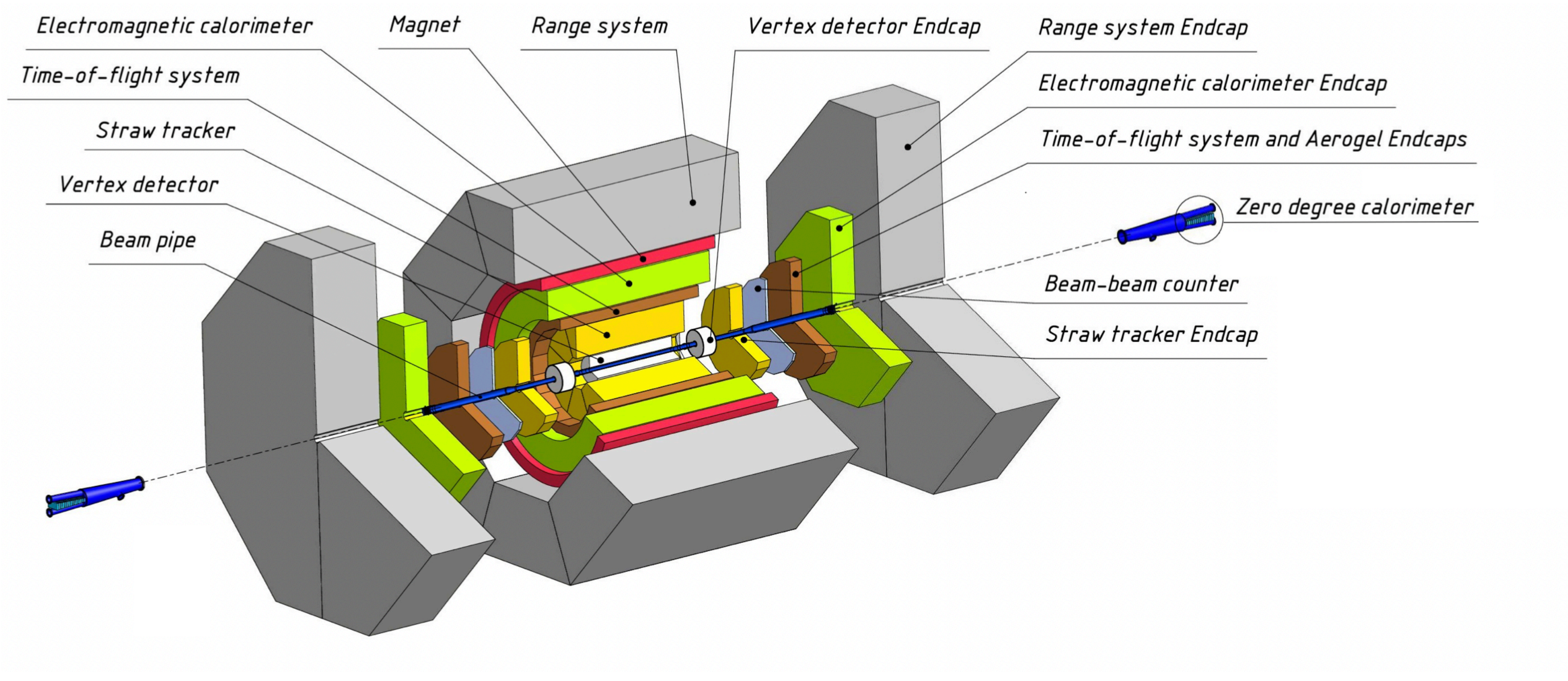
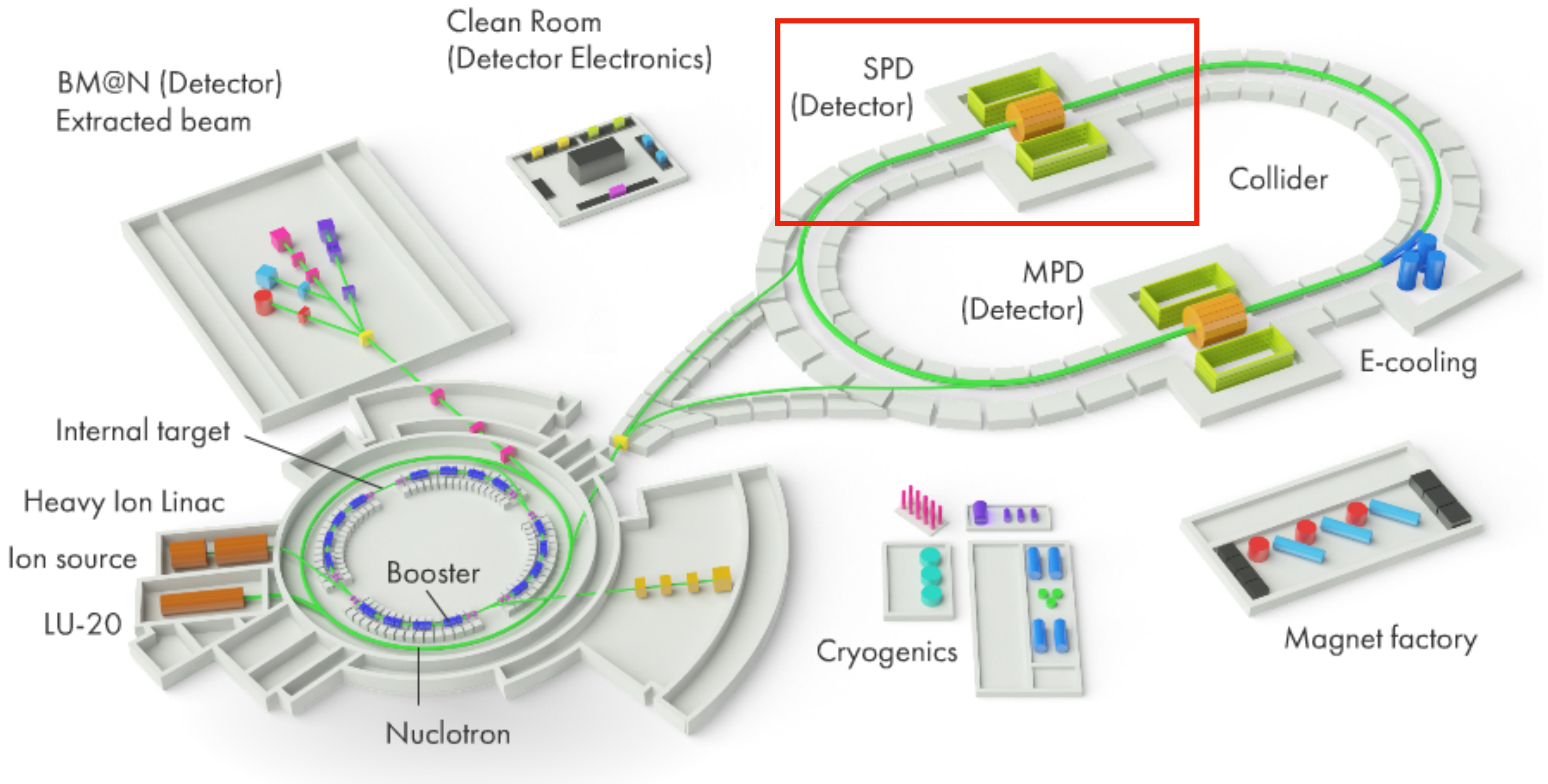
SPD Computing

Oleynik D. JINR LIT

SPD Spin Physics Detector

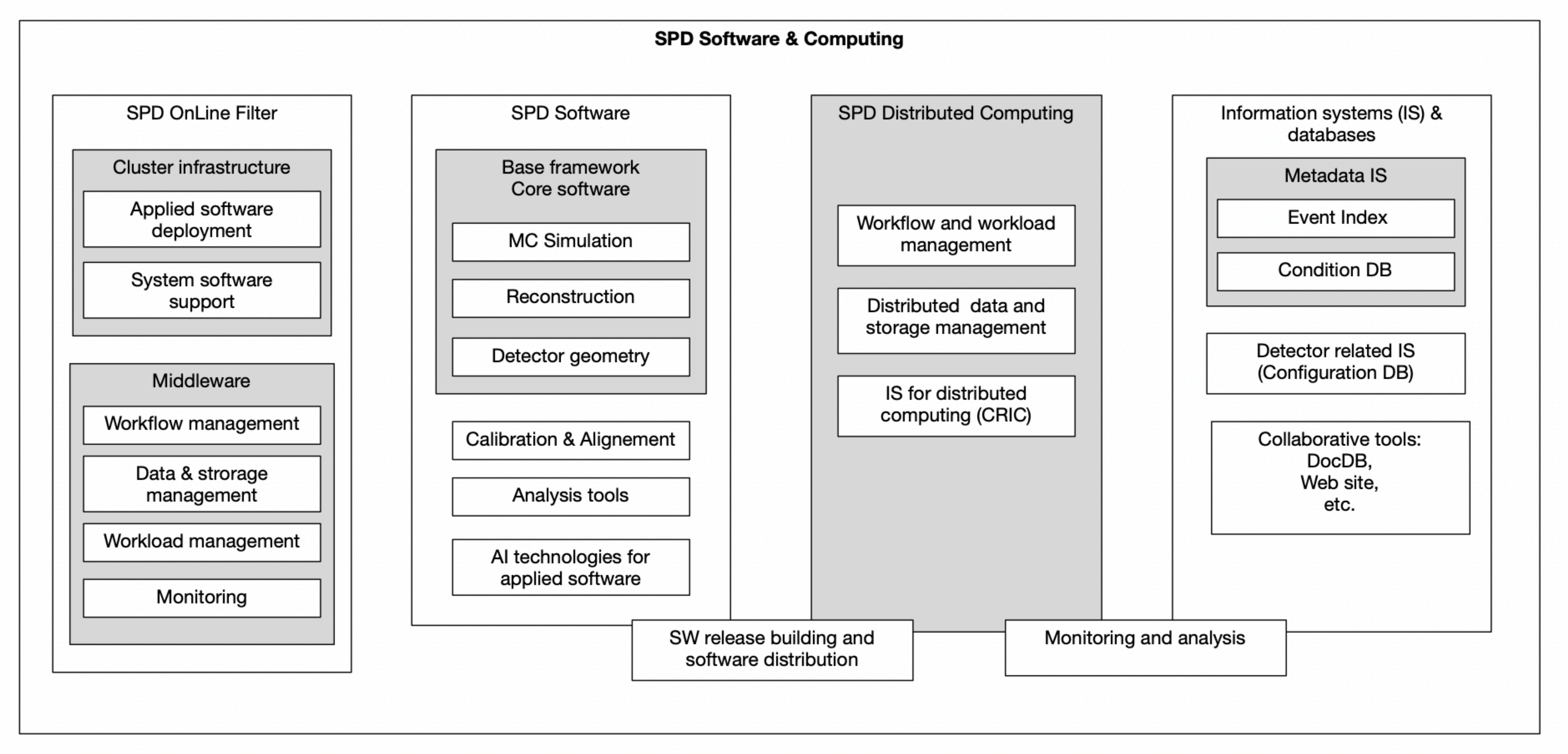
Study of the nucleon spin structure and spin-related phenomena in polarized *p-p*, *d-d* and *p-d* collisions





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

SPD Software and computing



SPD Software & Computing

SPD OnLine Filter

- Cluster infrastructure
  - Applied software deployment
  - System software support
- Middleware
  - Workflow management
  - Data & strorage management
  - Workload management
  - Monitoring

SPD Software

- Base framework Core software
  - MC Simulation
  - Reconstruction
  - Detector geometry
- Calibration & Alignement
- Analysis tools
- AI technologies for applied software

SPD Distributed Computing

- Workflow and workload management
- Distributed data and storage management
- IS for distributed computing (CRIC)

Information systems (IS) & databases

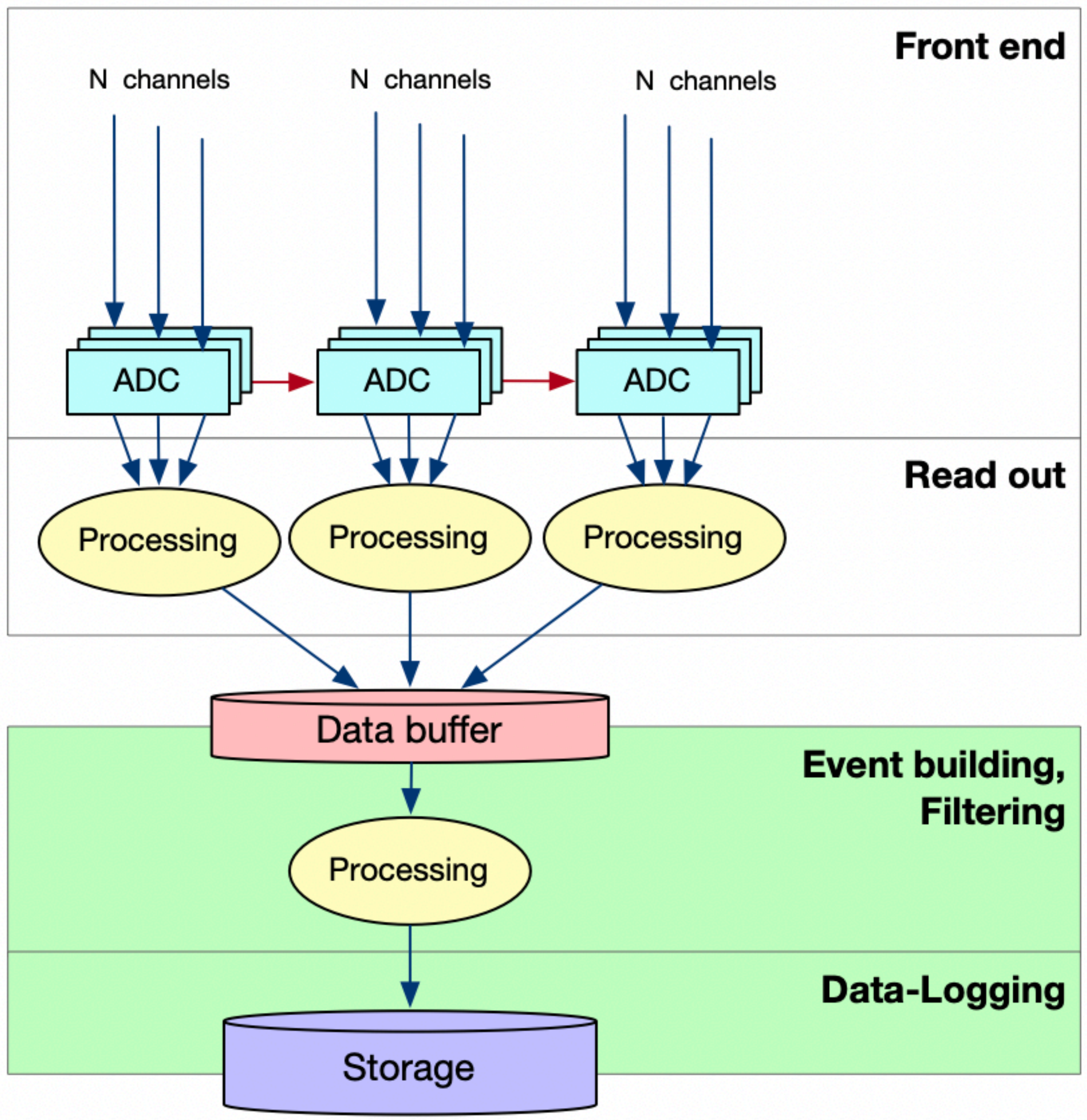
- Metadata IS
  - Event Index
  - Condition DB
- Detector related IS (Configuration DB)
- Collaborative tools: DocDB, Web site, etc.

SW release building and software distribution

Monitoring and analysis

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



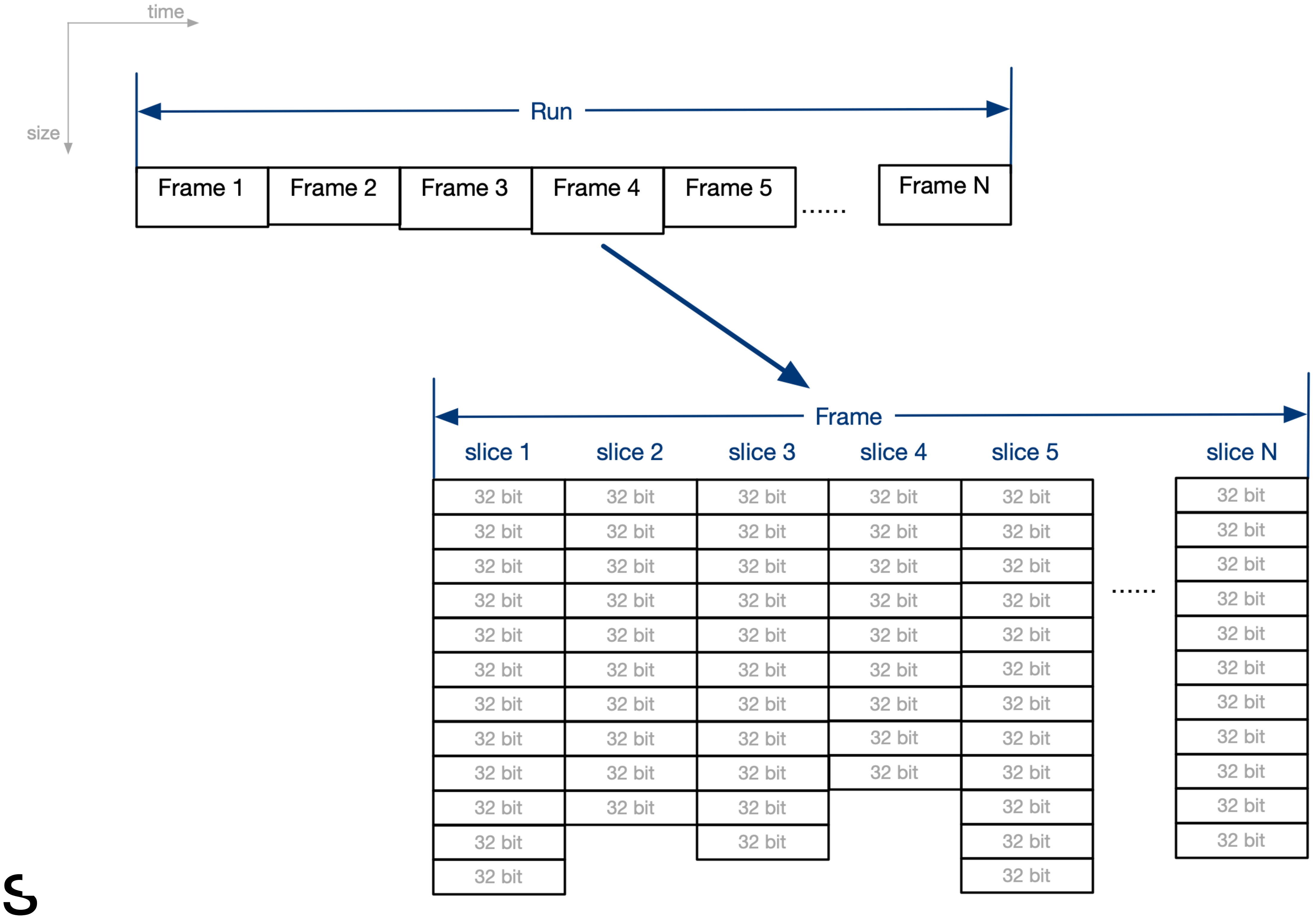
SPD as data source

- 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
- Comparable amount of simulated data

High-throughput computing for SPD data processing

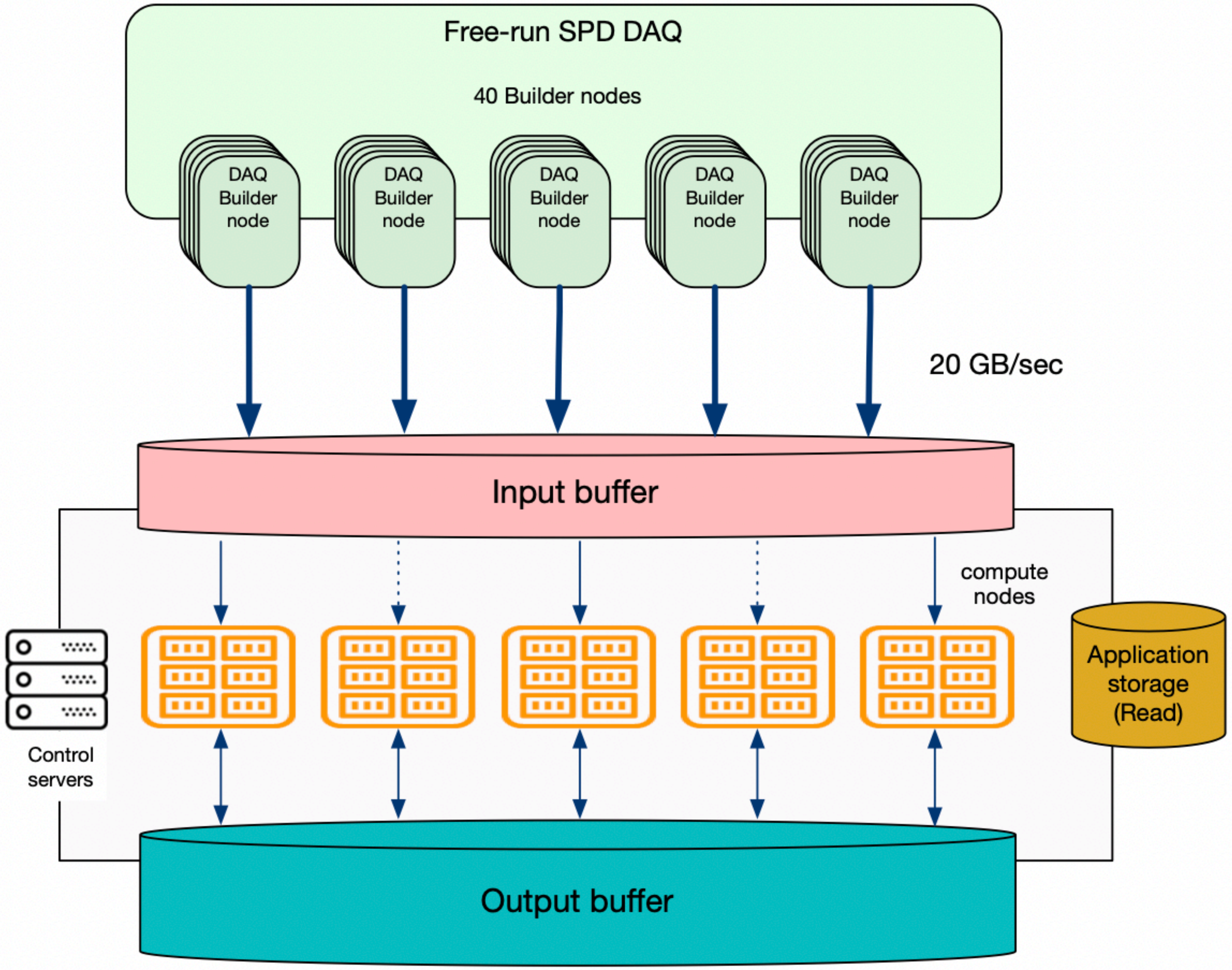
High-throughput computing (HTC) involves running many independent tasks that require a large amount of computing power.

- 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



Online filter infrastructure

- 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 managing of processing workflow, monitoring and other service needs.
- Buffer for intermediate output and for data prepared for transfer to long-term storage and future processing.

Dispatcher required functionality

Data management;

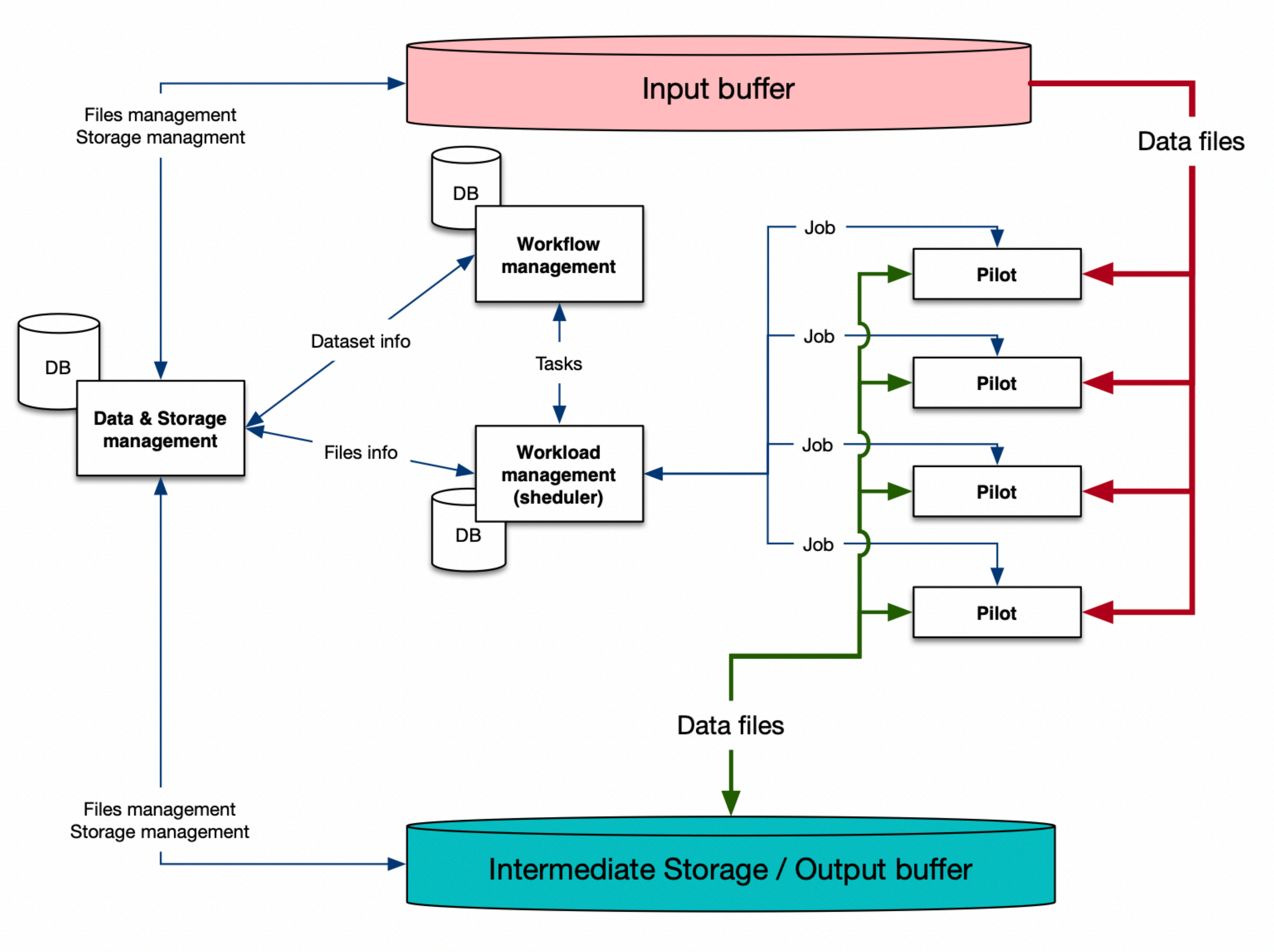
- *Support of data lifetime (registering, global transfer, cleanup);*

Processing management;

- *Generate jobs for each type of processing:*
  - *Events identification (building);*
  - *Verifying of processing results (AI vs traditional processing);*
  - *Select (Filter) events;*
  - *Pack (merge) output data for transferring to “offline”;*

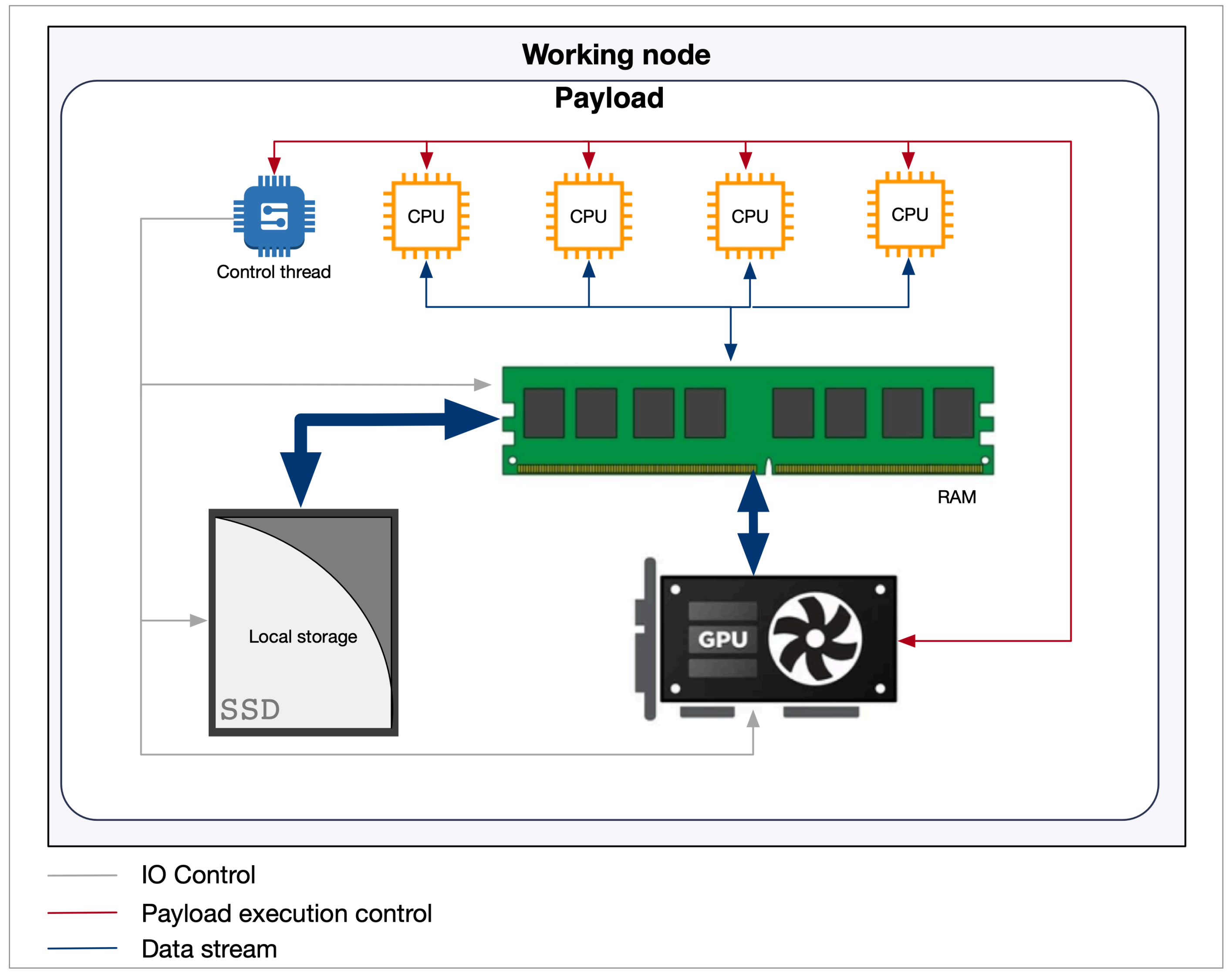
Workload management:

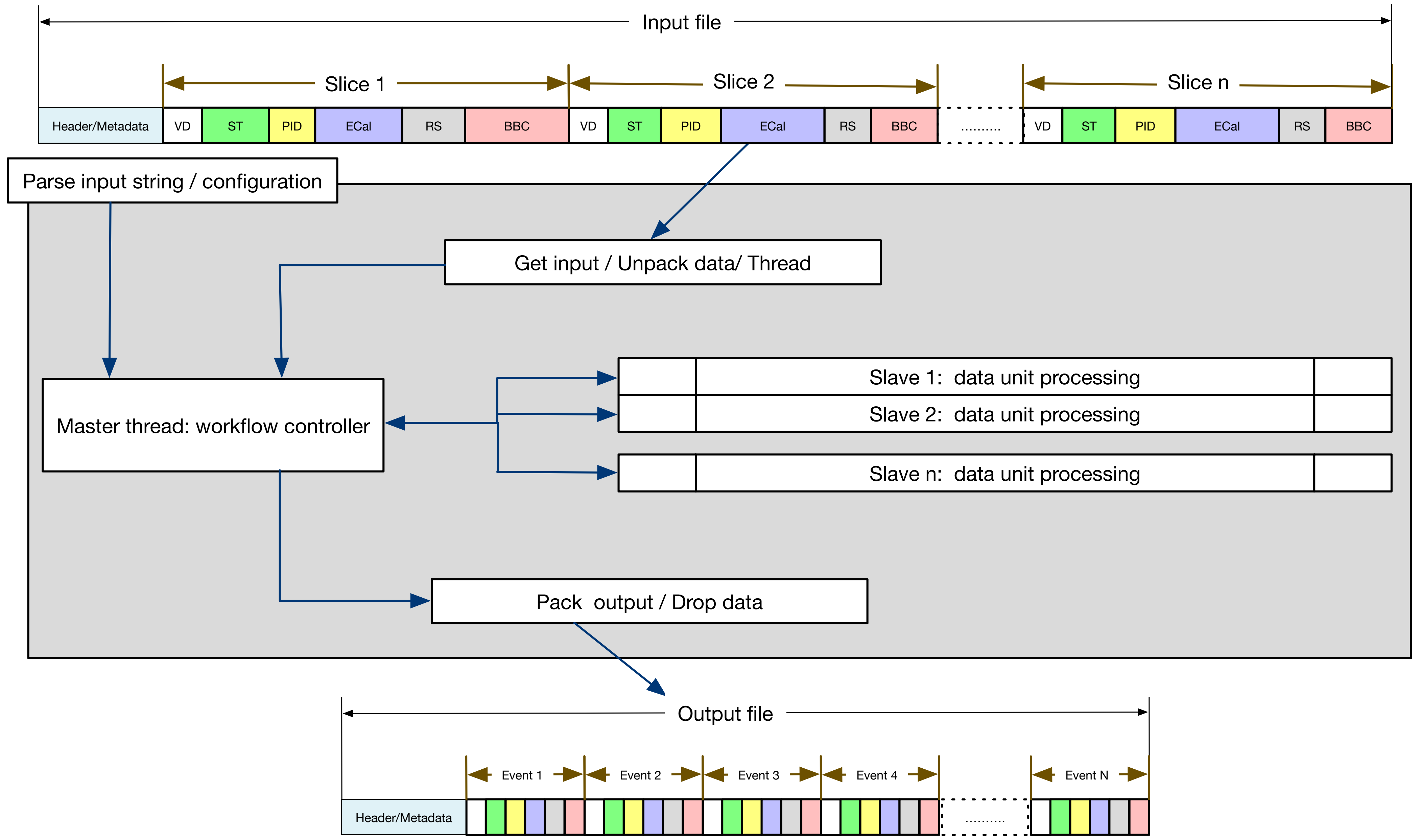
- *Dispatch jobs to pilots;*
- *Control of jobs executions;*
- *Control of pilots (identifying of "dead" pilots)*



Multithread processing

- Multicore computers already reality
  - Efficient usage requires multithreading processing
  - A lot of algorithms in HEP software stack does not support multithread execution (yet)
- We tries to explore multithread processing on data layer (each thread process own piece of data)

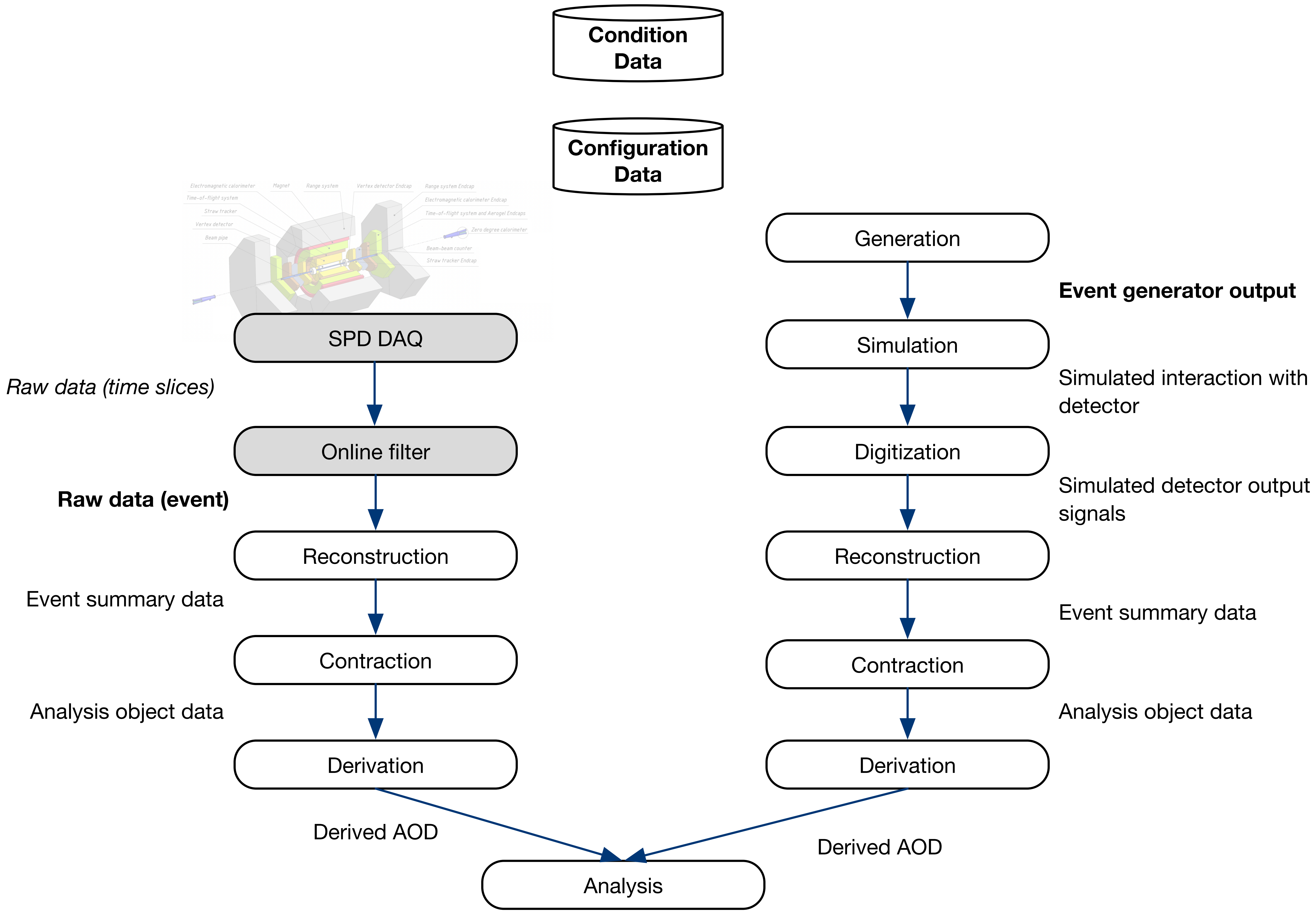




Offline processing

Reconstruction, Simulation

- Amount of data reduced, but data is not ready for analysis yet
  - Events contain raw or partially reconstructed data
  - Calibration and alignment is not applied yet
- Simulation - pure computation processing (will start mach early than apparatus will be ready)
  - Will require significant amount of computing resources



Condition Data

Configuration Data

SPD DAQ

Raw data (time slices)

Online filter

Raw data (event)

Reconstruction

Event summary data

Contraction

Analysis object data

Derivation

Derived AOD

Generation

Event generator output

Simulation

Simulated interaction with detector

Digitization

Simulated detector output signals

Reconstruction

Event summary data

Contraction

Analysis object data

Derivation

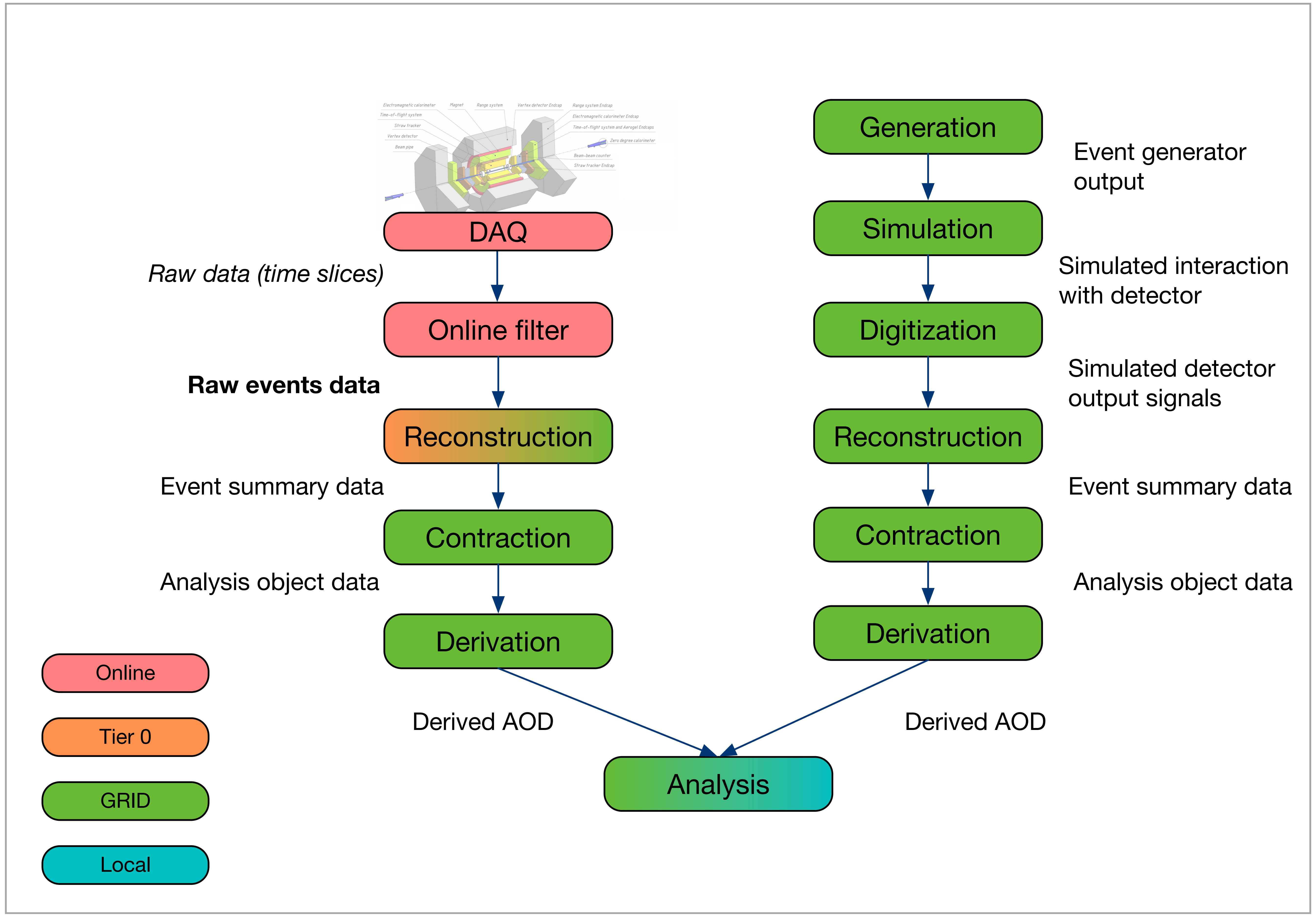
Derived AOD

Analysis

- Another type of computing facility required for routine offline processing – distributed data processing system (aka grid)

Processing steps and data types

- As reconstruction as simulation – are multistep workflows
  - Each step produces own data type, which correspond to different representation of events
  - So size of event will be different in different data type
- Why we need different types?
  - Some types of processing, like raw data, quite expensive or unique, producing of other types is resource consuming, another types good for long term storage but not optimal for final analysis because of redundancy



DAQ

Raw data (time slices)

Online filter

Raw events data

Reconstruction

Event summary data

Contraction

Analysis object data

Derivation

Derived AOD

Generation

Event generator output

Simulation

Simulated interaction with detector

Digitization

Simulated detector output signals

Reconstruction

Event summary data

Contraction

Analysis object data

Derivation

Derived AOD

Analysis

Online

Tier 0

GRID

Local

- Tier 0 – entry point to offline processing

Estimated data volumes in numbers

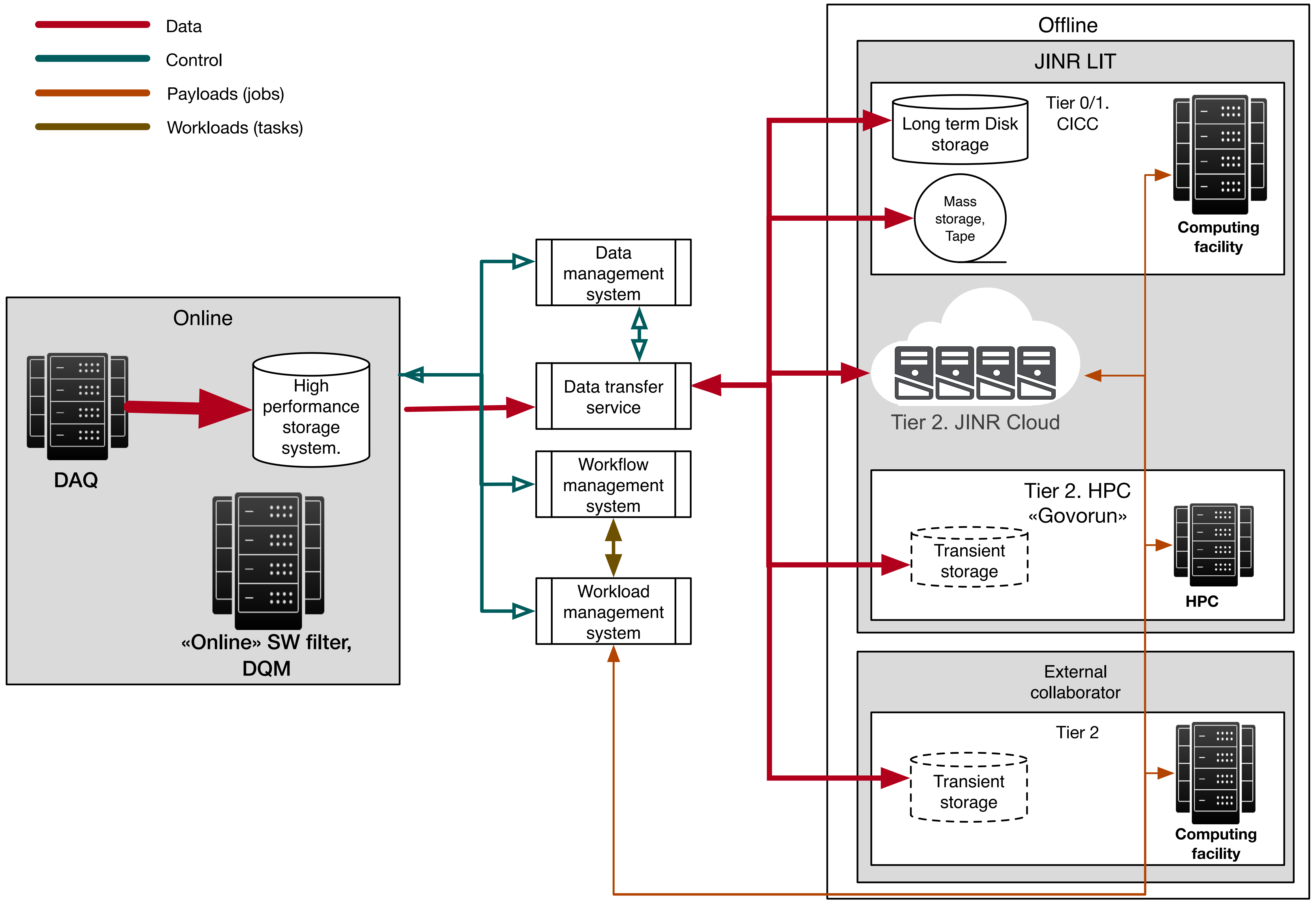
Why we need grid?

- Expected that 2*10^{12} events per year (EPY) should be processed
  - One trillion of reconstruction and one trillion of simulation (yep, this is BigData)
- With processing rate of one event per second per CPU - we will need to have more than 63000 fully loaded CPUs during the year
- To handle load of such level, distributed system will require to deal with any available computing resource like remote cluster, cloud infrastructure or HPC
- It’s quite hard to estimate requirements for storage resources for the moment, but even with size of event in few KB required storage will be on the level of tens of PB

Managing of data processing

in heterogenous distributed computing system

- Key middleware components required for efficient processing in grid:
    - **Workflow management system** - control the process of processing of data on each step of processing. Produce tasks, which required for processing of certain amount of data, manages of tasks execution.
    - **Workload management system** - processes tasks execution by the splitting of the task to the small jobs, where each job process a small amount of data. Manage the distribution of jobs across the set of computing resources. Takes care about generation of a proper number of jobs till task will not be completed (or failed)
    - **Data management system** - responsible for distribution of all data across computing facilities, managing of data (storing, replicating, deleting etc.)
    - **Data transfer service**: takes care about major data transfers. Allow asynchronous bulk data transfers.


Data
Control
Payloads (jobs)
Workloads (tasks)
Online
DAQ
High performance storage system.
«Online» SW filter, DQM
Data management system
Data transfer service
Workflow management system
Workload management system
Offline
JINR LIT
Long term Disk storage
Tier 0/1. CICC
Mass storage, Tape
Computing facility
Tier 2. JINR Cloud
Tier 2. HPC «Govorun»
Transient storage
HPC
External collaborator
Tier 2
Transient storage
Computing facility