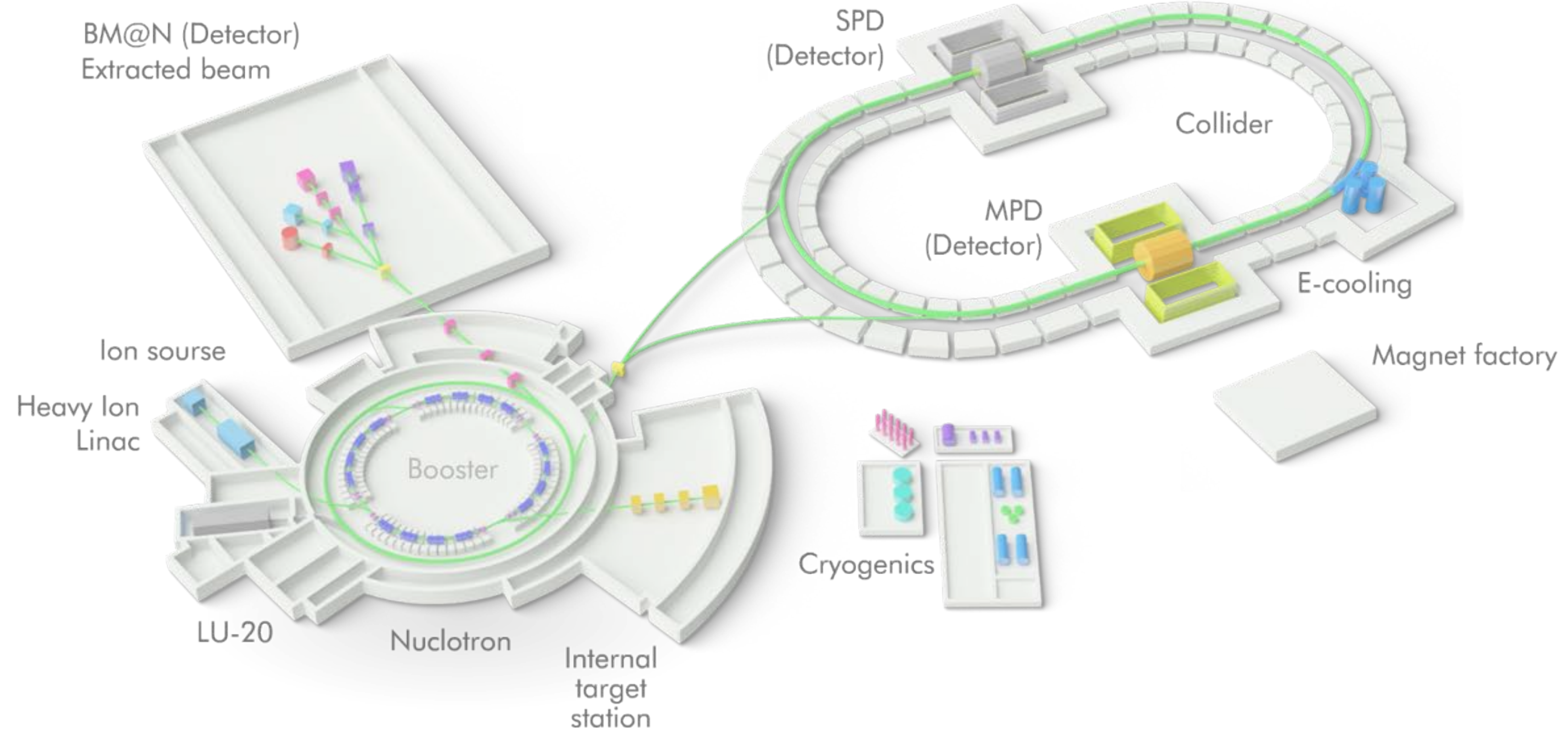
# SPD Online filter

## Первичная обработка экспериментальных данных эксперимента SPD

**Oleynik D. JINR LIT**

# What is HEP?

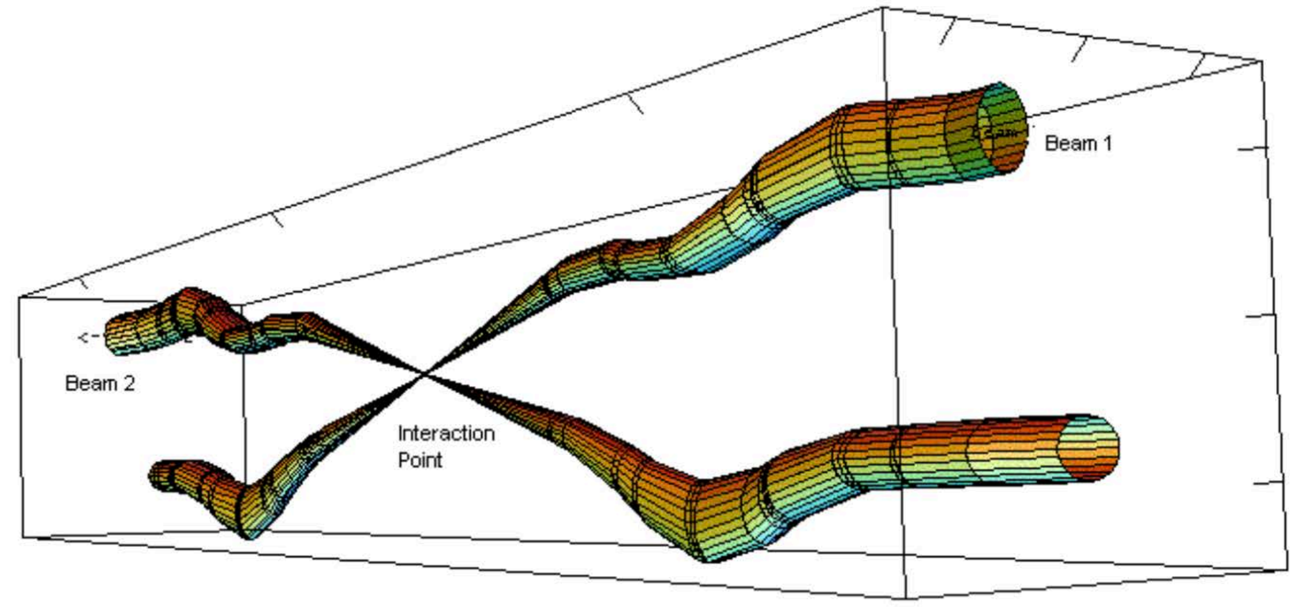
## HEP - High Energy Physics



- **Particle physics** or **high energy** physics is the study of fundamental particles and forces that constitute matter and radiation (Wikipedia)
- A **particle accelerator** is a machine that uses electromagnetic fields to propel charged particles to very high speeds and energies, and to contain them in well-defined beams
- A **particle detector**, also known as a **radiation detector**, is a device used to detect, **track**, and/or identify ionizing particles. Detectors can measure the particle energy and other attributes such as momentum, spin, charge, particle type, in addition to merely registering the presence of the particle.

# Data granularity

## What is “Event”?



Relative beam sizes around IP1 (Atlas) in collision

*In physics, and in particular relativity, an event is the instantaneous physical situation or occurrence associated with a point in spacetime (wikipedia).*

For HEP:

- **Interaction:** Two particles interact and somehow produce a change, be it in energy, trajectory or identity.
- **Collision:** Two particles are made to approach one another and actually undergo an interaction.
- **Beam crossing:** Two beam bunches pass through one another in the center of a detector.
- **Event:** During a beam crossing, one pair or multiple pairs of particles undergo a collision. In an event, often one collision dominates the signature in the detector.

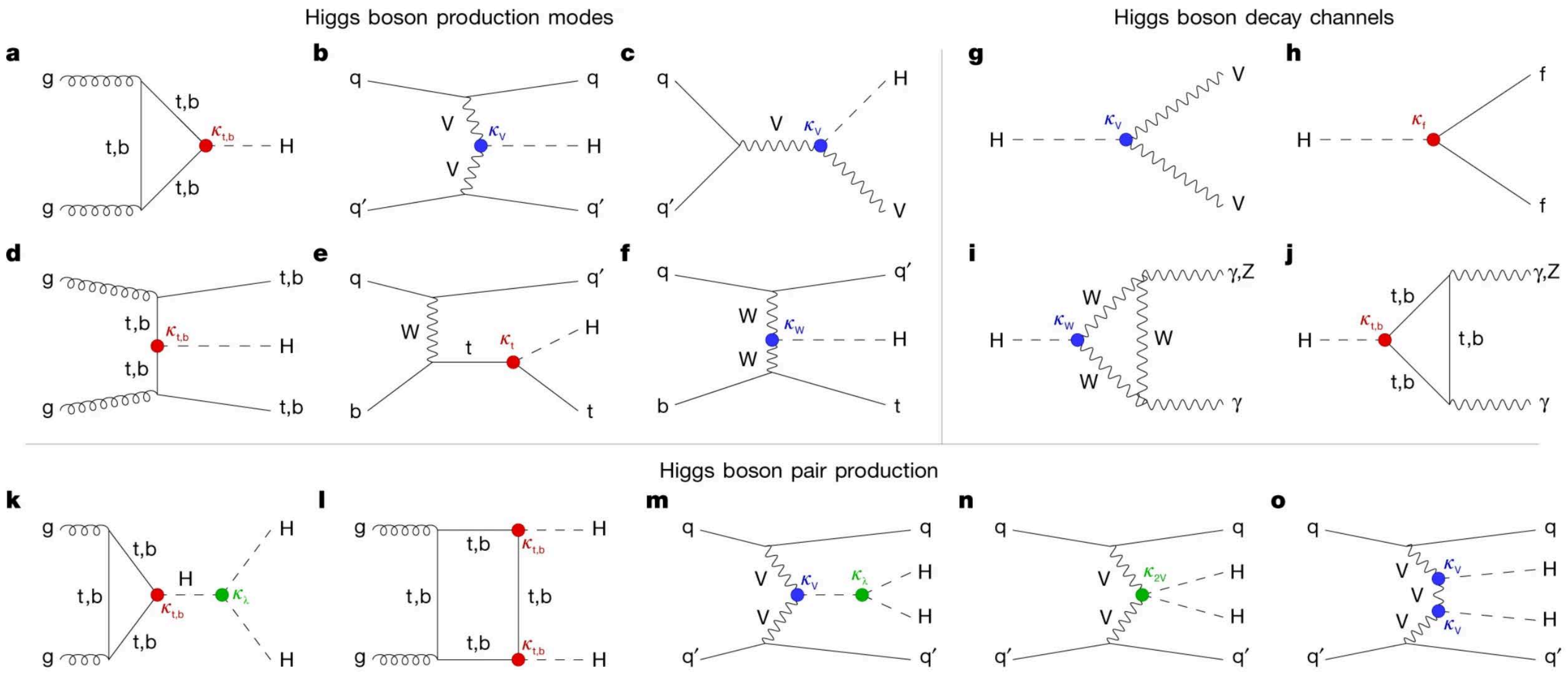
Each event may be processed independently, so for HEP computing - event is the least data unit.

By knowing the size of event, complicity of event processing and expected number of event for processing, you can estimate requirements for computing system.

# Why we care about “Event rate”

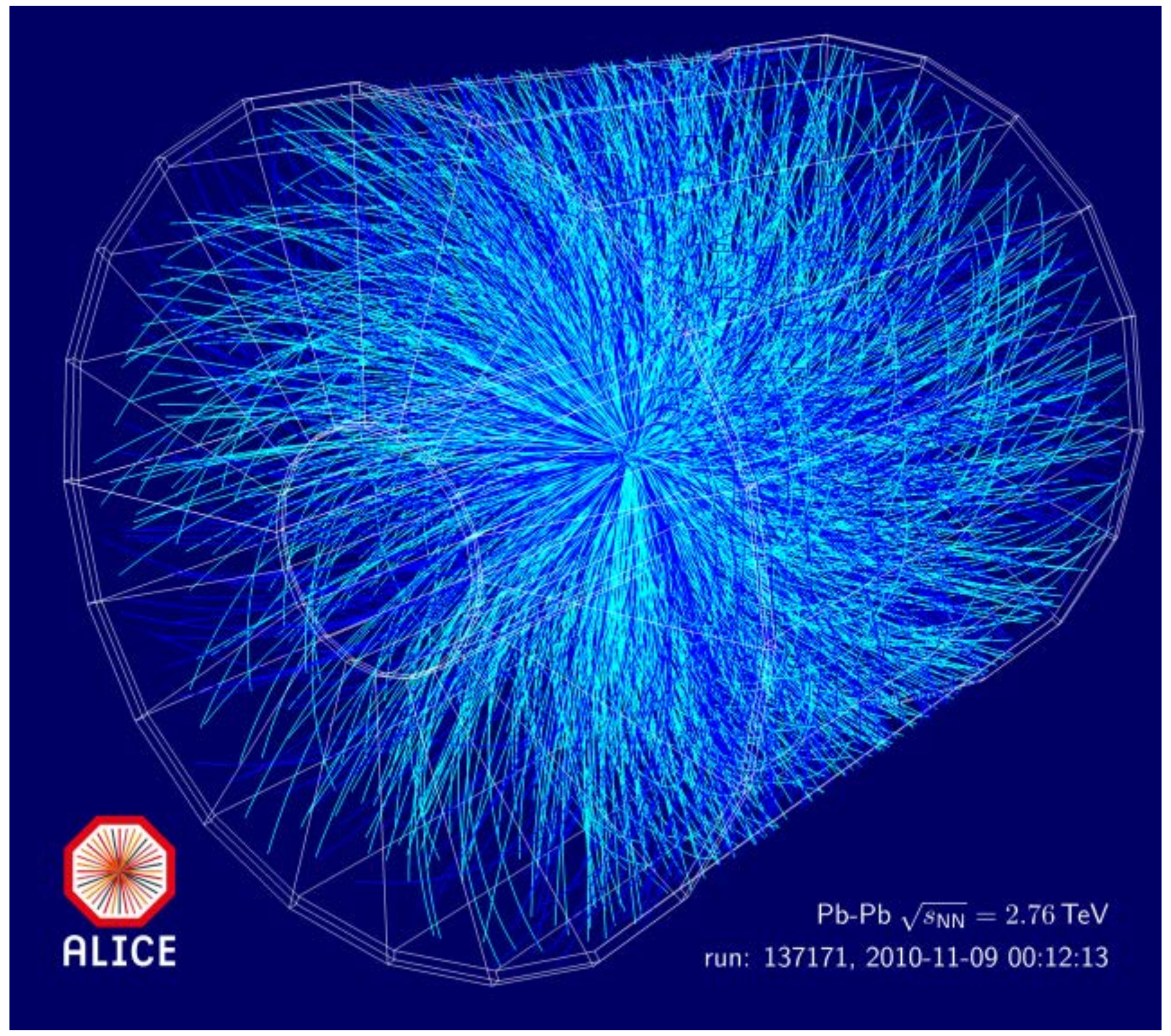
## Higgs boson example

- *Frequency of producing a Higgs boson that has decayed to two Z bosons each of which has decayed to an electron-positron pair is extremely rare: once in $10^{13}$ = 10 000 000 000 000*



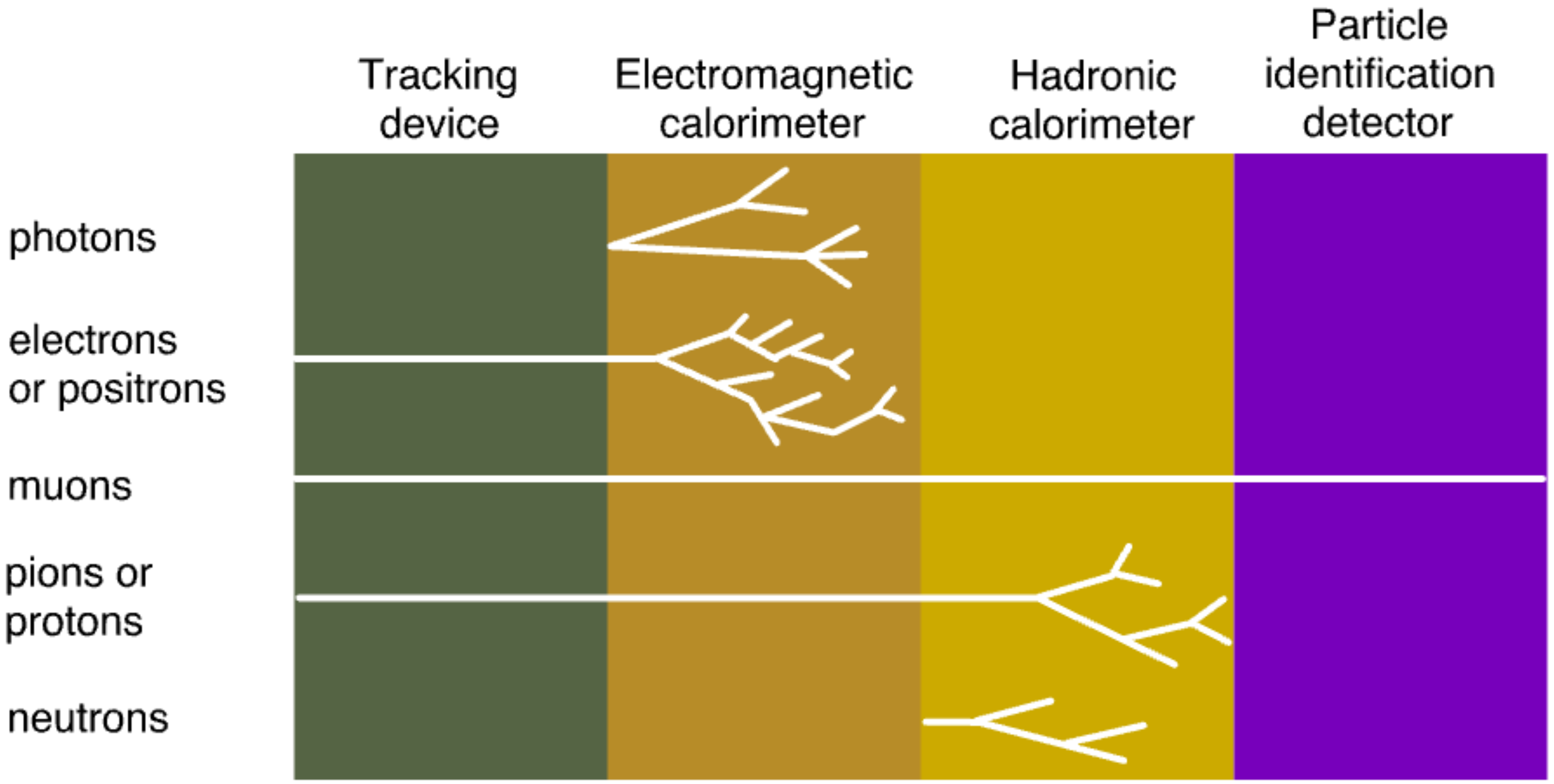
# How to estimate expected data volume

- Accelerator parameters:  luminosity
  - NICA:  $10^{27}$ $cm^2 \cdot s^{-1}$ (nucleon-nucleon) – $10^{34}$ $cm^2 \cdot s^{-1}$ (proton-deuteron)
- Expected efficiency of detector and DAQ: how many signals and how often they and be collected, how many events can be stored


ALICE
Pb-Pb $\sqrt{s_{NN}}$ = 2.76 TeV
run: 137171, 2010-11-09 00:12:13

| Experiment | Production rate (event/sec) | Raw event size (KB) |
|---|---:|---:|
| **SPD** | 150 000 | 50 |
| **MPD** | 7 000 | 1500 |
| **BM&N** | 5 000 | 500 |

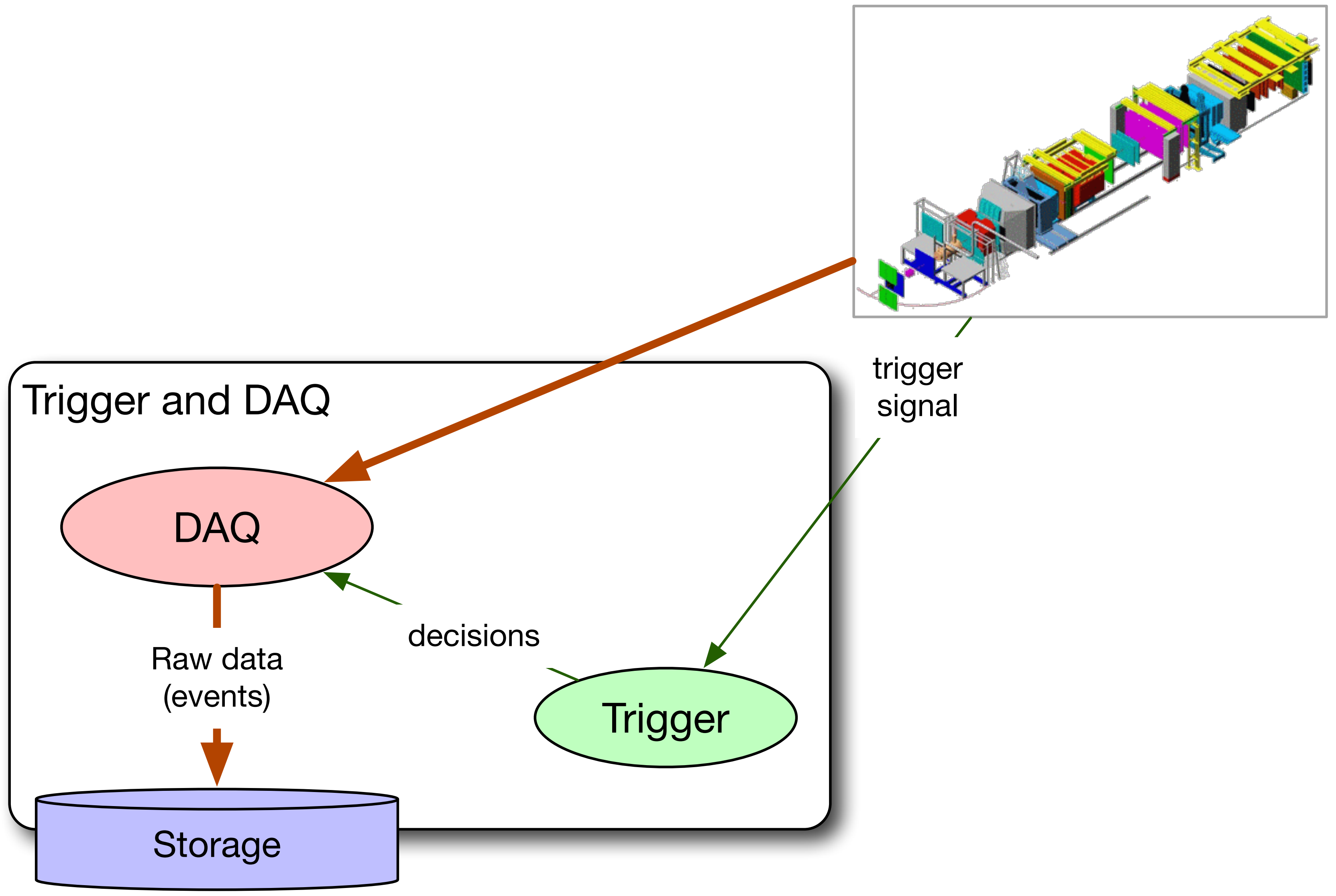
# Generic HEP detector



- **Tracking device** – detects and reveals the path of a particle
- **Calorimeter** – stops, absorbs and measures the energy of a particle
- **Particle identification detector** – identifies the type of particle using various techniques

# Data Acquisition System

## What is trigger?



Trigger and DAQ

DAQ

decisions

Raw data (events)

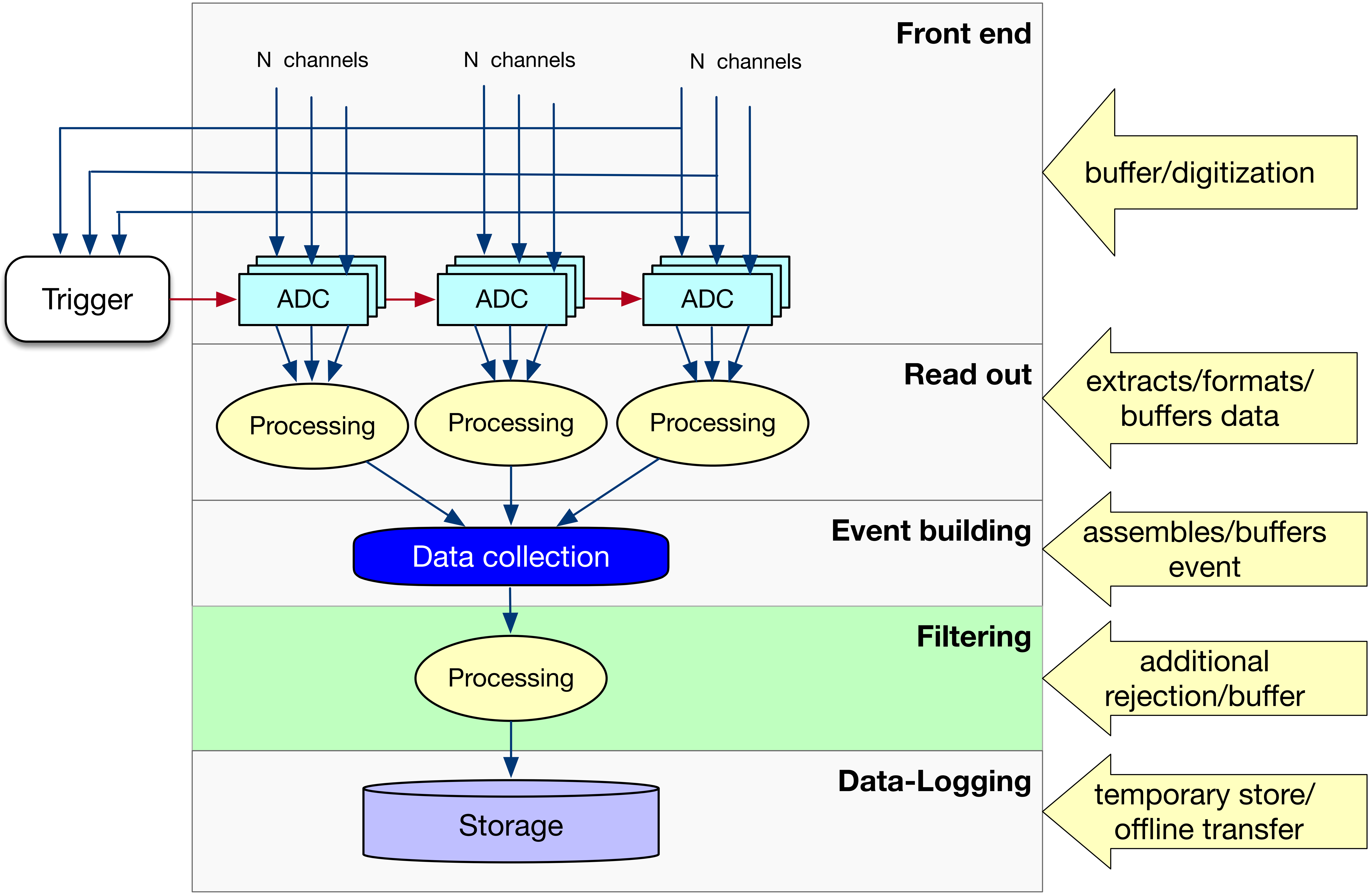
Trigger

Storage

trigger signal

- Main role of Trigger & Data acquisition (DAQ):
  - process the signals generated in the detectors
  - Select the ‘interesting’ events and reject the ‘boring’ ones
  - save interesting ones on mass storage for future processing or analysis
- Trigger, in general, something which tells you when is the “right” moment to take your data
- Trigger – process to very rapidly decide if you want to keep the data if you can’t keep all of them. The decision is based on some ‘simple’ criteria

# “Classical" DAQ



Front end

N channels

N channels

N channels

Trigger

ADC

ADC

ADC

Read out

Processing

Processing

Processing

Event building

Data collection

Filtering

Processing

Data-Logging

Storage

buffer/digitization

extracts/formats/ buffers data

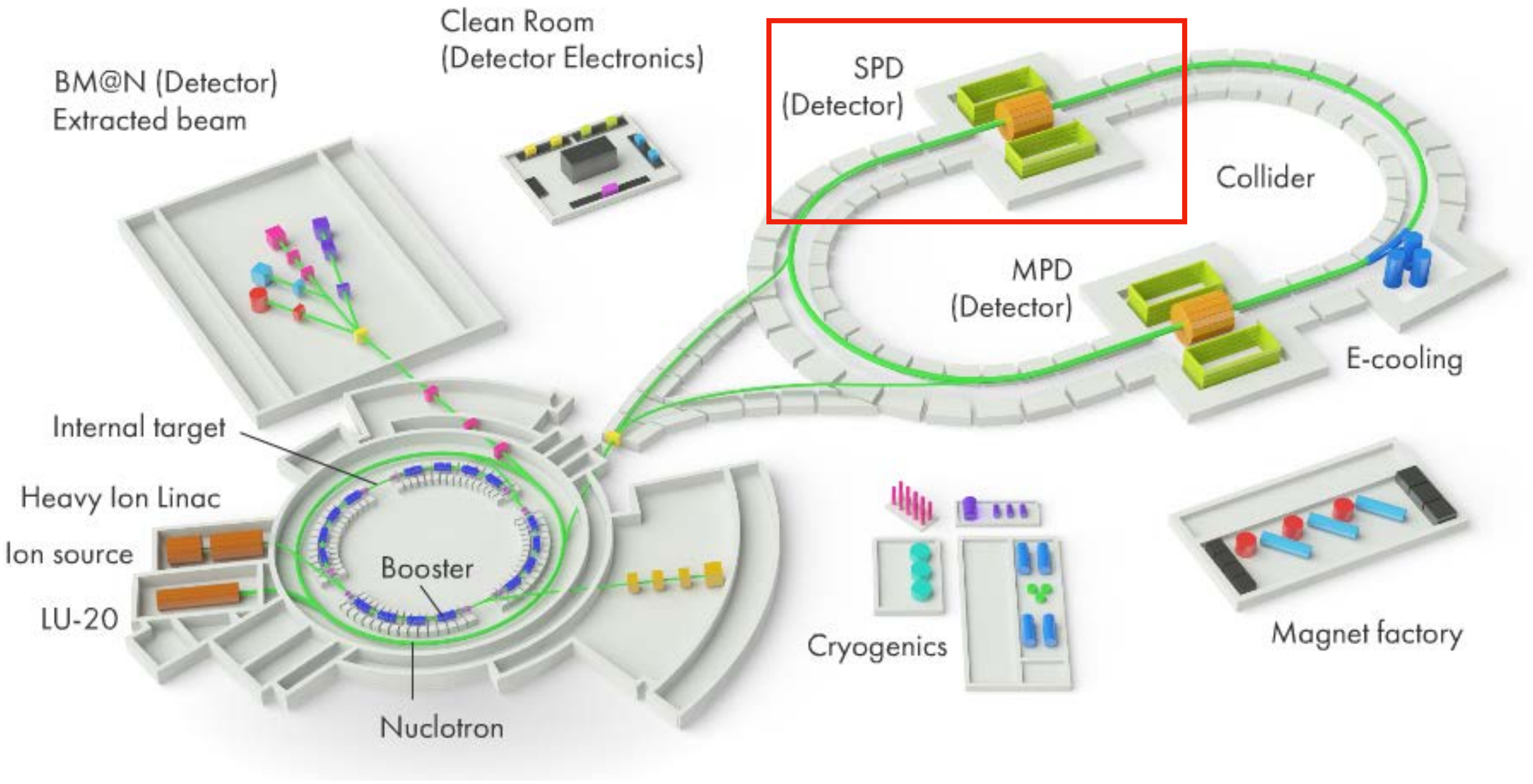
assembles/buffers event

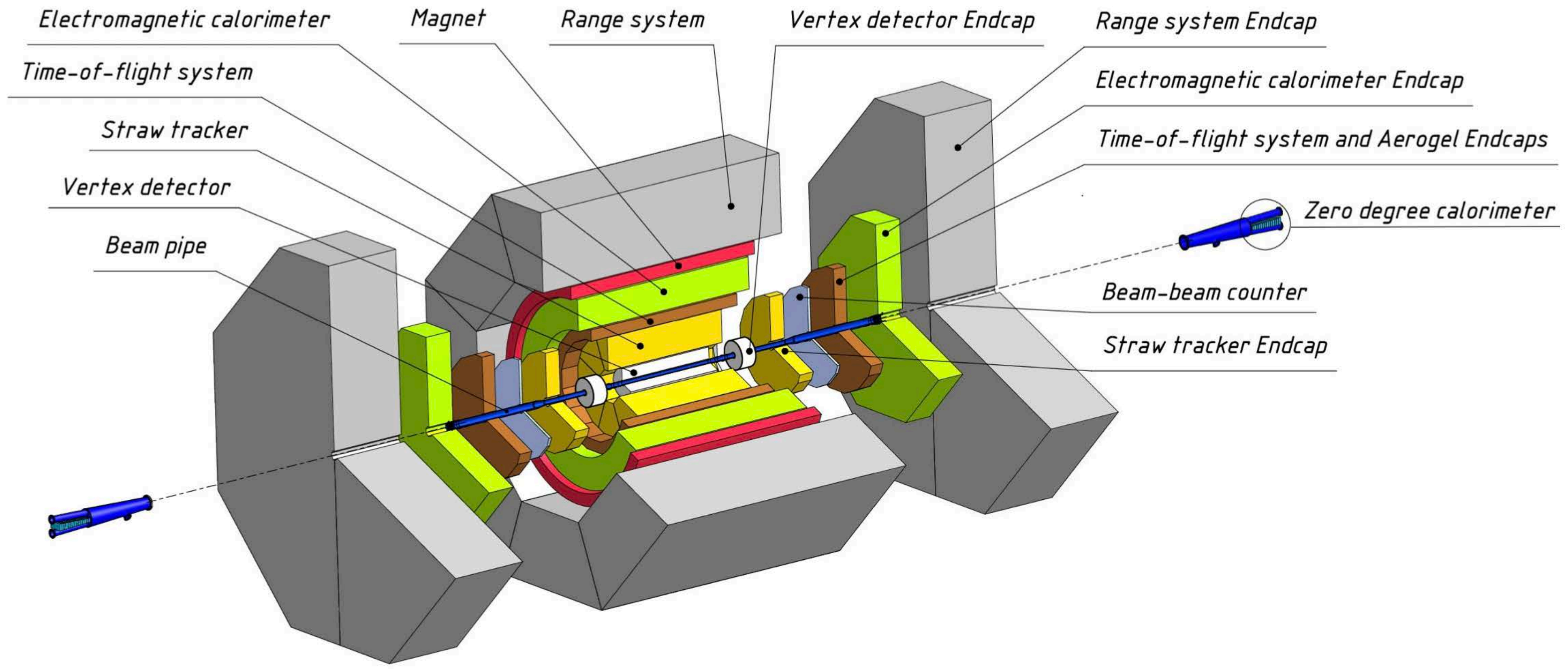
additional rejection/buffer

temporary store/ offline transfer

# SPD Spin Physics Detector

Study of the nucleon spin structure and spin-related phenomena in polarized *p*-*p*, *d*-*d* and *p*-*d* collisions





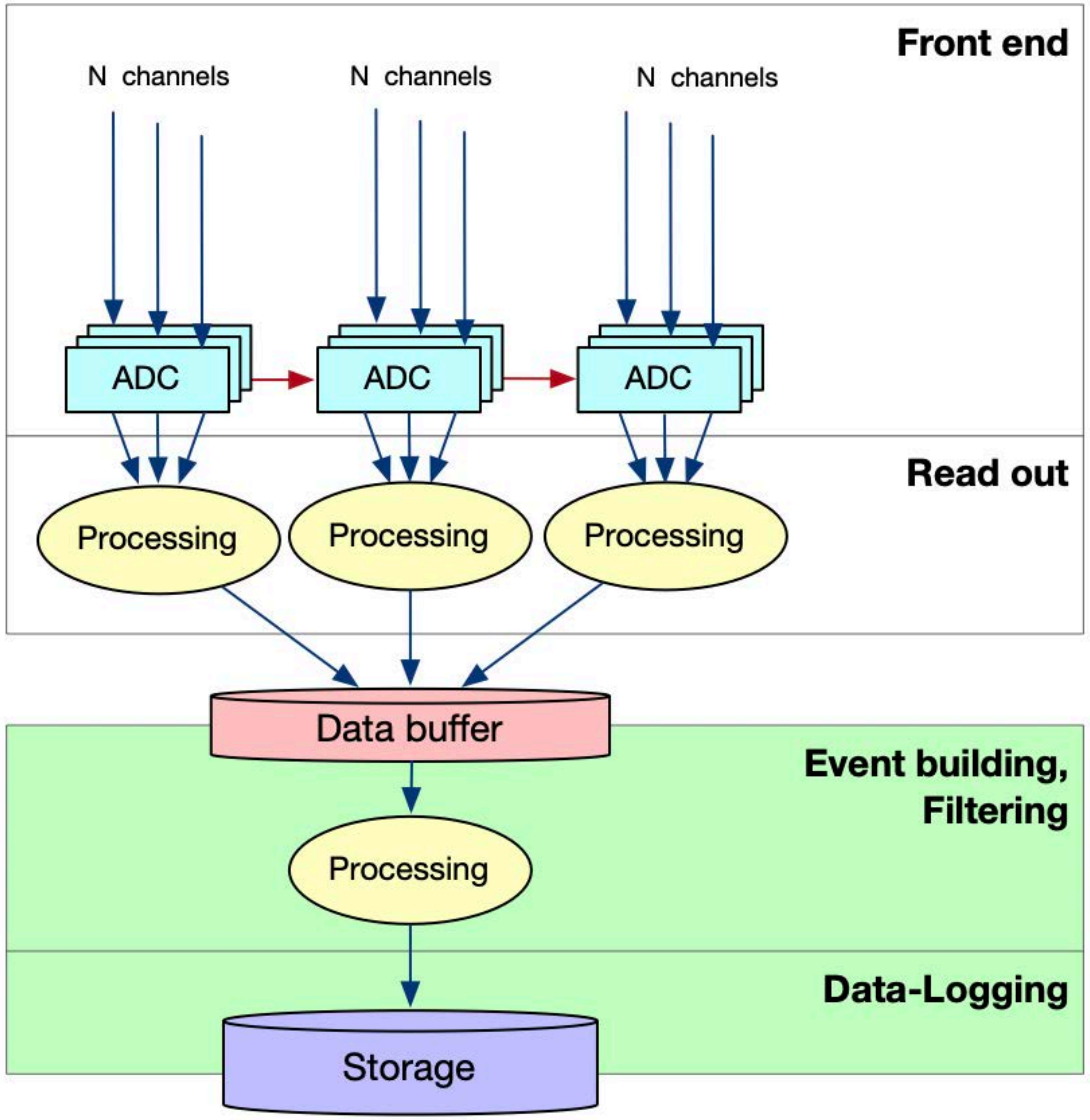
SPD - a universal facility for comprehensive study of gluon content in proton and deuteron

# SPD detector as data source

- Bunch crossing every 80 ns = crossing rate 12.5 MHz
    - ~ 3 MHz event rate (at $10^{32}$ $cm^{-2}s^{-1}$ design luminosity) = **pileups**
- **20 GB/s** (or **200 PB/year** "raw" data, **~$3*10^{13}$** events/year)
    - Selection of physics signal requires momentum and vertex reconstruction → no **simple trigger** is possible

# Trigerless DAQ

- Triggerless DAQ, means that the output of the system will not be a dataset of raw events, but a set of signals from sub-detectors organized in time slices
- To get data in proper format for future processing (reconstruction) and filtering of ‘boring’ events special computing facility named "Online Filter" in progress



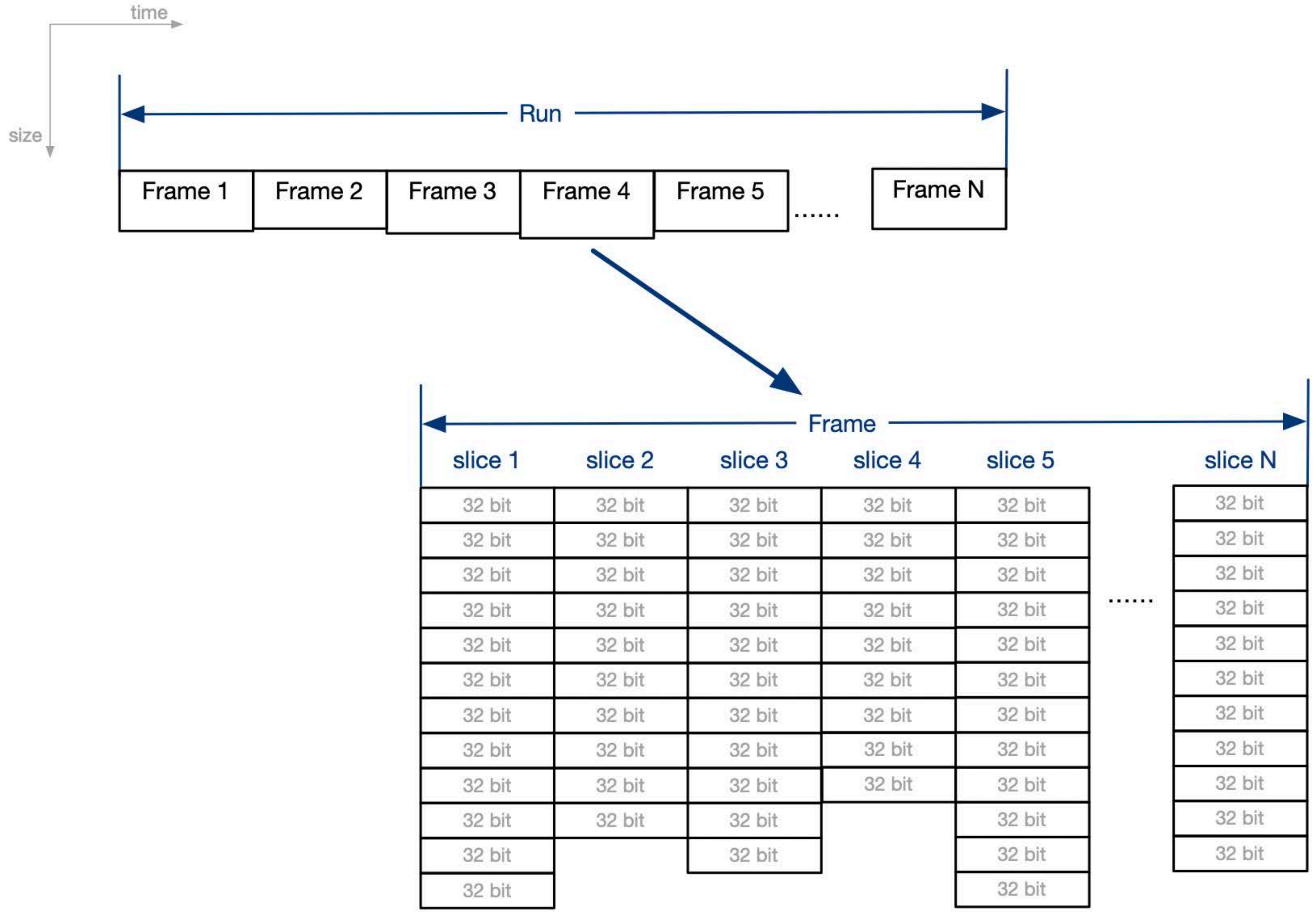
# HTC and HPC

- HPC tasks are characterized as needing large amounts of computing power for short periods of time, whereas HTC tasks also require large amounts of computing, but for much longer times (months and years, rather than hours and days). HPC environments are often measured in terms of FLOPS.
- The HTC, however, is not concerned about operations per second, but rather operations per month or per year. Therefore, the HTC field is more interested in how many jobs can be completed over a long period of time instead of how fast an individual job can complete.
    - As an alternative definition, the European Grid Infrastructure defines HTC as “a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks”, while HPC systems tend to focus on tightly coupled parallel jobs, and as such they must execute within a particular site with low-latency interconnects. Conversely, HTC systems are independent, sequential jobs that can be individually scheduled on many different computing resources.

# High-throughput computing for SPD data processing

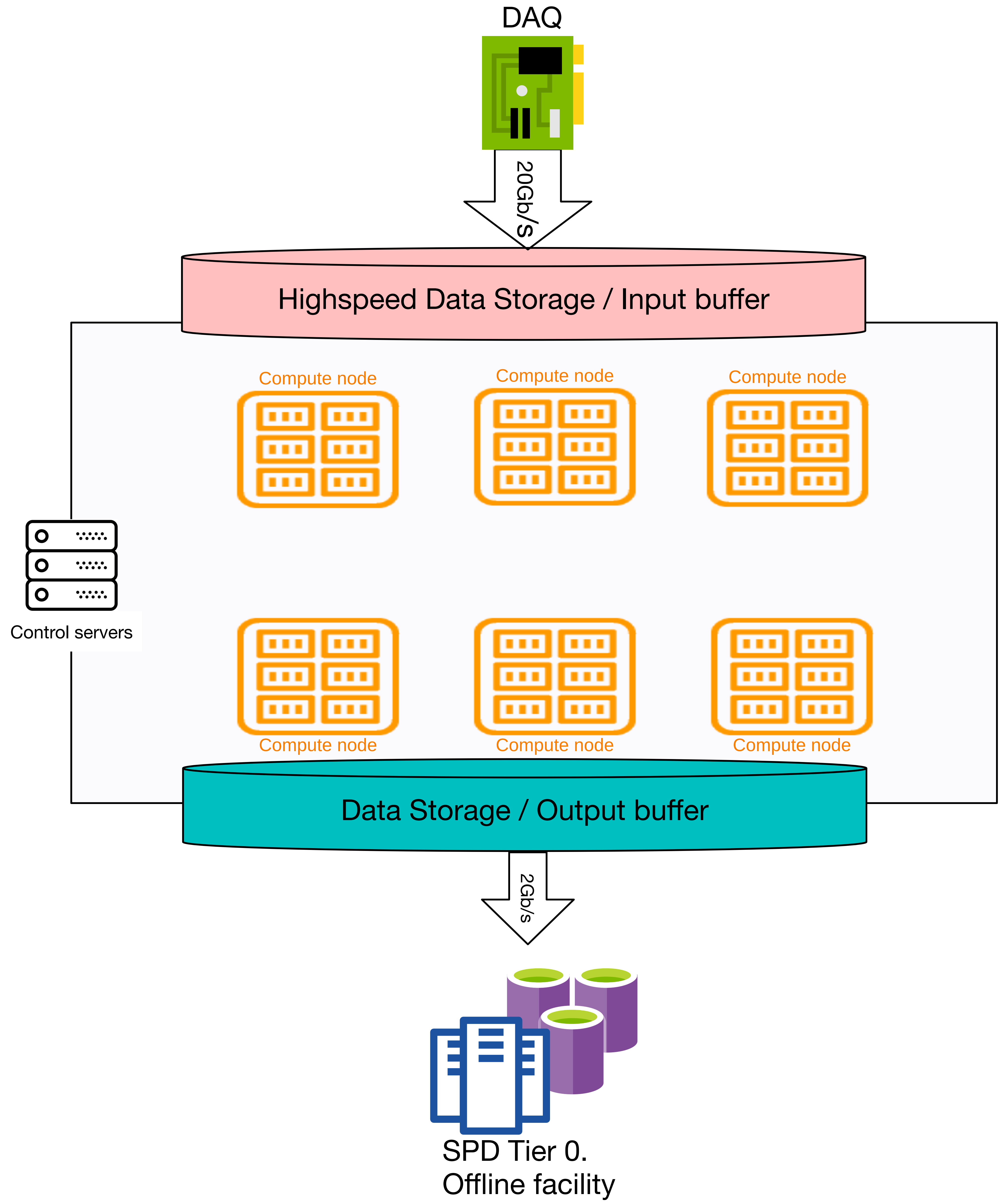
*High-throughput computing (HTC) involves running many independent tasks that require a large amount of computing power.*

- DAQ provide data organized in time frames and sliced to 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.



# Online filter

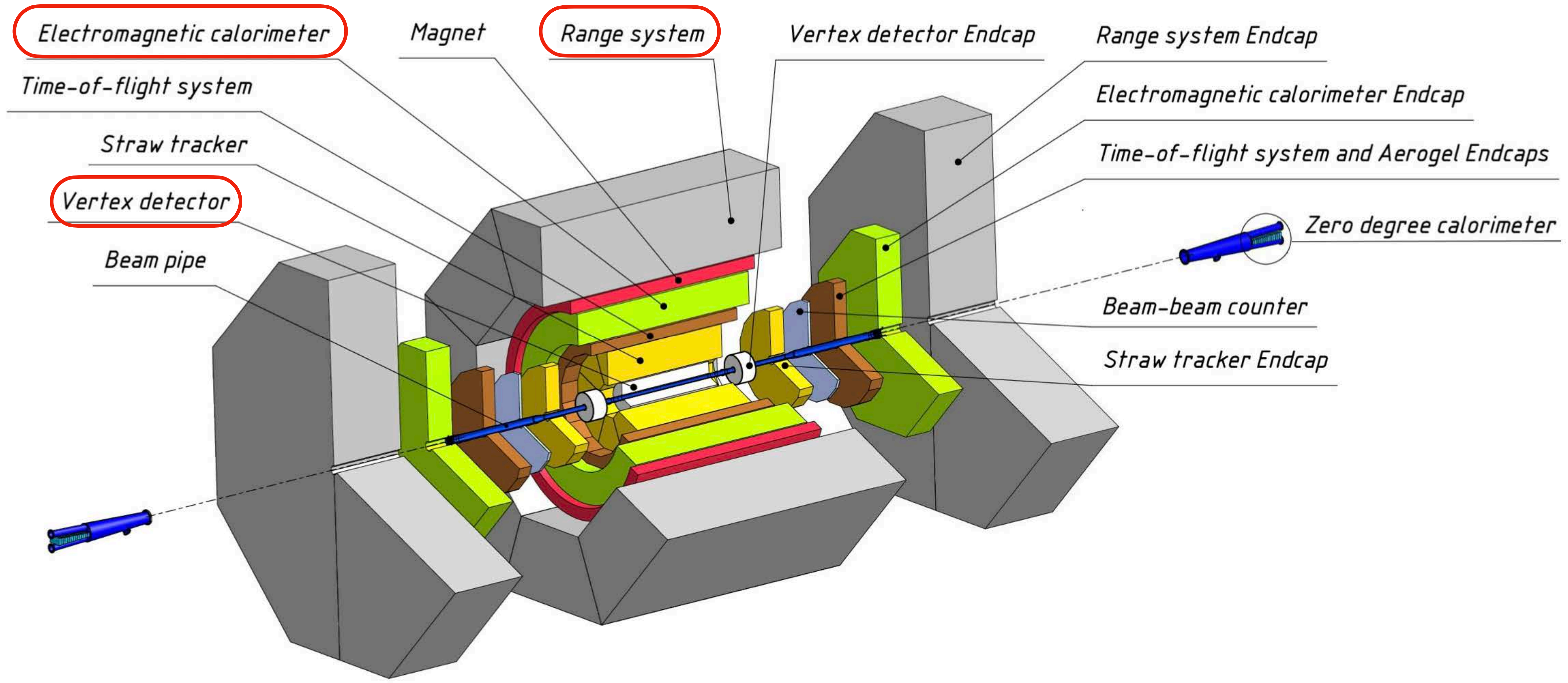
- SPD Online Filter is a high performance computing system for high throughput  processing
- This computing system should carry out next transformation of data: identify physics events in time slices; reorganize data (hits) in event's oriented format; filter 'boring' events and leave only 'hot'; settle output data, merge events into files and files in datasets for future processing



# Event unscrambling

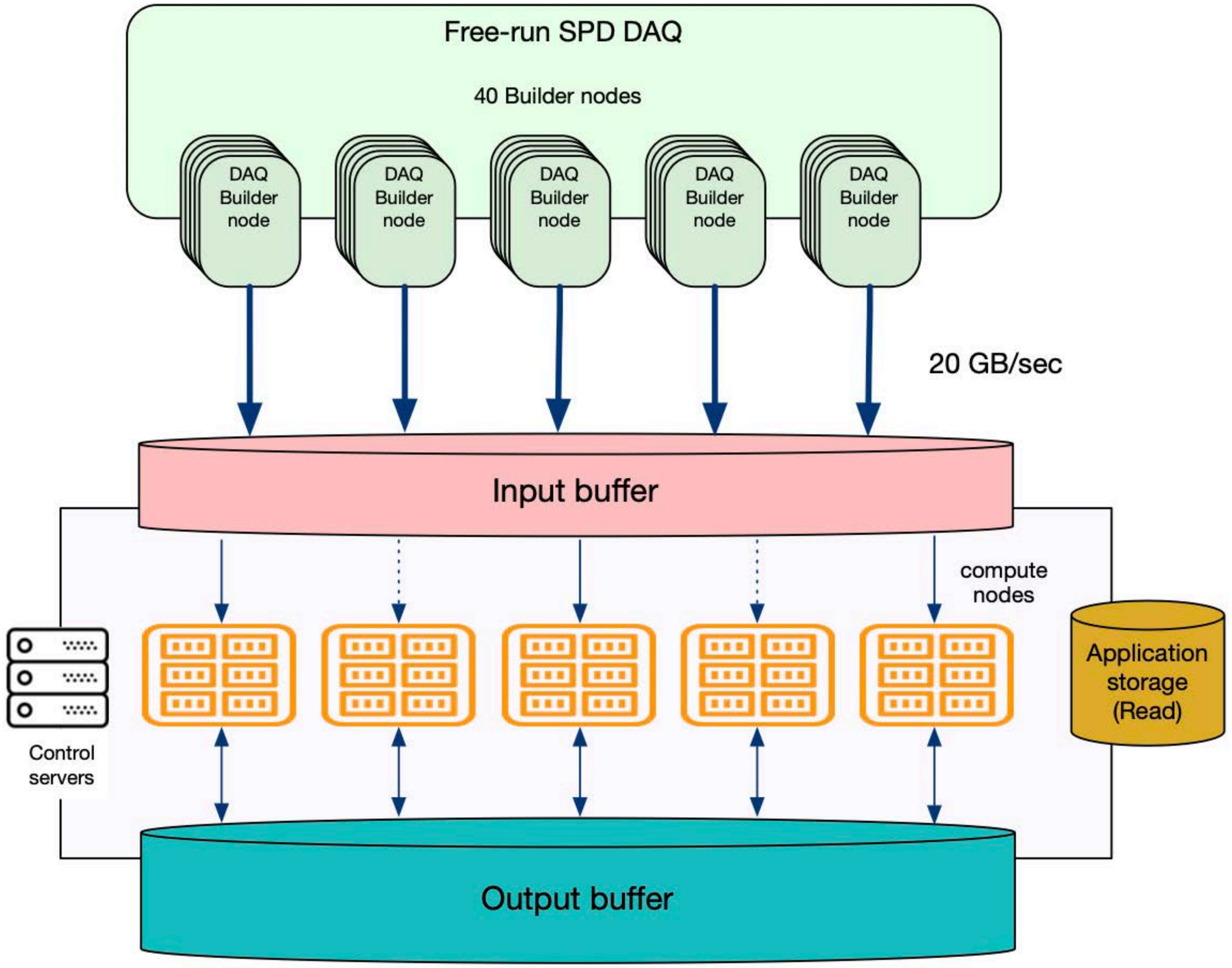
For each time slice

- Reconstruct tracks and associate them with vertices
- Determine bunch crossing time for each vertex
- Associate ECAL and RS hits with each vertex (by timestamp)
- Attach unassociated tracker hits in a selected time window according to bunch crossing time
- Attach raw data from other subdetectors according to bunch crossing time
- Call the block of information associated with each vertex an event
- Store reconstructed events

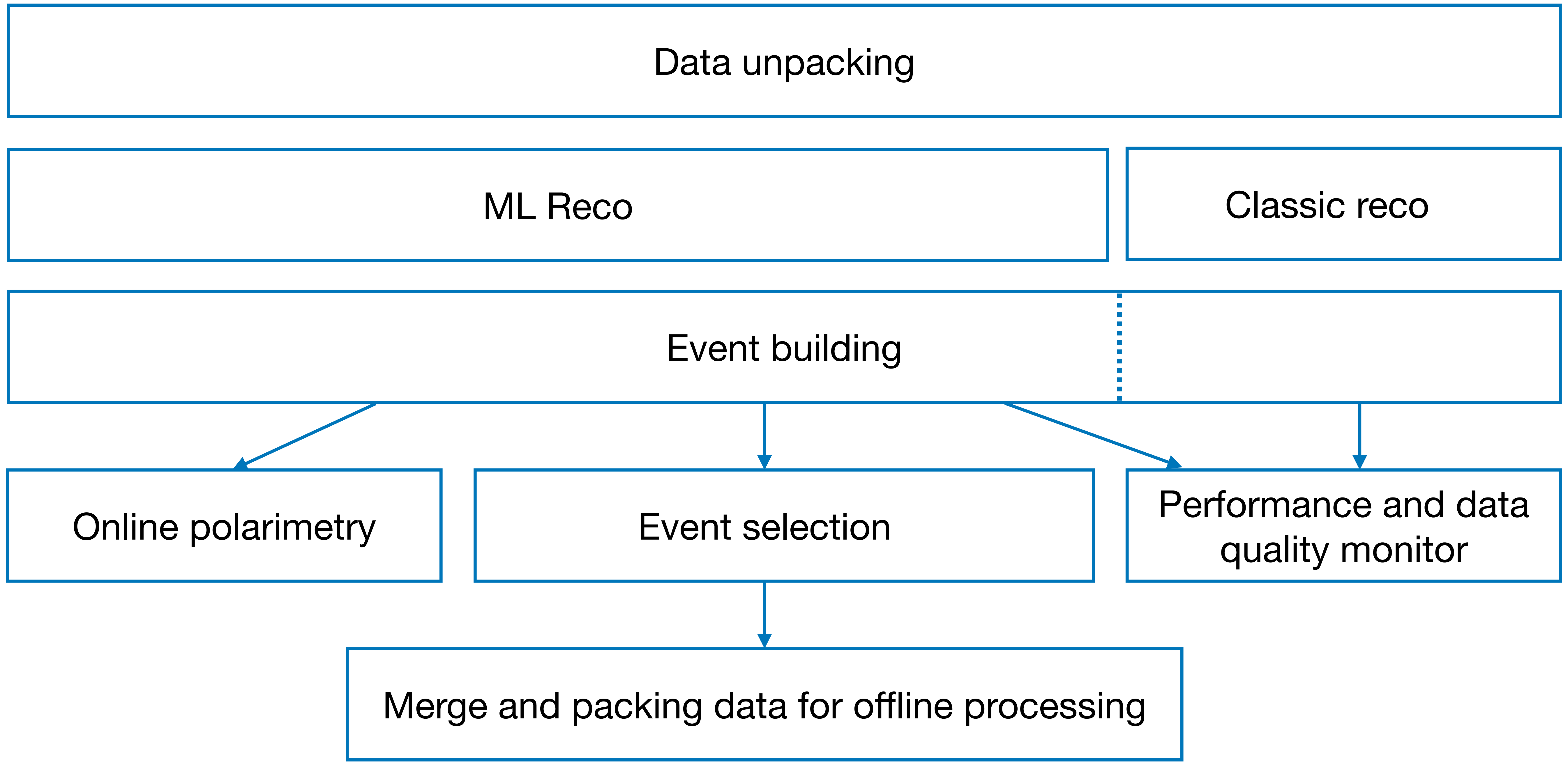


# Online filter infrastructure

- SPD Online Filter is a high performance computing system for high throughput processing
  - High speed (parallel) storage system for input data written by DAQ.
  - Compute cluster with two types of units: multi-CPU and hybrid multi CPU + Neural network accelerators (GPU, FPGA etc.) because we are going to use AI…
  - A set of dedicated servers for middleware which will manage processing workflow, monitoring and other service needs.
  - Buffer for intermediate output and for data prepared for transfer to long-term storage and future processing.



# Payload



Data unpacking

ML Reco

Classic reco

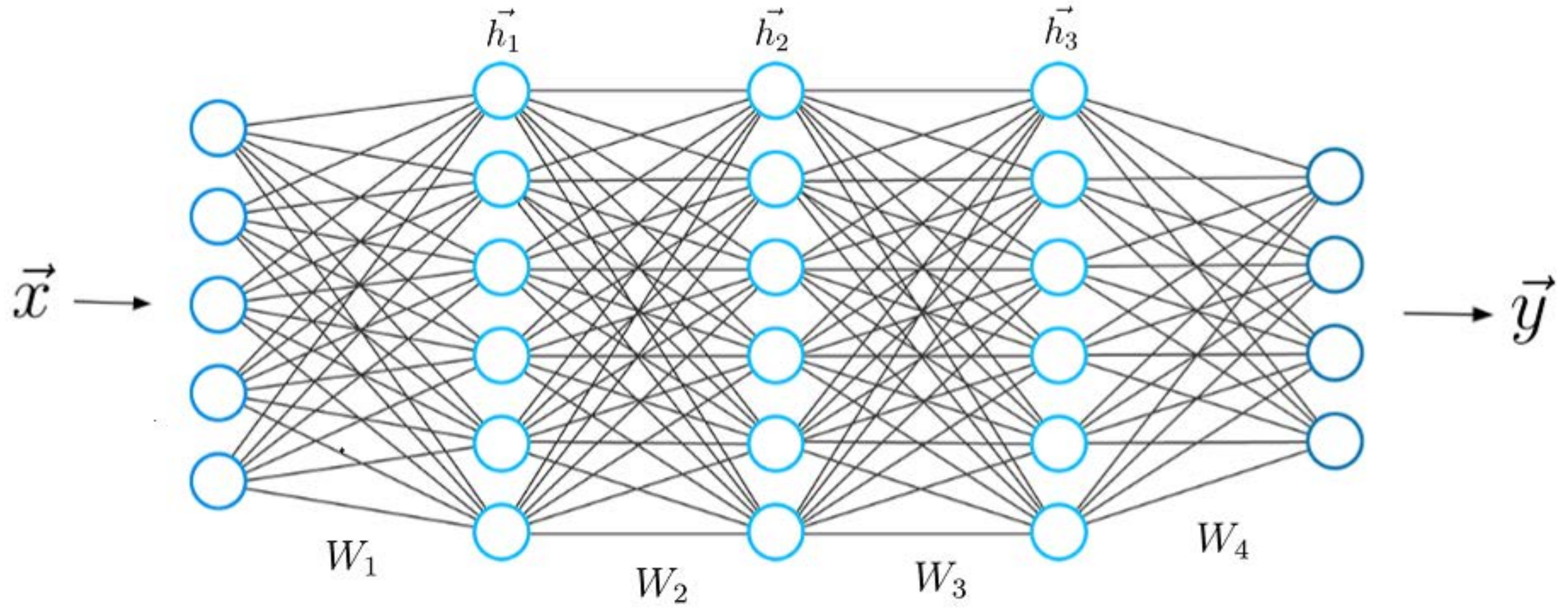
Event building

Online polarimetry

Event selection

Performance and data quality monitor

Merge and packing data for offline processing



Machine learning is a promising technology

# Middleware basic functionality

Data management;

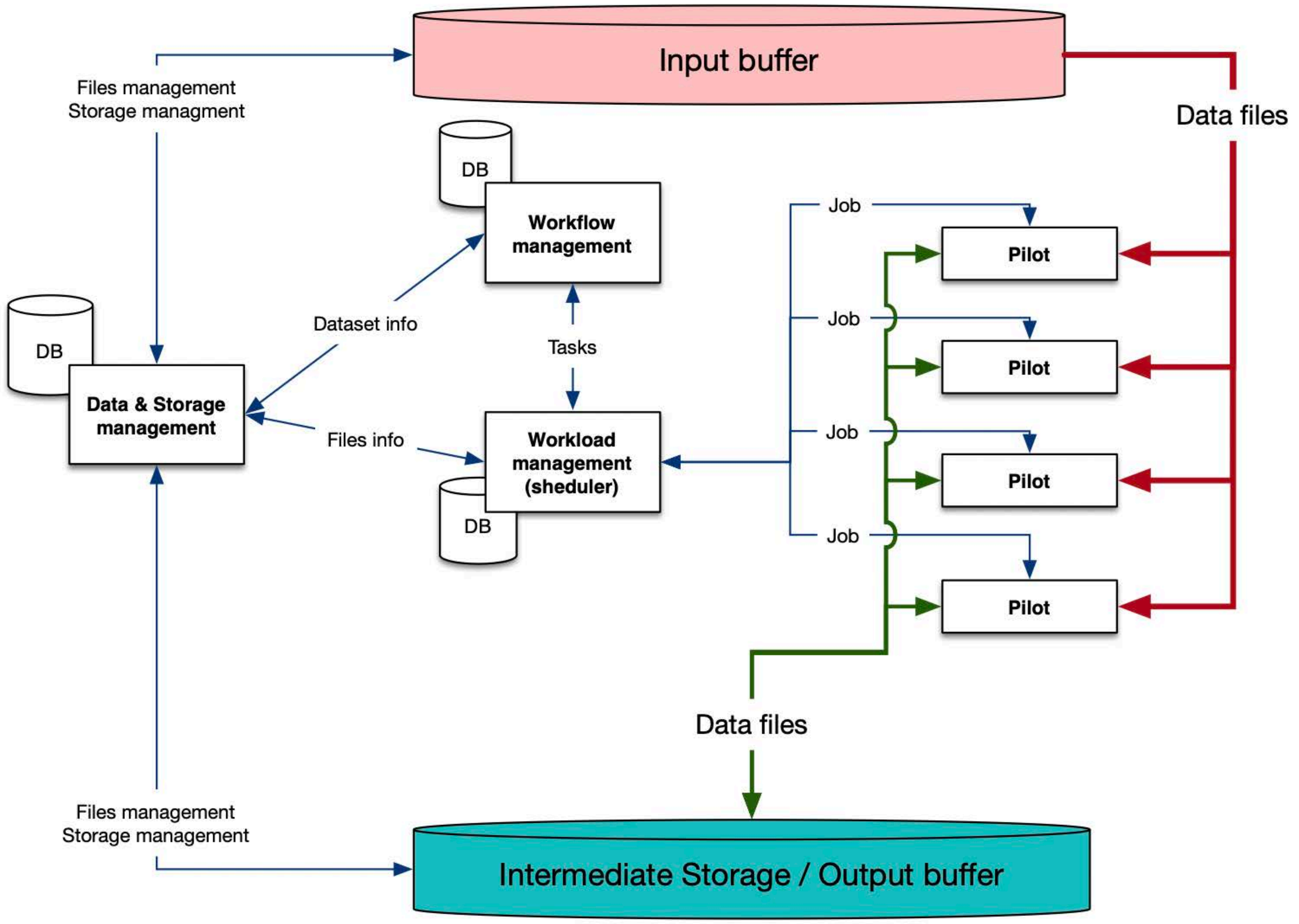
- *Support of data catalogue and storages management;*

Processing management;

- *Define processing chains*
- *Executes processing chains through generation of computational tasks*
- *Each task had to process a block of homogenous data*

Workload management:

- *Generate required number of jobs to perform a task*
- *Dispatch jobs to compute nodes through pilot application;*
  - *Control of jobs executions;*
  - *Control of pilots (identifying of "dead" pilots)*



# Workflow management system

- High level system which interacts with data management system and workload management system for realising multistep processing of blocks of data (datasets)
- The functional decomposition of the subsystem was carried out and the initial set of microservices was proposed
  - “Chain definer” - human oriented (web) application which allow define sequences of processing steps
  - “Processing starter” - microservice responsible for triggering of processing chains
  - “Chain executor” - microservice responsible for control of execution of processing chain.
- To do: coding a set of interfaces as for user so with external systems, tuning of state model.

# Data management system

- Support of data catalogue and storages management;
- A lot of work already done for data management system starting from decomposition of functionality to microservices, definition of set of tools for storage management, realisation of DB design for data catalogue (datasets and files) up to definition and realisation of the set of internal and external interfaces
- Microservices:
  - dsm-register – responsible for registration of input data from DAQ in the catalogue
  - dsm-manager – realise interfaces to the catalogue for subsystems
  - dsm-inspector – realise auxiliary tools for storage management (consistency check, cleanup, dark data identification)

# Workload management

- Realize a task execution process by shredding a required number of jobs to provide controlled loading to compute  facility, tacking into account priority of tasks and associated jobs. A task is one step in a processing chain of a block of data. Job is a processing of a single piece of data (file or few files).
- Microservices: task manager, task executor, job manager, job executor
- Workload management system intensively interacts with Pilots, Data Management system, accept tasks from Workflow management system and reports progress of execution back
- Base architecture in place, coding of cross service and and external interfaces.

# Pilot

- Pilot is the application to manage the execution of a job, prodused by  WMS, on a compute node. It responsible for setting up of environment, stage-in/out of data from storage to compute node, execution of payload and monitoring of different conditions during execution. Intensively interacts with workload management system
- Base architecture and initial functionality of pilot application is defined. It is a multithread application with interactions between threads through queues
- In progress intensive development of interfaces and initial definition of internal entities

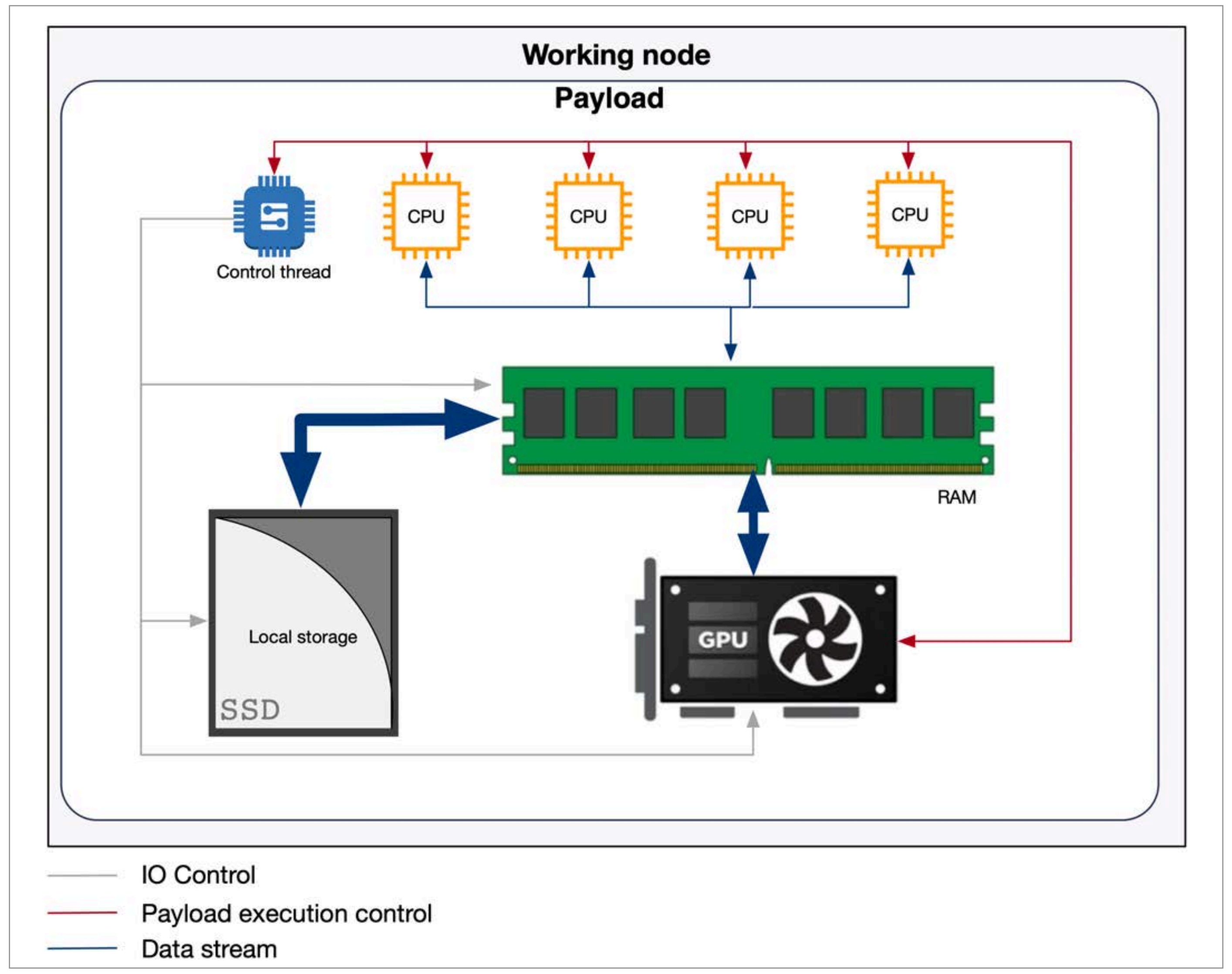
# What else?

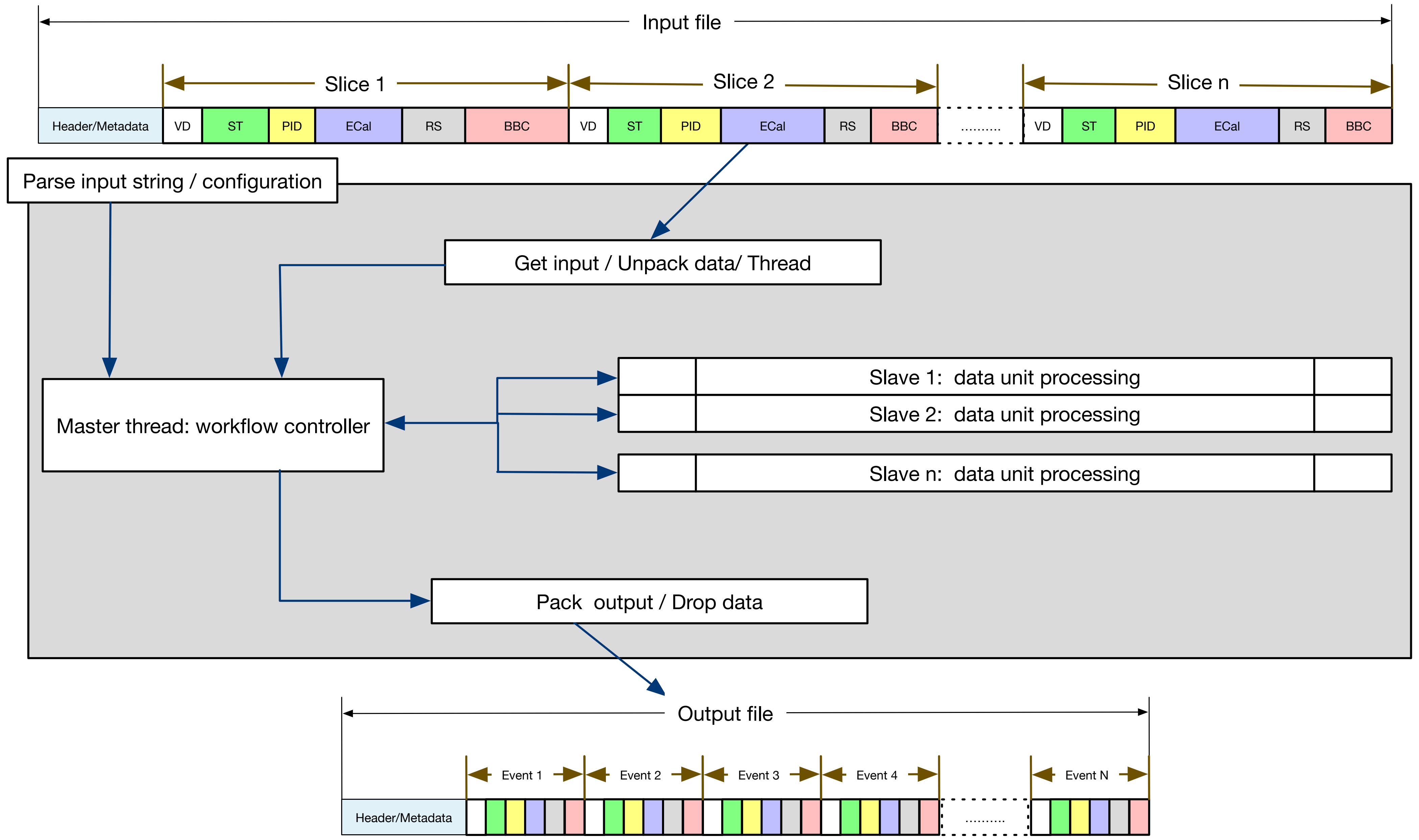
## in middleware

- **Monitoring**
- Sources of auxiliary data: mapping, geometry, etc.
- Applied software repository management
- Middleware deployment machinery

# Applied software: framework

- Design principles:
  - Separation of Algorithms and Data
  - Interfaces between User code and framework
  - Separation between transient and persistent data
- Multicore architectures support (OneAPI/OneTBB?)




Input file
Slice 1
Slice 2
Slice n
Header/Metadata
VD
ST
PID
ECal
RS
BBC
..........
Parse input string / configuration
Get input / Unpack data/ Thread
Master thread: workflow controller
Slave 1: data unit processing
Slave 2: data unit processing
Slave n: data unit processing
Pack output / Drop data
Output file
Event 1
Event 2
Event 3
Event 4
Event N
Header/Metadata
..........

# BackUp