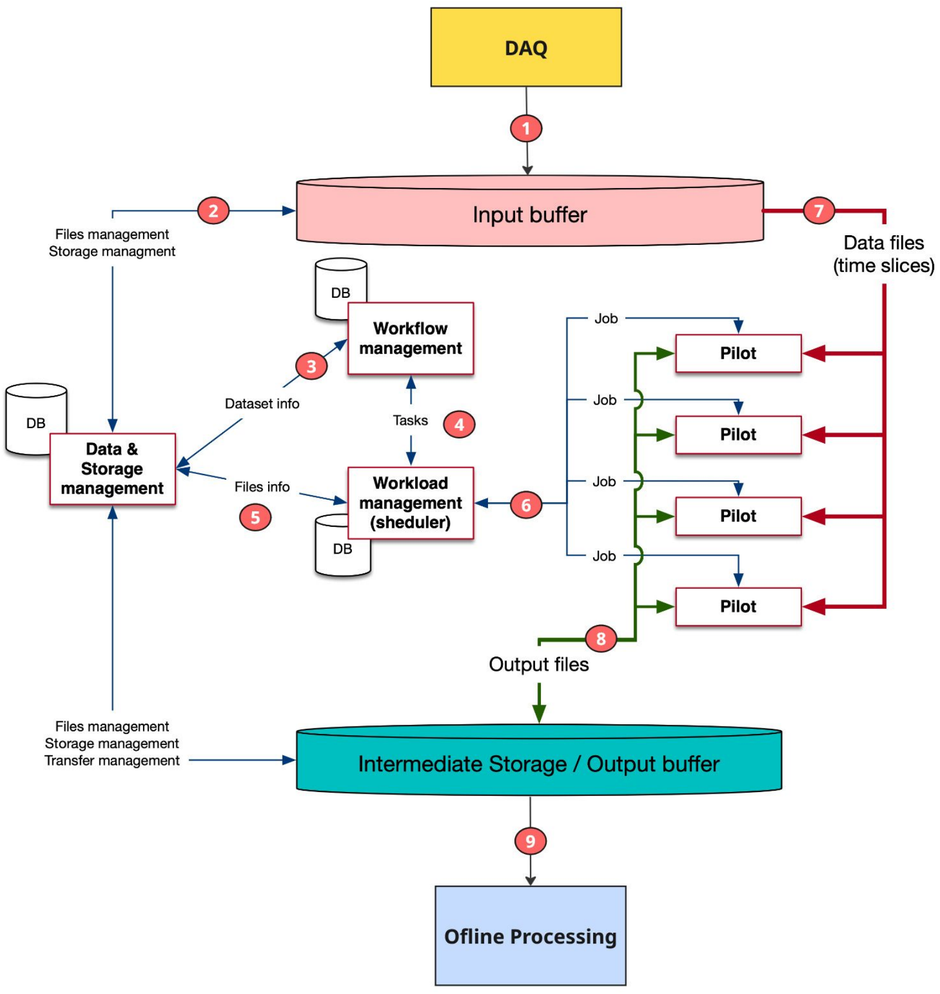
We built the SPD Online Filter middleware as a suite of containerized microservices delivered via **Docker** and deployed on our local JINR Cloud through GitLab CI/CD, ensuring consistent environments and rapid, automated rollouts.

At its core, **FastAPI** (backed by **Pydantic**) provides a high-performance, async web framework—complete with dependency-injection support—for defining clean, type-safe REST endpoints.

For persistence, we chose **PostgreSQL** paired with **SQLAlchemy 2.0**'s modern async ORM (and **asyncpg** for low-overhead database I/O), while **Alembic** handles all schema evolution through versioned migration scripts.

RabbitMQ, accessed via aio-pika, supplies reliable, brokered messaging to decouple producers and consumers across the filter pipeline. Together, this stack delivers a scalable, maintainable, and testable architecture—its clear separation of concerns, built-in data validation, and containerized delivery make it ideal for the real-time, multi-stage filtering demands of the SPD experiment.

HIGH-LEVEL ARCHITECTURE



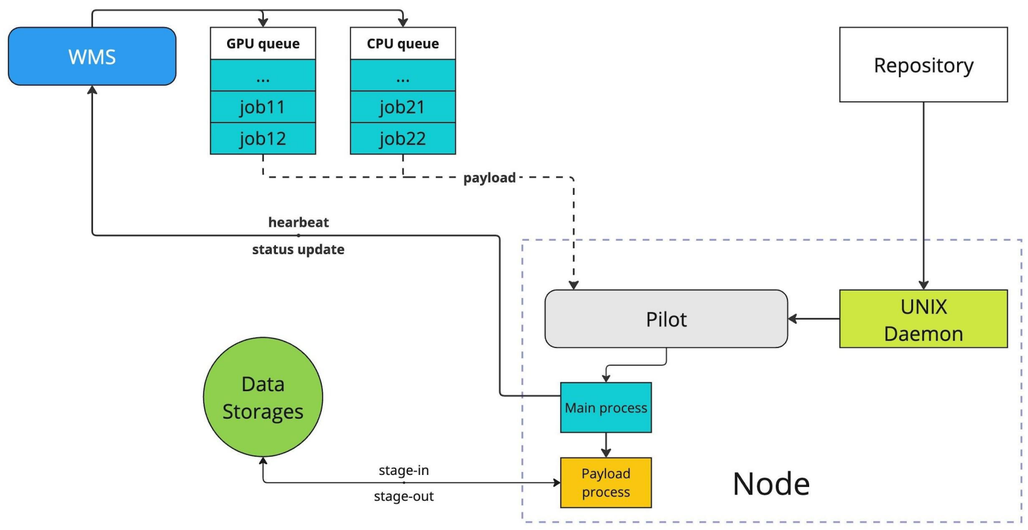
SPD Online Filter is a primary data processing facility designed for the high throughput, multi-step processing of data from the SPD detector.

Data Management System (DMS)
- Data lifecycle support (data catalog, consistency check, cleanup, storage);

Workflow Management System (WfMS)
- Define and execute data processing chains by generating the required number of computational tasks;

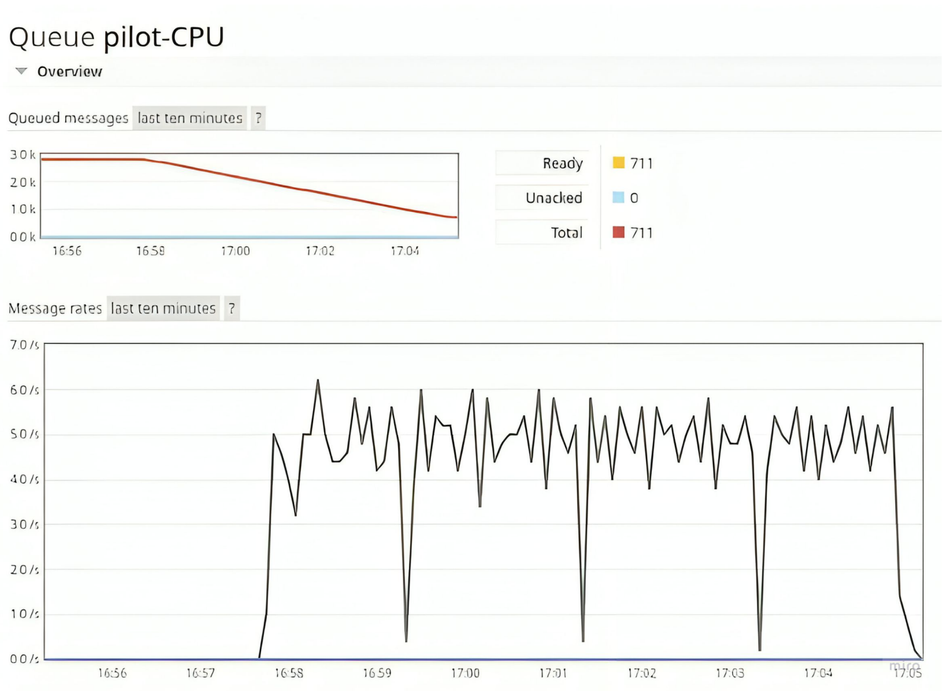
Workload management System (WMS)
- Create the required number of processing jobs to perform the task;
- Control job execution through pilots working on compute nodes;
- Handles efficient use of resources.

PILOT AGENT



Pilots are an integral part of the WMS and are responsible for executing jobs on compute nodes, organizing their execution and communicating various information about the progress and state of the WMS node to the services. Compute nodes differ only in the availability of specialized co-processors (GPUs) and are assigned to the appropriate message broker based on the computational needs of the job.

FIRST «LOAD TESTING»



In our first "load" test, 100 concurrent Pilot agents processed approximately 2,100 jobs in 7 minutes (≈15 s per job, including stage-in and stage-out) on standard JINR Cloud VMs using a simplified synthetic payload.

MESHCHERYAKOV LABORATORY OF INFORMATION TECHNOLOGIES