

# Evolution of the Data Quality Monitoring and Prompt Processing System in the protoDUNE-SP experiment



M.Potekhin  
Brookhaven National Laboratory  
for the DUNE Collaboration

## ProtoDUNE-SP

ProtoDUNE-SP is a large-scale prototype of the Single-Phase variant of the Liquid Argon Time Projection Chamber constructed by the DUNE collaboration.

- Full-size Anode and Cathode Plane Assemblies, with “Cold Electronics” located inside the cryostat, TPC channel count: 15,360
- Created in the extension of the CERN North Area Hall, with a purpose-built tertiary beam from the SPS, commissioned during Summer’18
- The experiment successfully took test-beam data in the Fall of 2018, and data taking continues in 2019 with an extended Cosmic Ray run
- Another test-beam run planned in 2021
- Collected more than 3PB of raw data

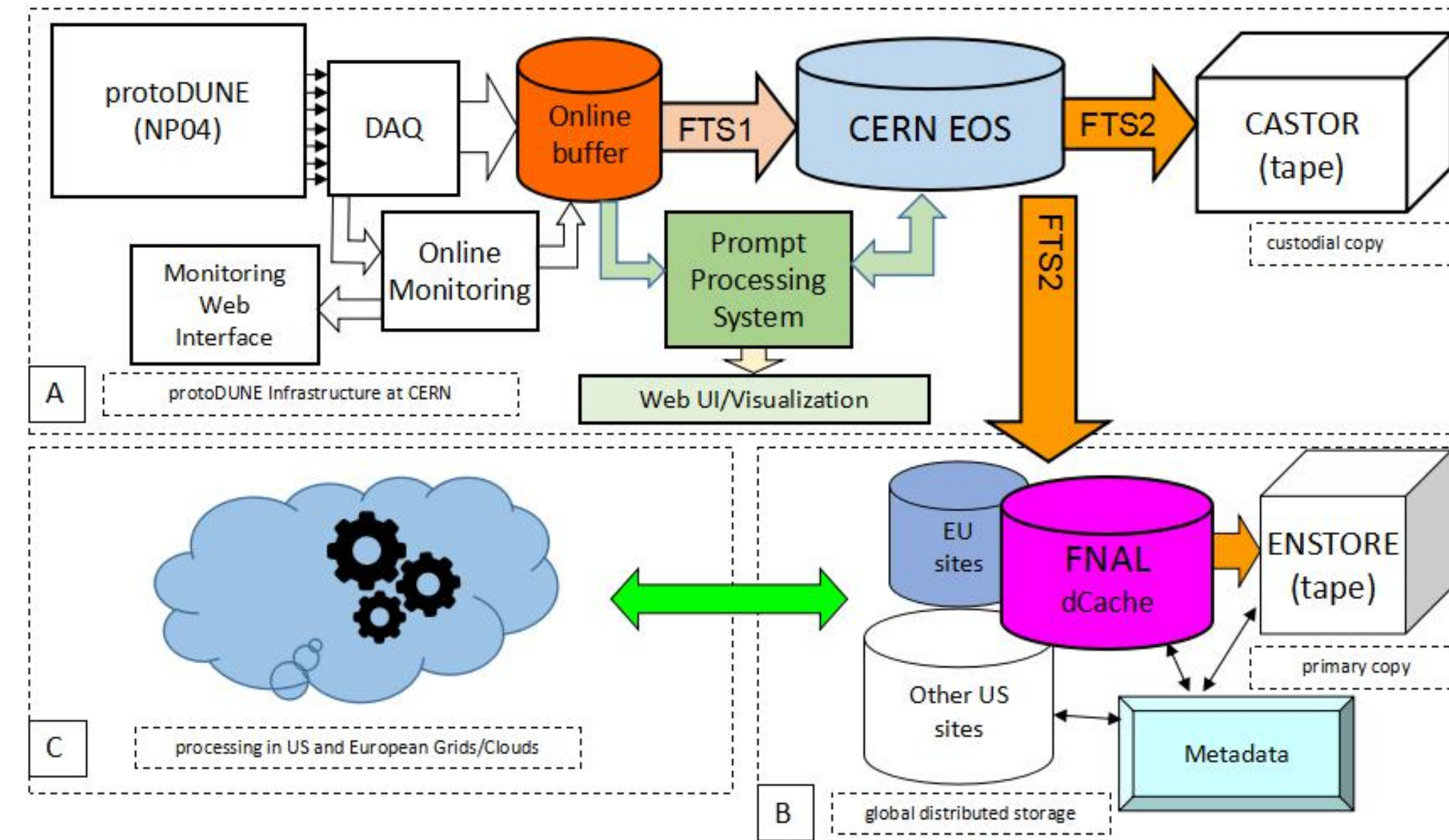


## Data Quality Monitoring

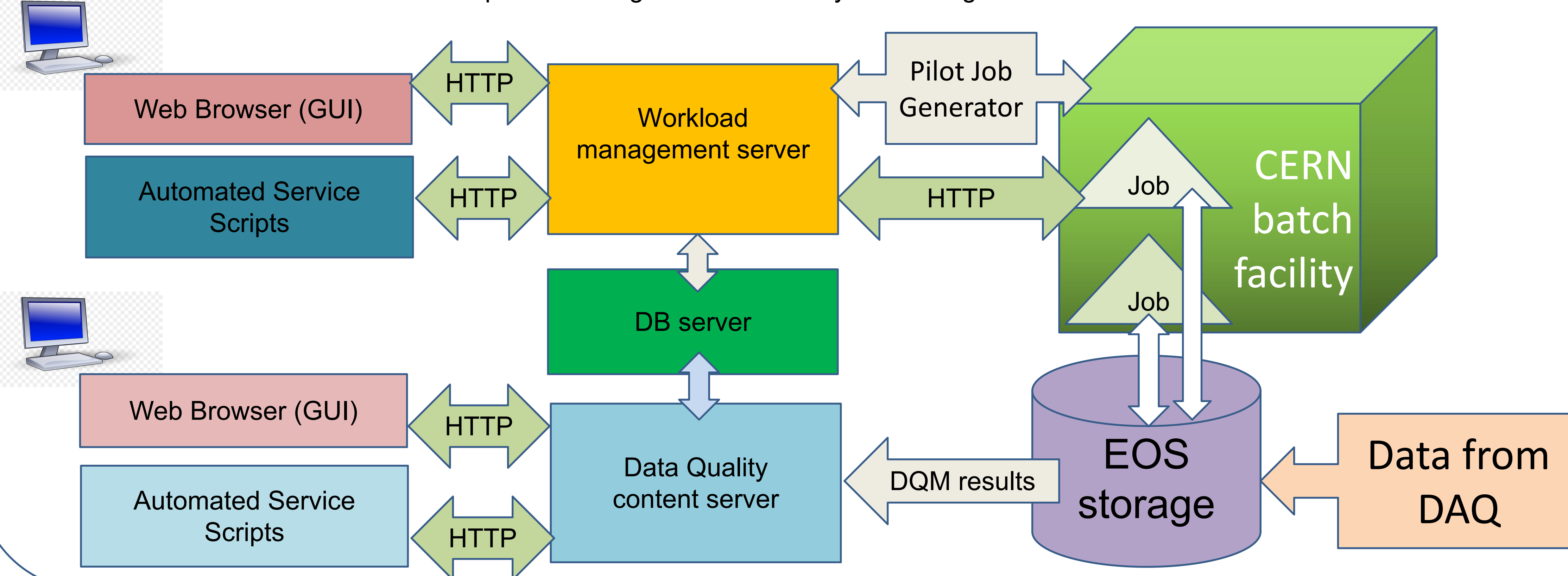
ProtoDUNE-SP needs a capable Data Quality Monitoring (DQM) system for commissioning and ongoing operation. Its functionality is complementary to the DAQ and implemented separately for the following reasons:

- Most of the DQM applications (e.g. the ones requiring signal processing, filtering, FFT etc) have longer execution times than typical functions implemented in the Online Monitor of the Data Acquisition System in ProtoDUNE-SP and may require more CPU
- DQM requires cataloging and storage of a variety of data products it creates, which is not in the scope of the DAQ
- DQM needs agility in making quick adjustments to the application code and introducing new applications as necessary, which is not compatible with the requirement that the DAQ software must be quite stable, reliable and constantly operational
- Some components of DQM require information “out-of-band” i.e. from sources outside of the data streams available in DAQ

## ProtoDUNE-SP data flow



## Prompt Processing and Data Quality Monitoring in ProtoDUNE-SP



## Software Stack (services)

- Python 3.+
- Django 2.+
- Django tables 2
- Apache 2.4
- PostgreSQL
- Google Charts

## Hosting (services)

- CERN OpenStack VMs
- CENTOS 7
- Snapshots - for recovery as well as scaling
- 2 instances of Apache + 1 spare
- 1 instance of PostgreSQL

## Evolving DQM software

During the commissioning phase and throughout operations the DQM applications were added, modified and removed from the processing scheme according to the needs of the experiment. At various points in time the following payloads were run:

- 2D event display based on raw data
- The TPC monitoring application with an extensive set of histograms at various levels of channel aggregation, FFT charts etc
- Health monitor for the Front End Electronics motherboards
- Prompt reconstruction with metrics such as track candidate count, hit count etc
- Track candidate based Argon purity estimations with time series stored in the DB
- Signal-to-noise ratio monitoring
- Data preparation for the 3D event display implemented on a separate server
- Out-of-band feed from the hardware Argon purity monitor

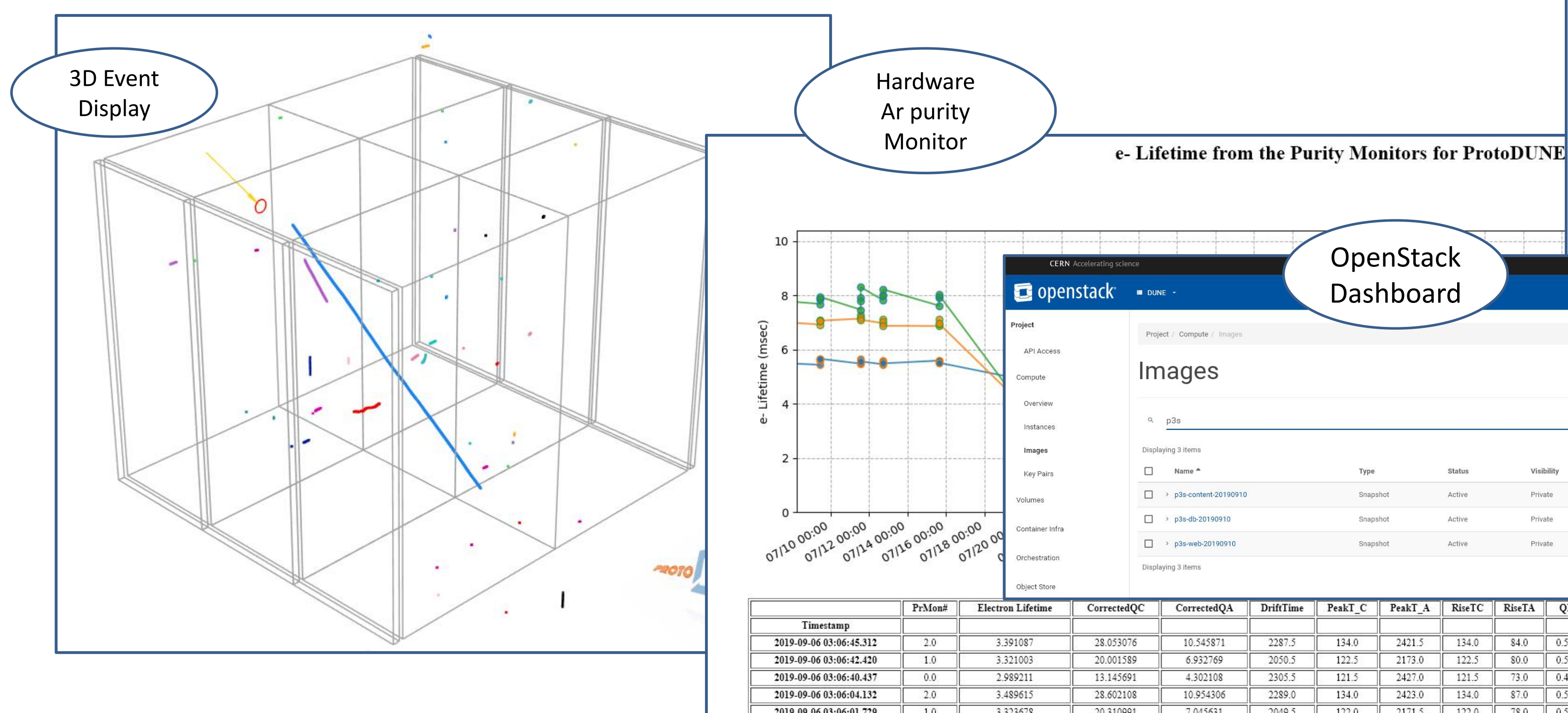
The core DQM application platform (LArSoft) was periodically updated and the software provisioning was done via CVMFS. Additions and modifications of the DQM payloads were facilitated by metadata generated by applications which was used to automatically generate menus and layouts for graphics on Web pages, without changes on the server side.

## Evolving hardware

The system is using a pilot-based approach to workload management, inspired by PanDA and Dirac. This made it easy to deploy the system first on a small ad hoc cluster, then migrate it to hardware located in the experiment's data room, and finally to OpenStack (services) and the central batch facility at CERN (worker nodes). The OpenStack instances were later modified for better scalability.

## Evolving monitoring

Monitoring tools were added to the prompt processing system to address a few infrastructure issues encountered during its operation, e.g. collecting metrics for file transfer latency, batch system performance, storage required by the input data and displaying alarms to operators as necessary.



FFT