Offline Software and Computing for the SPD experiment

Alexey Zhemchugov
(JINR)
on behalf of SPD Collaboration

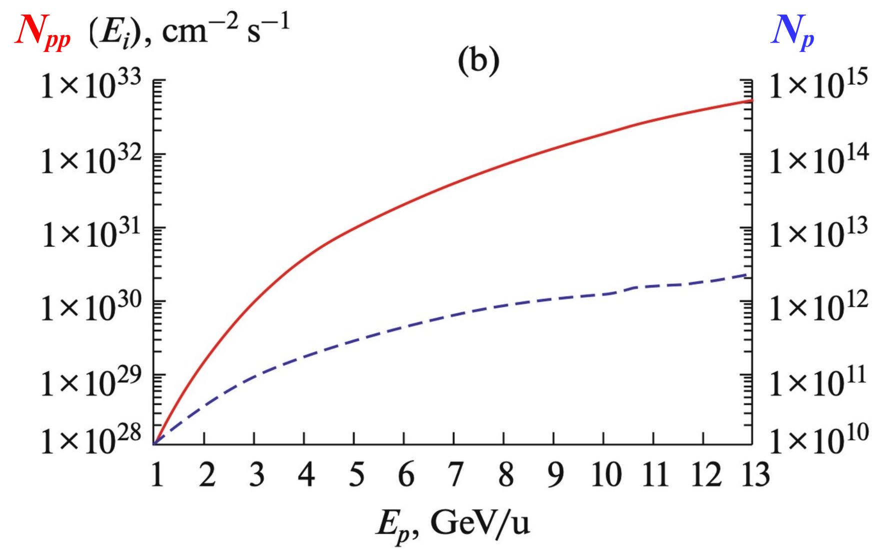
08 July 2021

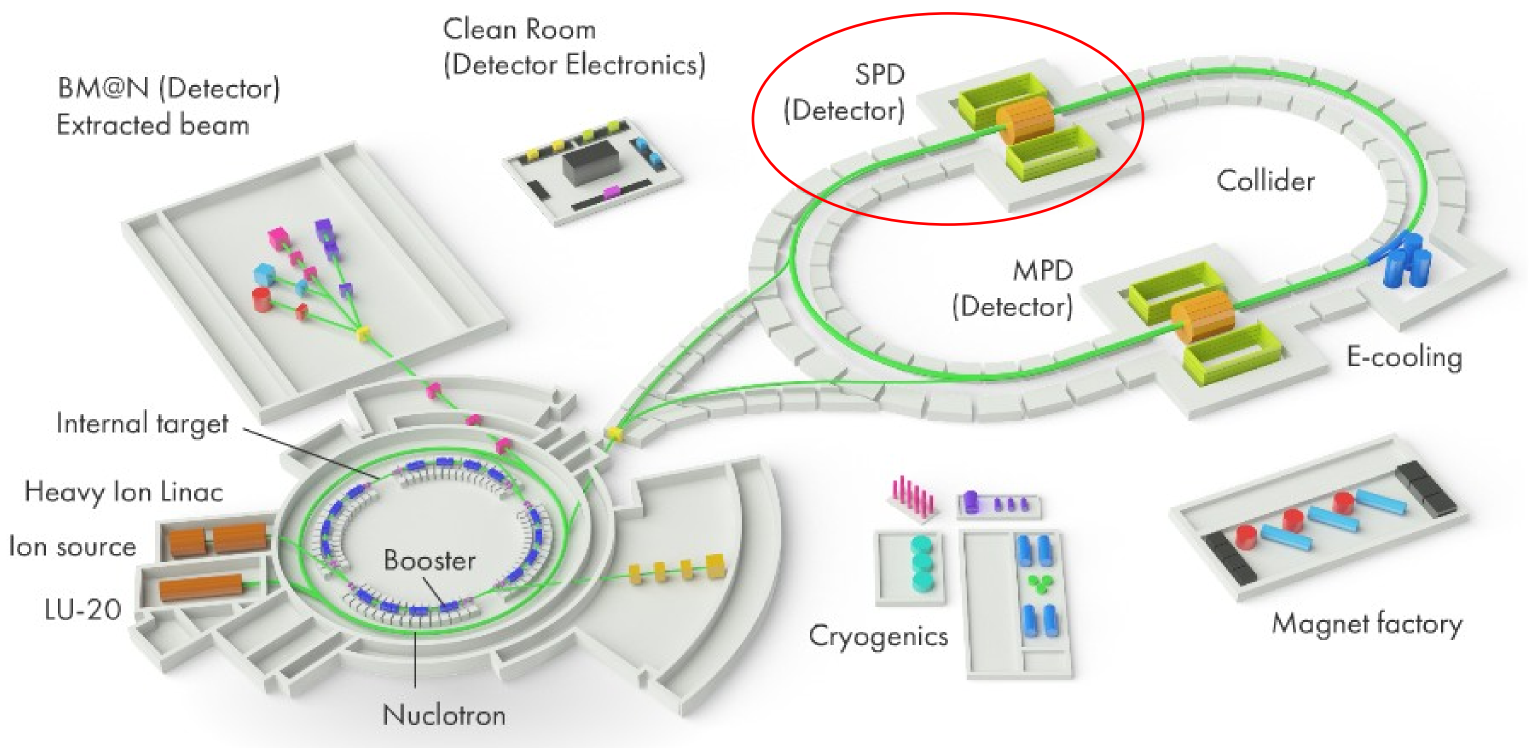
The 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2021)

SPD at NICA

p ↑ p ↑ : $\sqrt{s} \leq 27$ GeV
d ↑d ↑ : $\sqrt{s} \leq 13.5$ GeV
d ↑p ↑ : $\sqrt{s} \leq 19$ GeV

U, L, T
|P|>70%





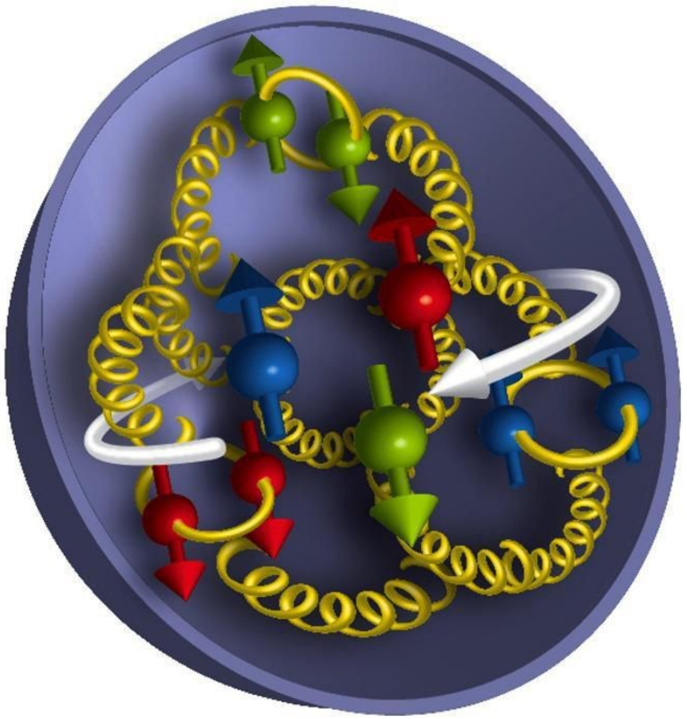
Brief history of SPD

- 2007 — idea of the project, SPD was included as an activity to the general list of NICA activities at JINR
- 2014 — Letter of Intent (approved by the JINR Program Advisory Committee)
- 2016, 2018 — SPD-oriented workshops in Prague
- 2019 — SPD project is approved by PAC (till 2022)
- 2019 — first proto-collaboration meeting
- 2020 — Preparation of the Conceptual Design Report *http://arxiv.org/abs/2102.00442*
- 2021 – SPD Collaboration was born. Preparation of the Technical Design Report
- 2025+ – Start operation

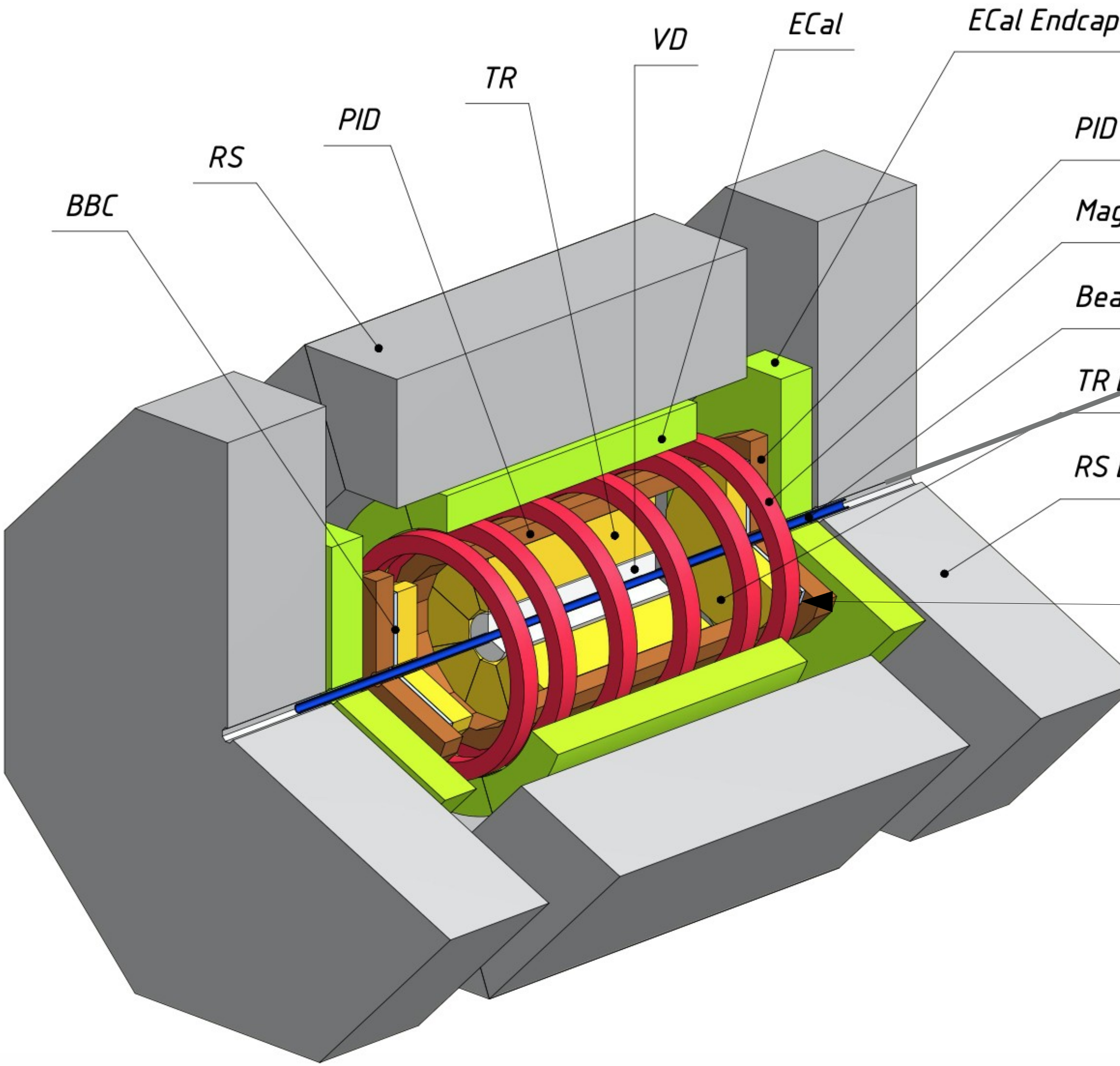
Physics program

- SPD - a universal facility for comprehensive study of gluon content in proton and deuteron at large x
  - Prompt photons
  - Charmonia
  - Open charm
- Other spin-related phenomena
- Other physics

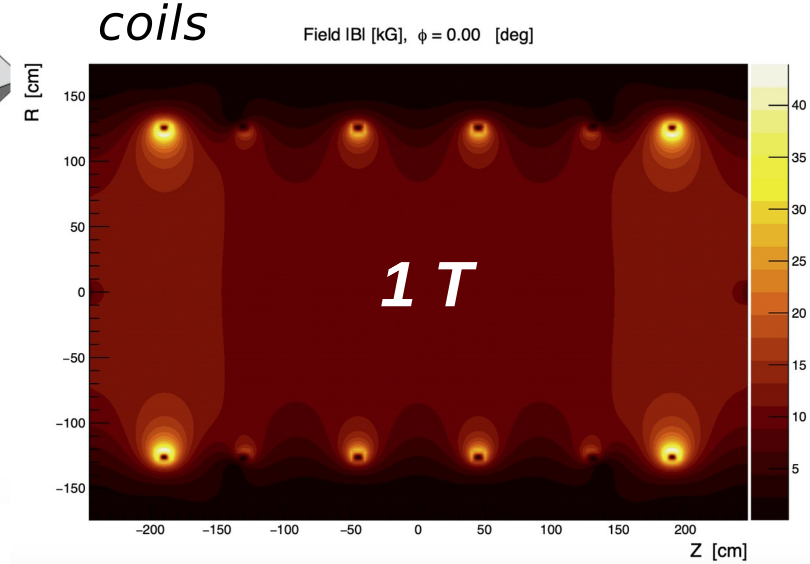
More details in arXiv:2011.15005



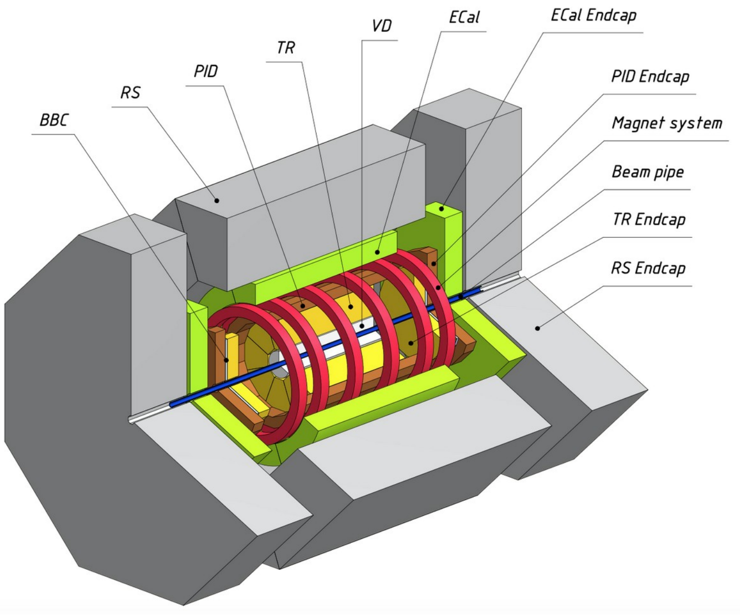
Detector overview



6 superconductive solenoidal coils



SPD as a data source

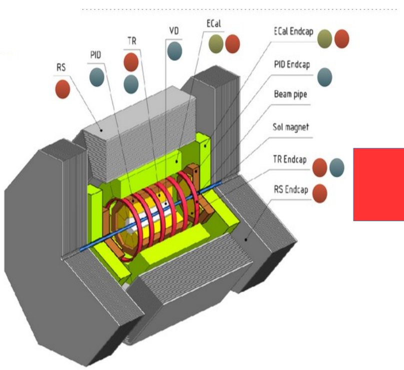


- Bunch crossing every 76.3 ns = crossing rate 13 MHz
- ~ 3 MHz event rate (at 10^{32} $cm^{-2}s^{-1}$ design luminosity) = pileups
- 20 GB/s (or 200 PB/year (raw data), $3*10^{13}$ events/year)
- Selection of physics signal requires momentum and vertex reconstruction → no simple trigger is possible

The SPD detector is a medium scale setup in size, but a large scale one in data rate!

Data workflow

SPD Hall



DAQ

Online Filter
Monitoring
DQA

Tier-0: JINR Computing Center

Reco

Worldwide Grid

Analysis

Reco

MC

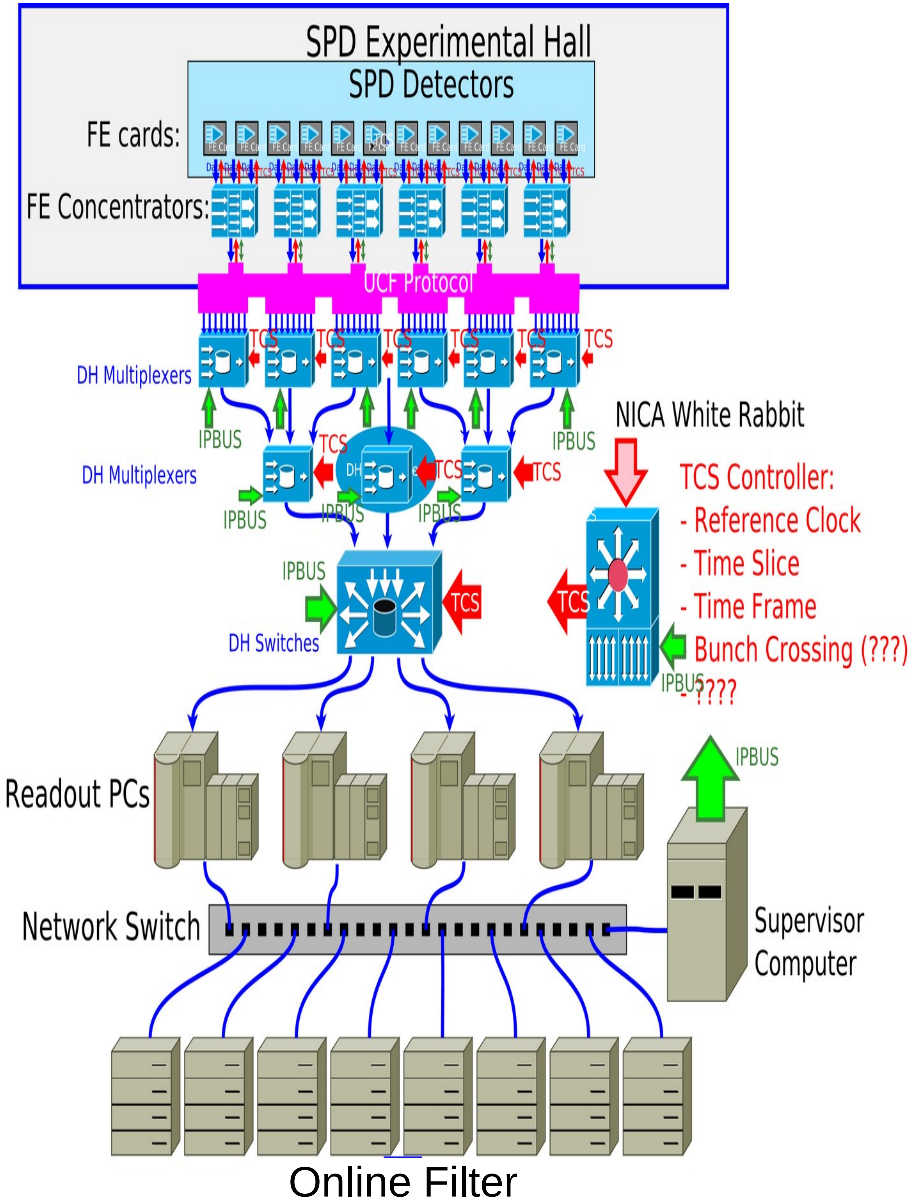
Continuous data reduction:

- DAQ: noise suppression
- Online filter: event building, partial reconstruction, software trigger
- Offline computing: data analysis and long term storage

Free-running DAQ

- No hardware trigger
- Self-triggered FEE digitizes data and sends it to DAQ
- Zero suppression
- Timestamp added
- Several FEE options are being considered, using experience of PANDA and COMPASS projects
- Data from FEE is collected and managed by FPGA-based DAQ system

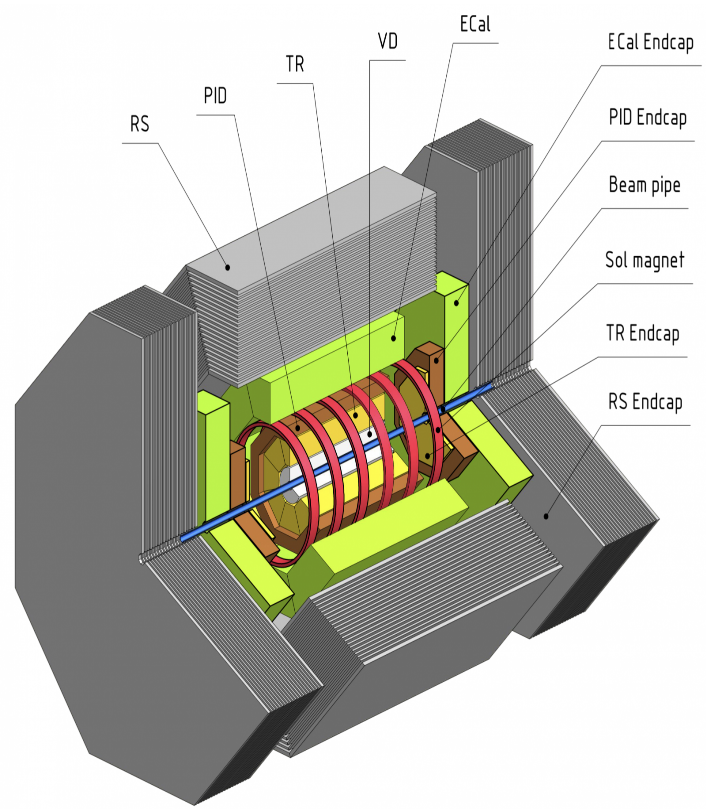
FPGA-based free-running DAQ

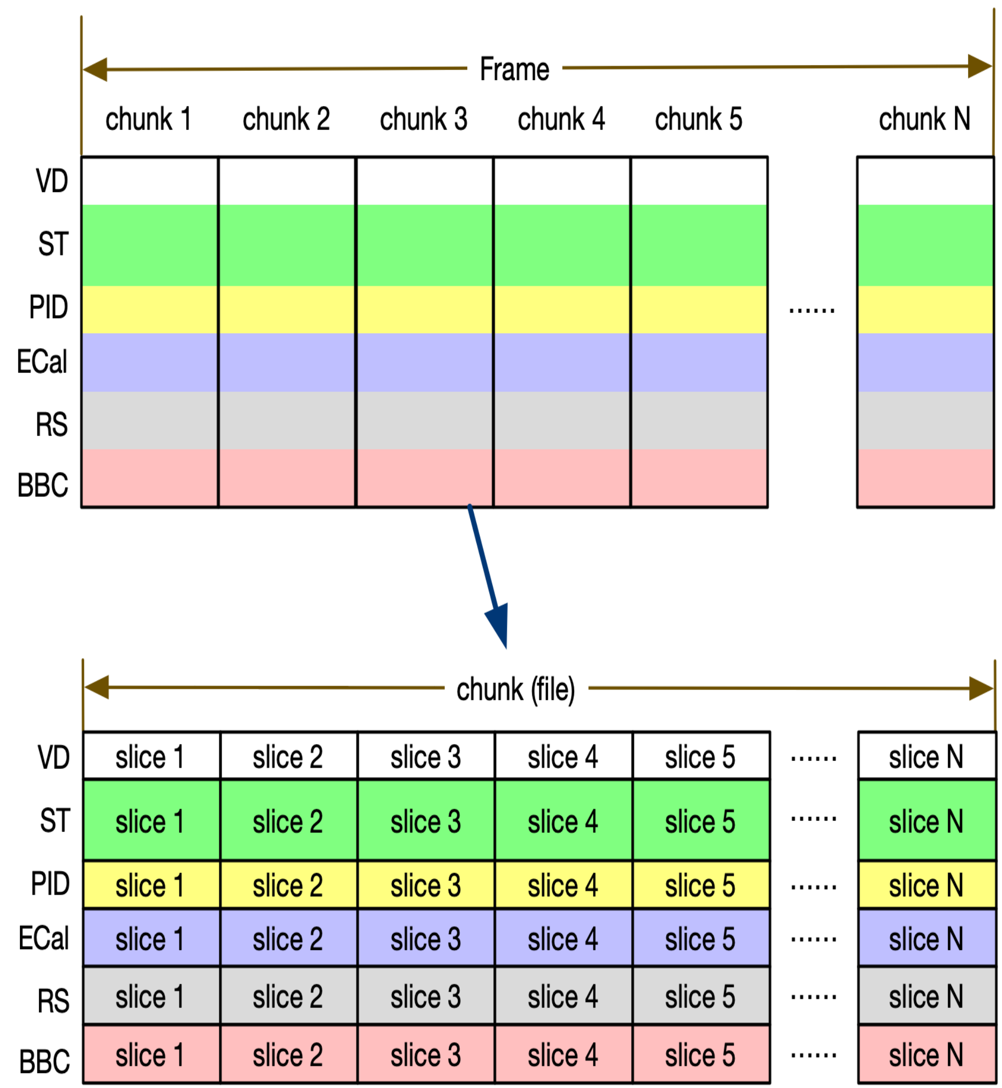


- The concept is confirmed at Belle2 and COMPASS
- White Rabbit is used for time reference
- Identical DAQ modules (TCS, data concentrators, MUXs and switches) for all subsystems
  - only FPGA firmware is different
- Simple data treatment (e.g. noise removal) is possible

Input data structure

No trigger = No classical events anymore





Online Data Filter

High-performance heterogeneous computing cluster

- Partial reconstruction
  - Fast tracking and vertex reconstruction
  - Fast ECAL clustering
- Event unscrambling
- Software trigger
  - several data streams

Machine learning is a key technology

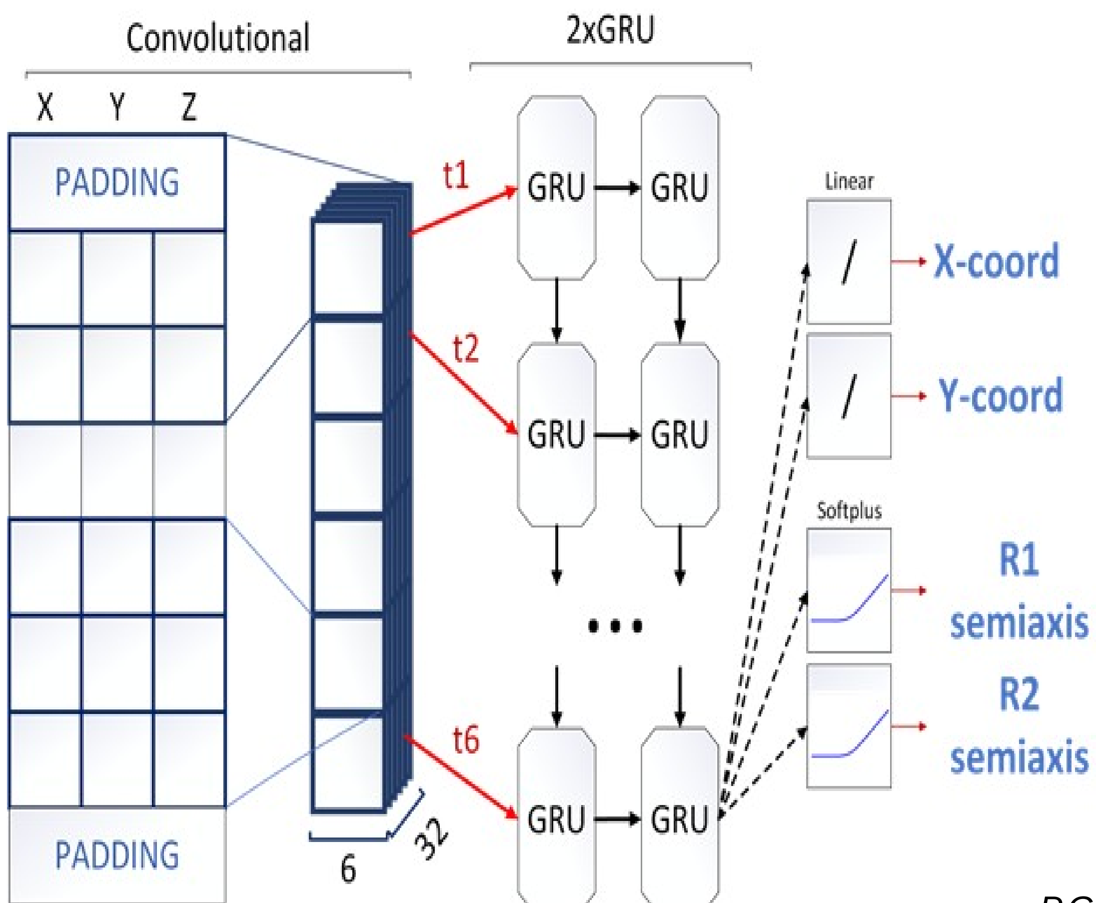
- Monitoring and Data quality assessment
- Local polarimetry

Online reconstruction

- Fast ML reconstruction
  - tracking
  - primary vertex reconstruction
  - ECAL clusters
  - π^0 reconstruction, ZDC, BBC for online polarimetry
  - RS tracks and showers
- python → C++
- Continuous monitoring of the ML reco performance is needed to keep control of systematics
- Classic reconstruction: the same as ML reconstruction but using traditional algorithms for small fraction of data
- Assume that precise calibration constants and alignment are not available
- Assume that noise level is not known *a priori*

Example: TrackNETv2

Model Architecture



- Works like learnable version of the Kalman filter
- for the starting part of a track predicts an elliptical area at the next station where to search for the continuation
- if there is not continuation candidate track is thrown away

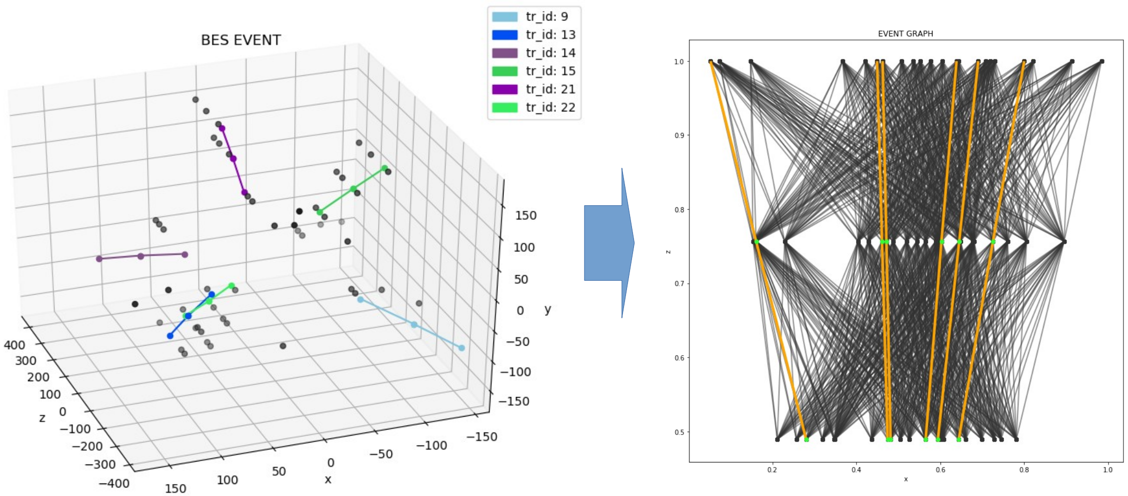
Results (BM@N experiment, NICA):

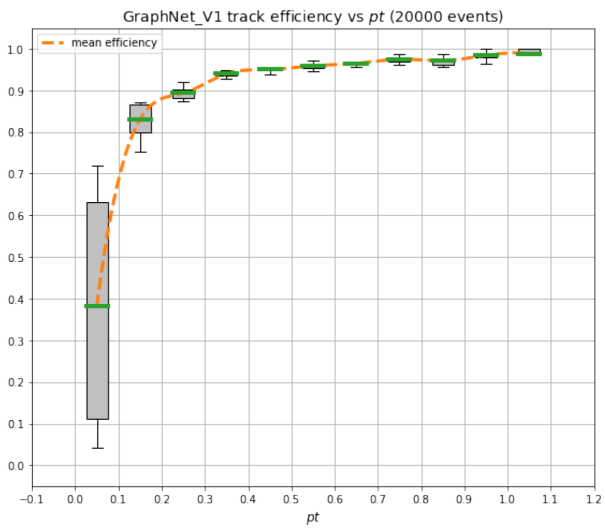
- 12K tracks/sec on Intel Core i3-4005U @1.70 Ghz
- 96% of tracks were reconstructed without any mistake

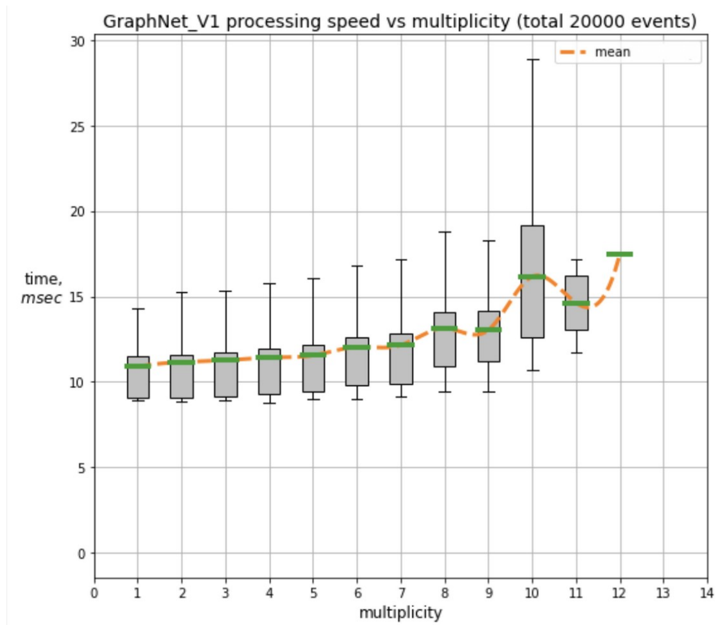
P.Goncharov, G. Ososkov et al. arXiv:1812.03859

Ososkov G.A. et al., Computer Research and Modeling, 2020, vol. 12, no. 6, pp. 1361-1381

Example: GraphNet



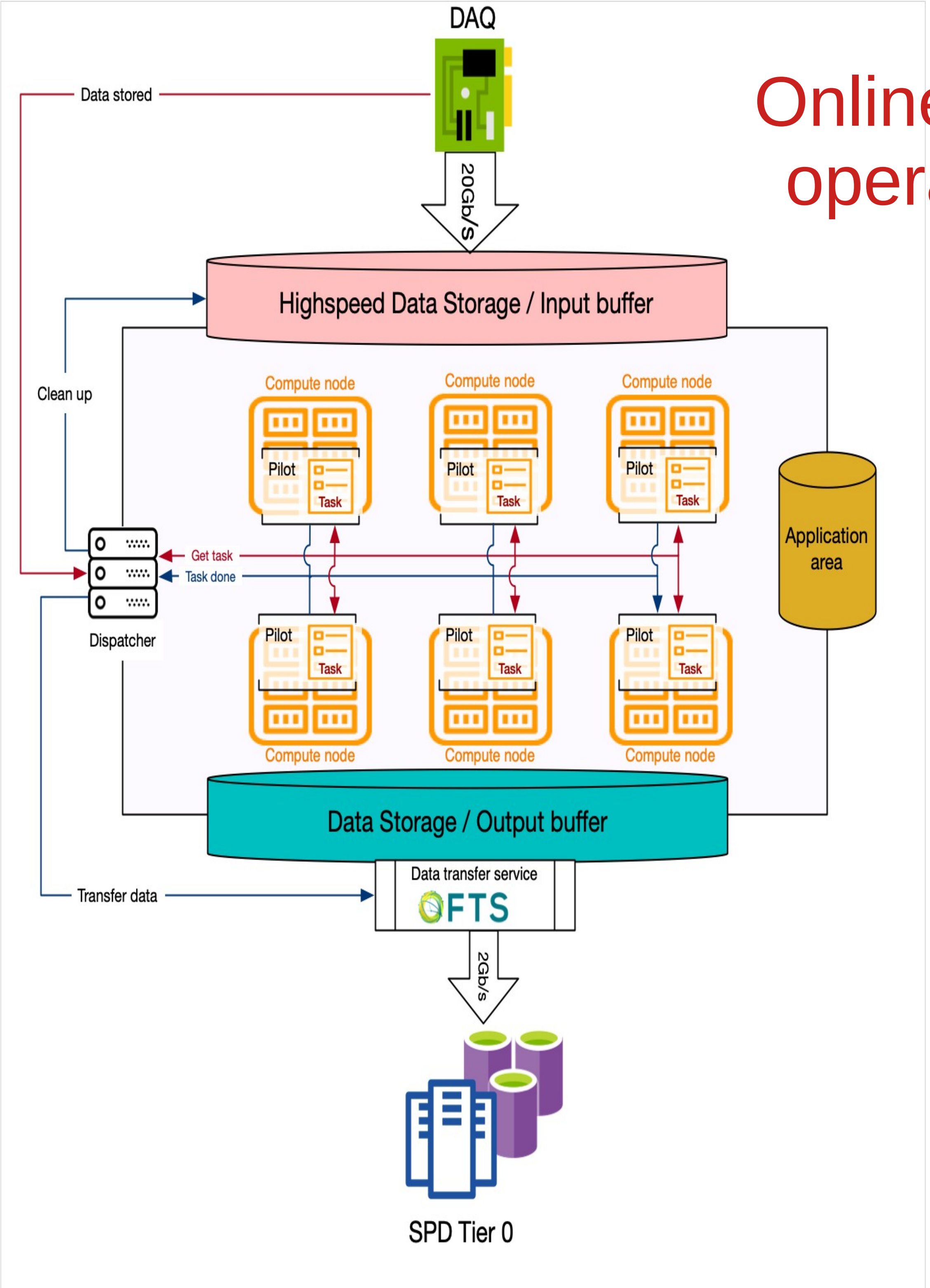




- An event is represented as a graph: hits become nodes of the graph
- The graph is transformed to a Reversed Directed Graph (RDGraph)
- GNN is trained using the RDGraph to find track segments

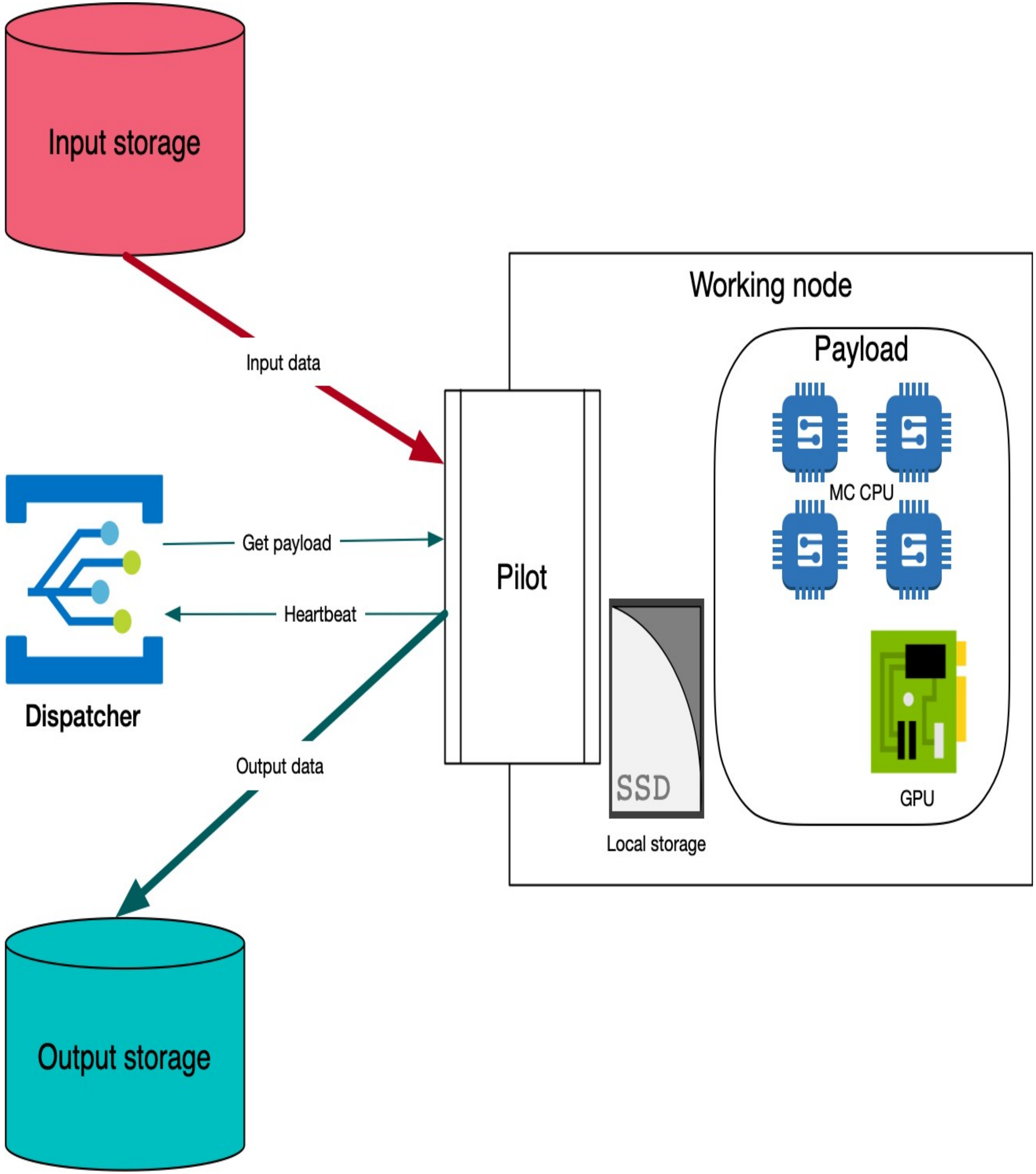
Ososkov G.A. et al., Computer Research and Modeling, 2020, vol. 12, no. 6, pp. 1361-1381

Online filter operation



Main ingredients

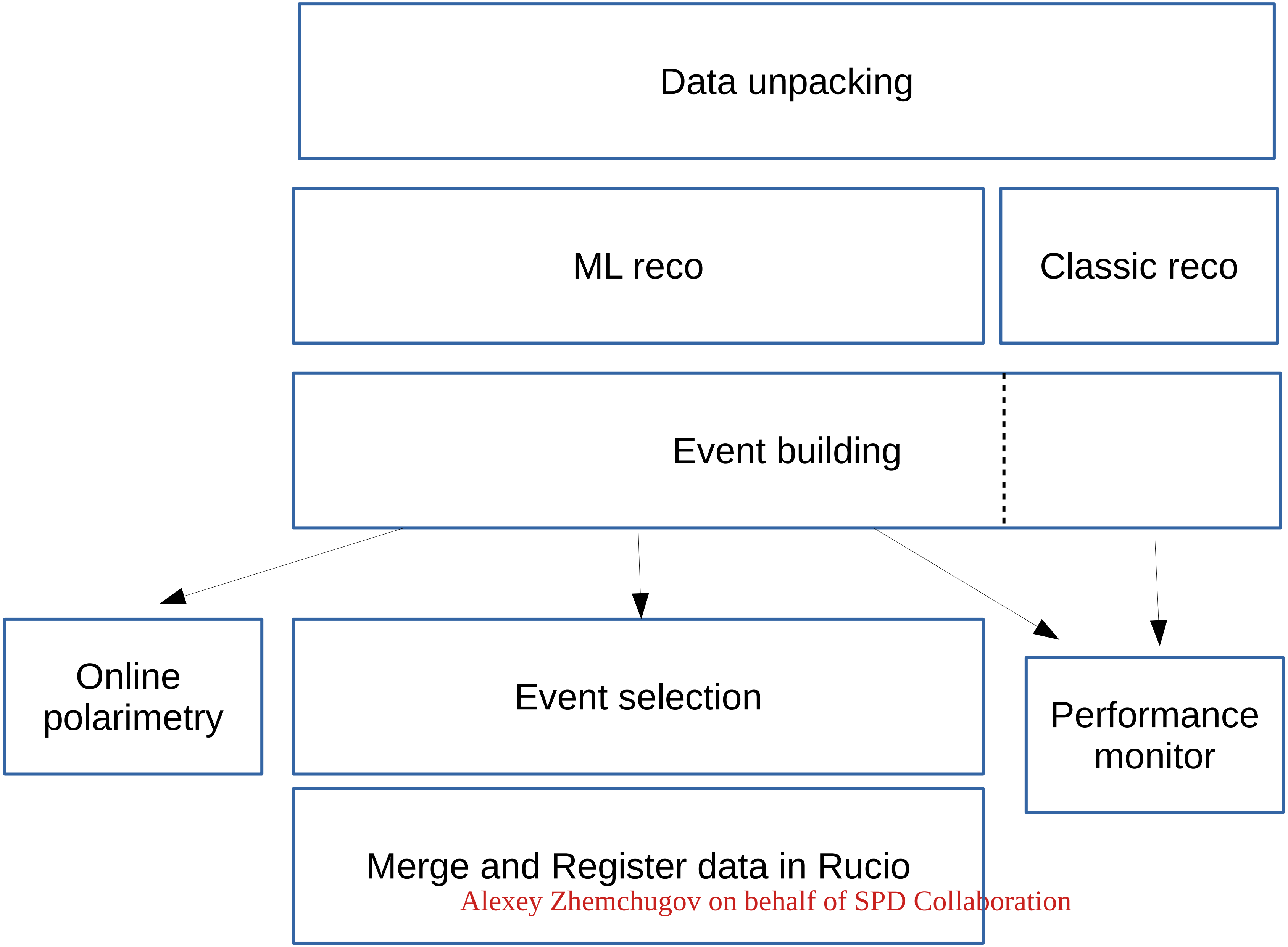
- **Input buffer:** 20 GB/s write, 20 GB/s read, delete 5 files/s
- **Output buffer:** 2x400 MB/s write, 2x400 MB/s read
- **Dispatcher**
- **Identical workers:** multicore nodes with GPUs or FPGA co-processors. 1000 or 5000 WNs ?— depends on the performance of our algorithms!
- *We should foresee using these computing resources for offline data processing between the data taking campaigns*


Input storage
Working node
Payload
Input data
MC CPU
Get payload
Pilot
Heartbeat
SSD
Dispatcher
Output data
GPU
Local storage
Output storage

The pilot

- Constantly runs at a WN
- Communicates with the dispatcher
- Copies data from the input buffer to the WN
- Calls the reconstruction software (ML, classic, merging — depends on the dispatcher’s instruction)
- Copies the resulting file to the output buffer

The payload

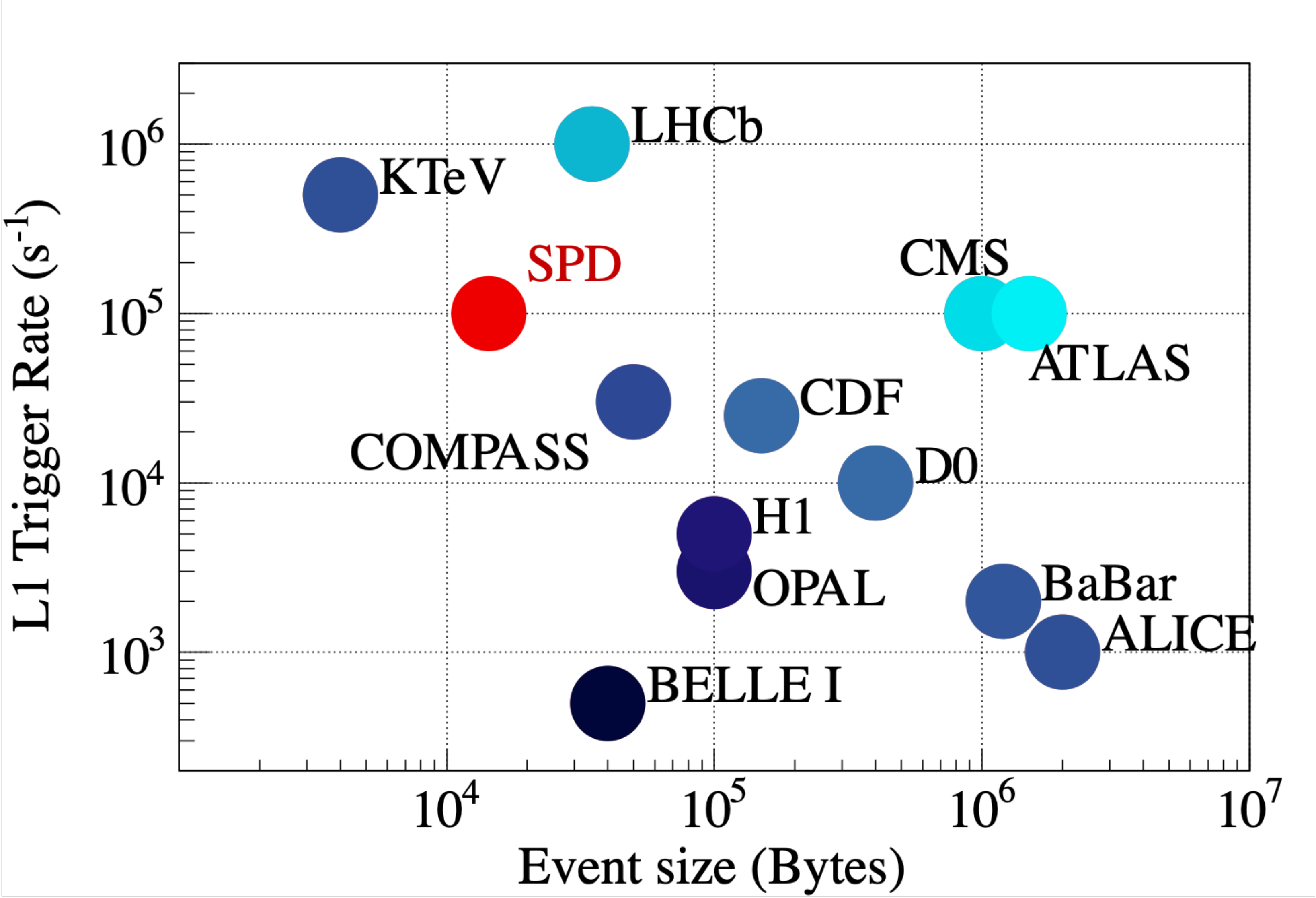

Data unpacking
ML reco
Classic reco
Event building
Online polarimetry
Event selection
Performance monitor
Merge and Register data in Rucio

HDF5 as a data format

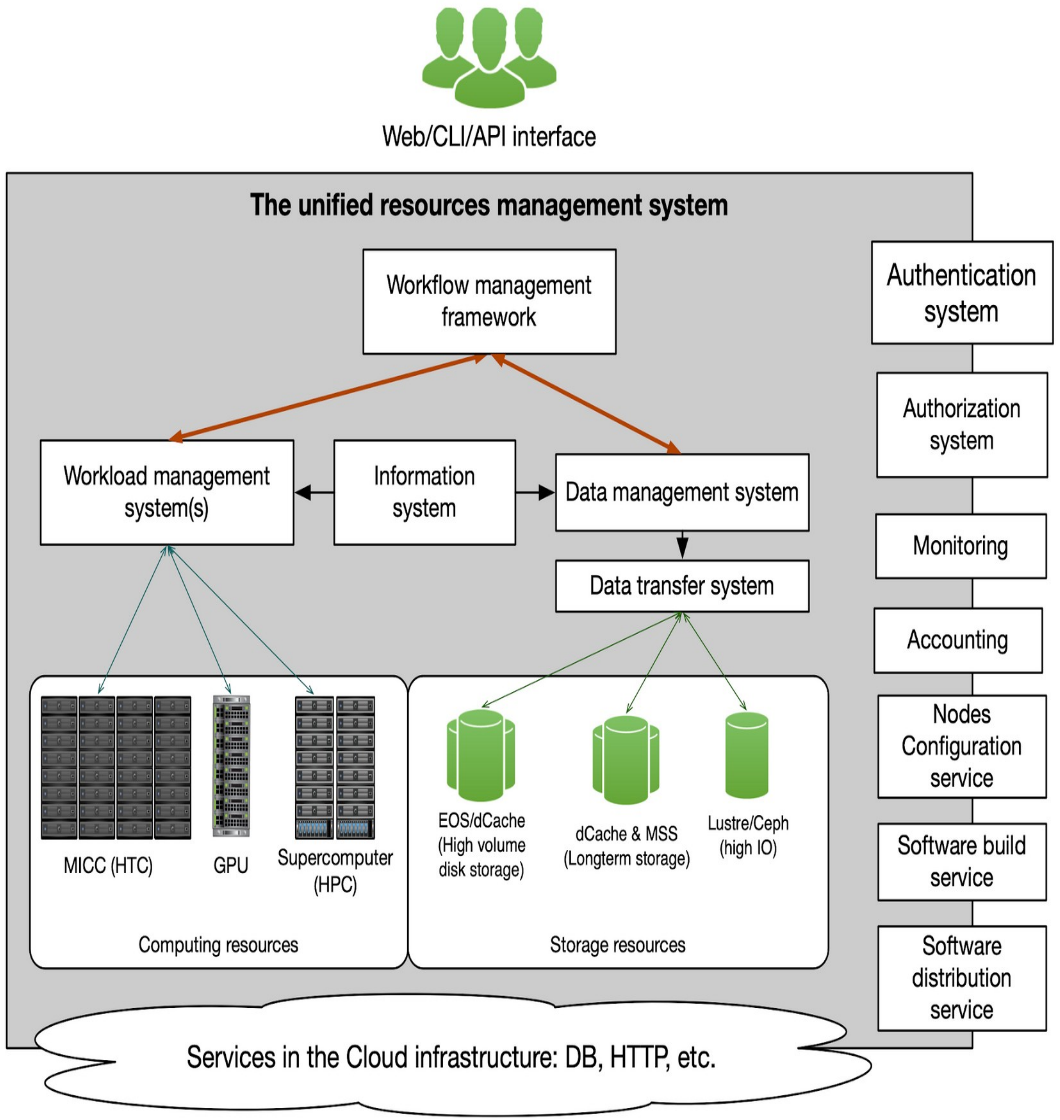
- ROOT is a good format for the current approaches to the data analysis using ROOT
- Less good for the computing system
- Less good for the Python data analysis ecosystem
- Attempts to use HDF5 in FairRoot and Gaudi/Key4HEP (via Podio) were made already

A dedicated R&D to evaluate HDF5 as an intermediate data format for the SPD is needed

After the online filter

- 

Computing system



All basic components are already available from LHC experiments:

- *Workload management: likely PANDA*
- *Data management: RUCIO and FTS*
- *Software distribution: CVMFS*

Adaptation to operate with the SPD event model and offline software is needed

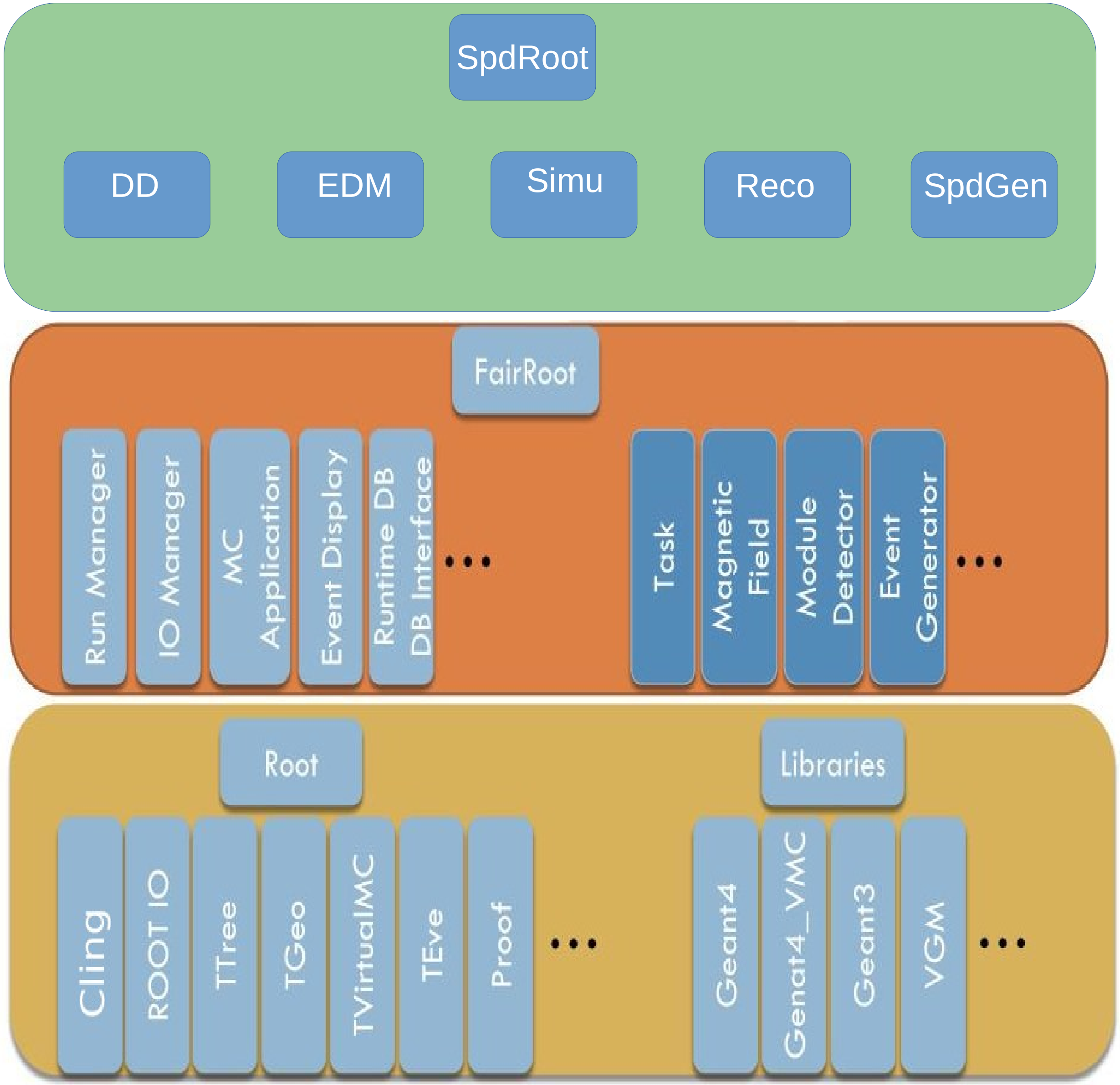
SPD Offline Software

- Core Framework
- Detector Description
- Event Generators
- Simulation
- Reconstruction

Git repository: http://git.jinr.ru/nica/spdroot

Documentation Wiki: https://git.jinr.ru/nica/spdroot/-/wikis/home

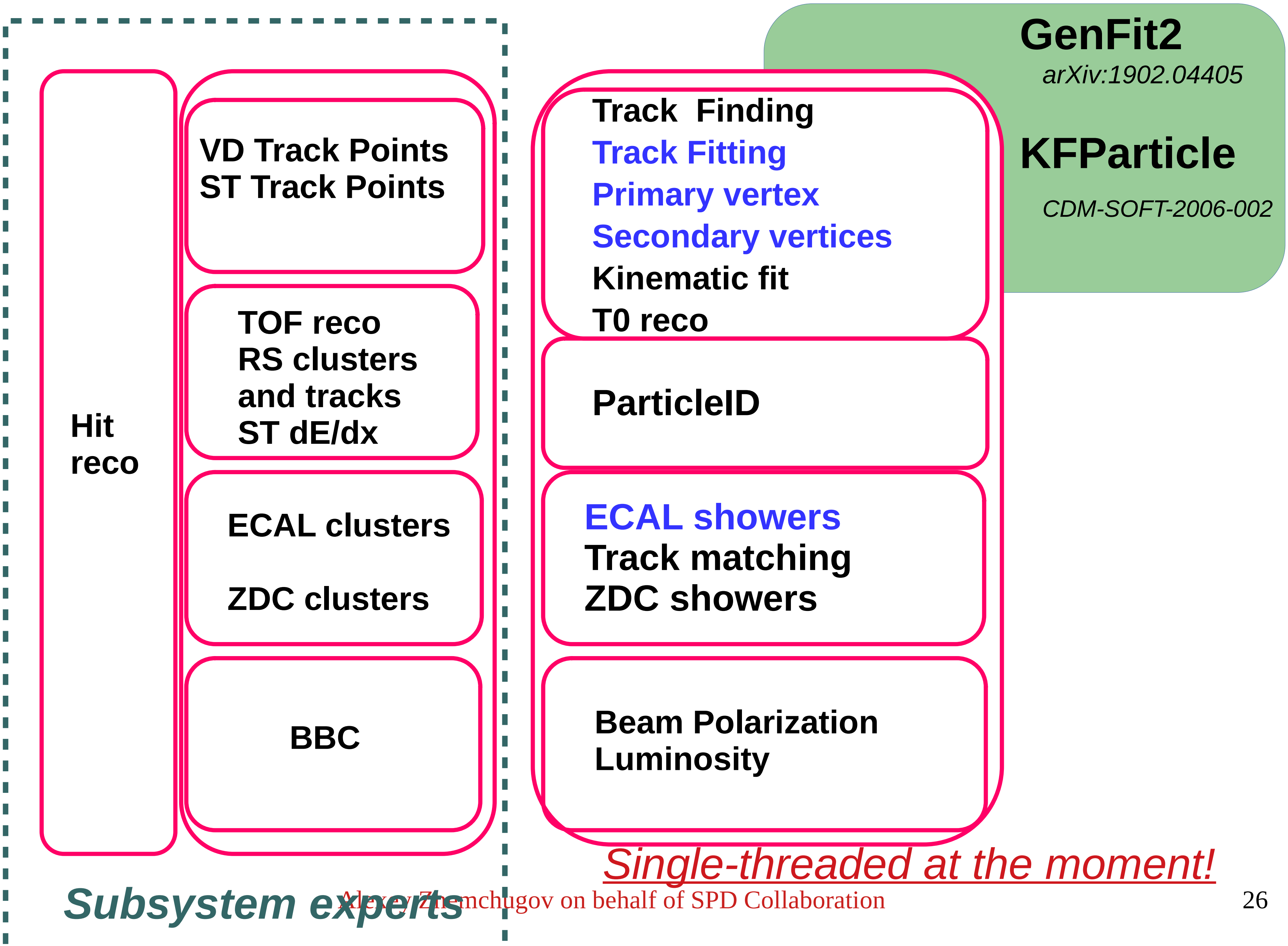
Core Framework: SpdRoot

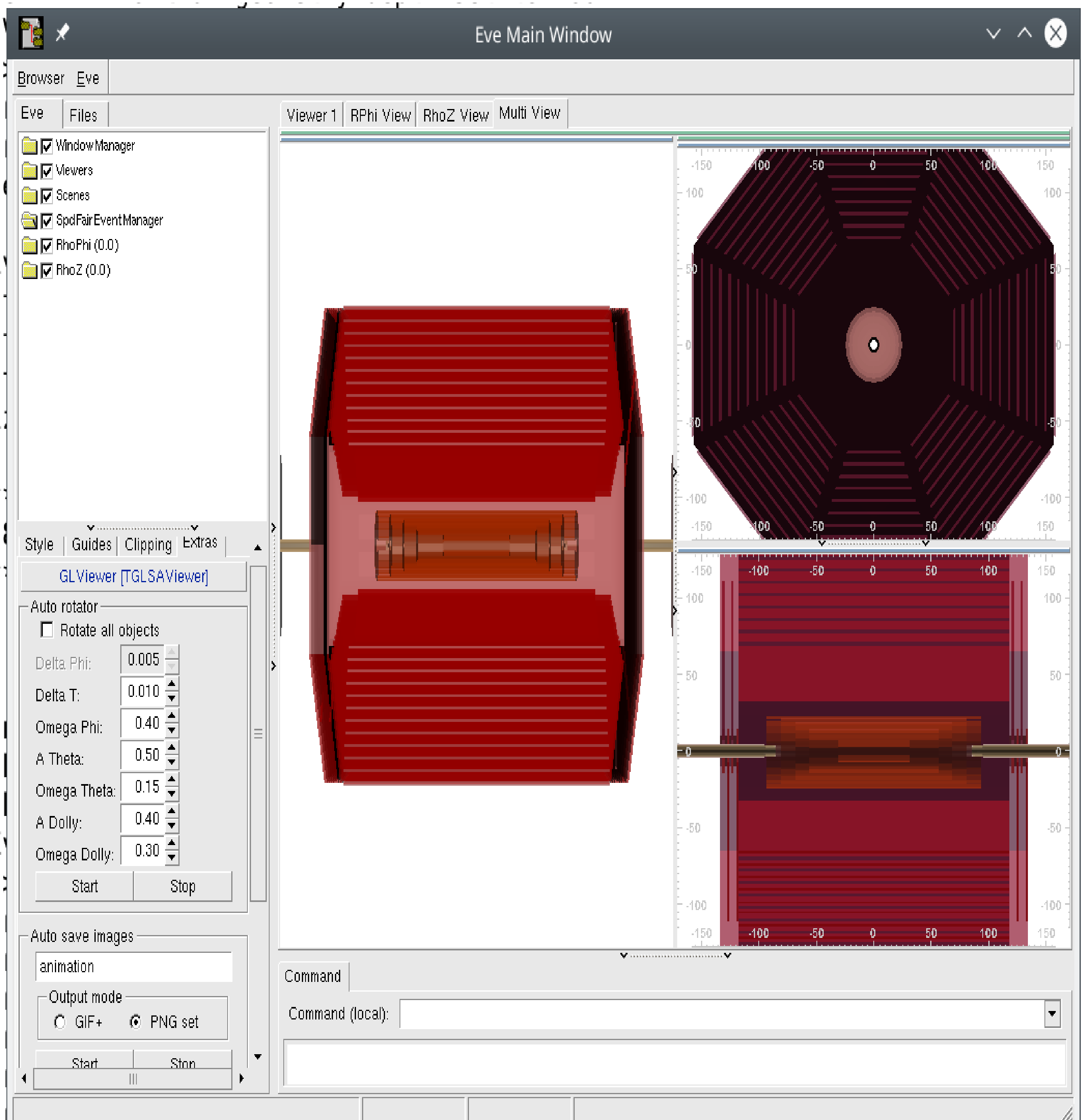


Simulation

- Virtual MC based on Geant4 backend
- Several options for the magnetic field
- Fast simulation exists for:
  - ITS, STS, ECAL, RS
  - No PID, ZDC, BBC ...
- Full simulation is not reasonable until the detector concept is fixed… but we expect it will happen rather soon!
- A lot of work to simulate the real detector, not an ideal one!

Reconstruction




Eve Main Window
Browser Eve
Eve Files
Viewer 1 RPhi View RhoZ View Multi View
WindowManager
Viewers
Scenes
SpdFairEventManager
RhoPhi (0.0)
RhoZ (0.0)
Style Guides Clipping Extras
GLViewer [TGLSAViewer]
Auto rotator
Rotate all objects
Delta Phi: 0.005
Delta T: 0.010
Omega Phi: 0.40
A Theta: 0.50
Omega Theta: 0.15
A Dolly: 0.40
Omega Dolly: 0.30
Start Stop
Auto save images
animation
Output mode
GIF+ PNG set
Start Stop
Command
Command (local):

Software R&D: Multithreading and alternative architectures

Goal: to improve the SPD algorithms and software to be able running at multicore machines and GPU and/or FPGA coprocessors

- Online Filter
- Simulation
- Reconstruction
- Core framework

Software R&D: FairRoot vs Gaudi

Goal: to evaluate Gaudi/Key4HEP as an SPD software framework

Basic functionality of these frameworks is very similar, but ...

	Gaudi/Key4HEP	**FairRoot**
Multithreading and alternative architectures	++	+
Support	+++ (**HSF**, ATLAS, LHCb, FCC)	+ (FAIR, NICA, ALICE?)
Use in real experiments	+++ (ATLAS, BESIII, LHCb)	+ (HADES, BM@N)

Software R&D: Conditions DB, Calib&Align

Goal: to develop a solution to handle geometry, conditions and calibration data

- The Database (10 PB/year ~ O(100000) running jobs)
- Geometry description
- Alignment
- Run info and conditions
- Calibration procedure and constants
- Integration to the computing system

Software R&D: Computing system prototype and a mock-up test

Goal: to demonstrate that the computing system is capable to handle the SPD data rate

- Information system
- Data management
- Task management
- Working prototype and a mock-up test of O(1 PB) scale (like a simulation of 1E9 events)
- Samara University's cluster "Sergey Korolev" is interested to join

Software & Computing R&D in 2021

- ML-based event reconstruction and an Online Filter prototype
- HDF5 as a data format
- Multithreading and alternative architectures
- FairRoot vs Gaudi
- Conditions DB, Calib&Align
- Computing system prototype and a mock-up test

Summary

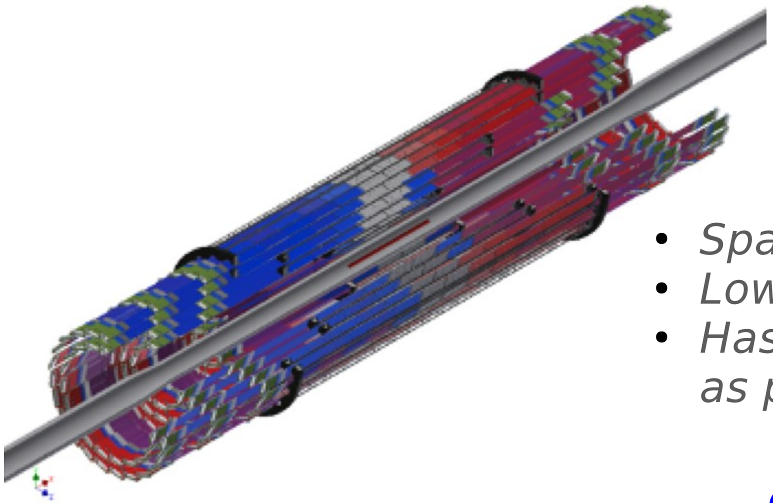
- Preparation of the SPD experiment is making good progress.
- The SPD detector is a medium scale setup in size, but a large scale one in data rate. That poses a significant challenge both to the computing system and offline software.
- Efficient online filter and distributed computing system are the two keys to the success of SPD data processing.
- A lot of interesting problems to be solved. A lot of MSc/PhD theses to be prepared
- The main goal of the software project in 2021: TDR preparation
  - Software and computing infrastructure for simulation and reconstruction for physics studies
  - (Extremely interesting) R&D for the chapter 'Computing and Offline Software' for the TDR

Tracking system

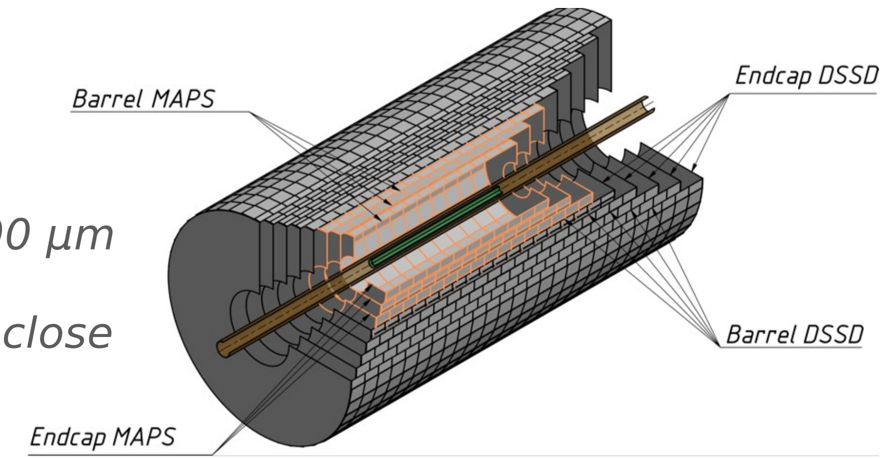
Vertex detector

two options:

5 layers of DSSD

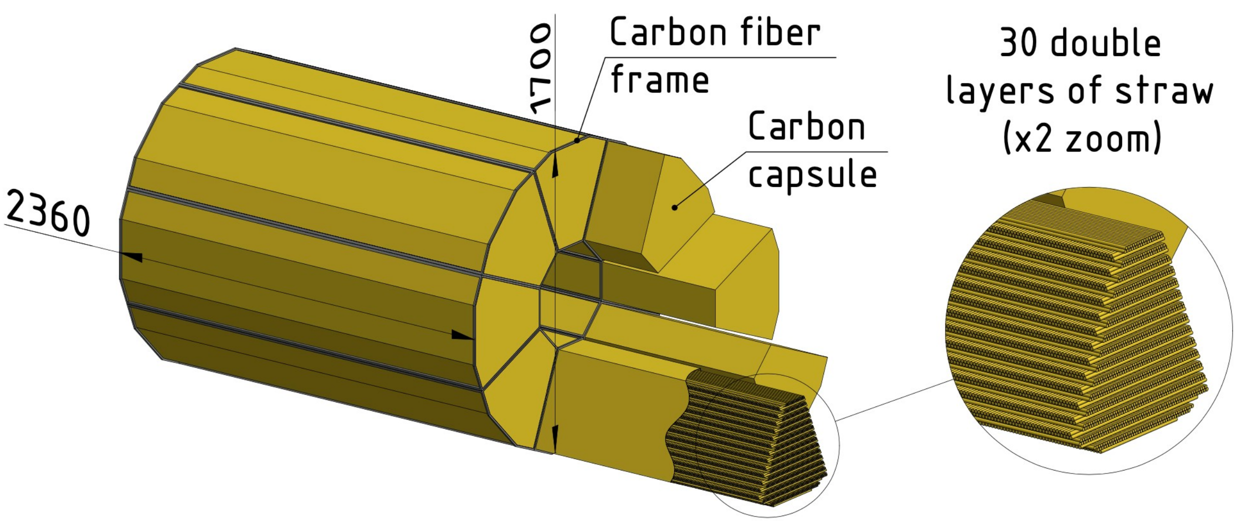


3 layers of MAPS + 2 layers of DSSD



- Spatial resolution <100 μm
- Low material budget
- Has to be installed as close as possible to the IP

Straw tracker



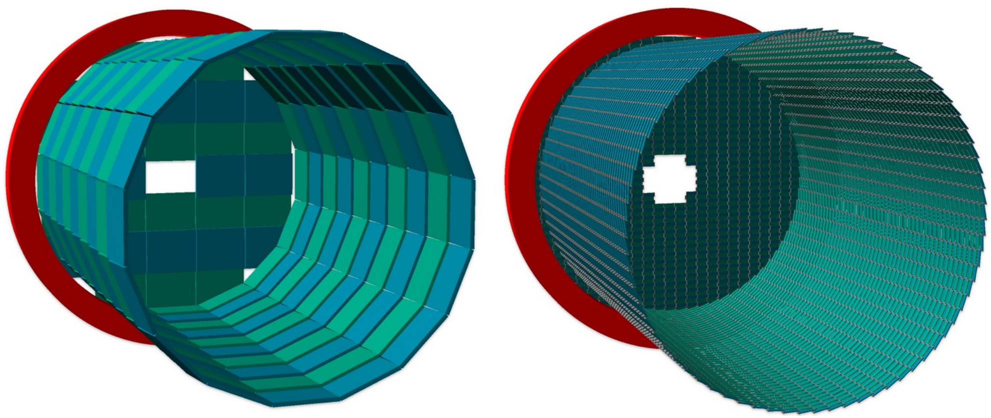
- Spatial resolution ~150 μm
- Low material budget
- Operation in magnetic field of about 1 T
- dE/dx measurement

Particle identification system

TOF system

Aerogel-based PID

mRPC-based *Scintillator-based*

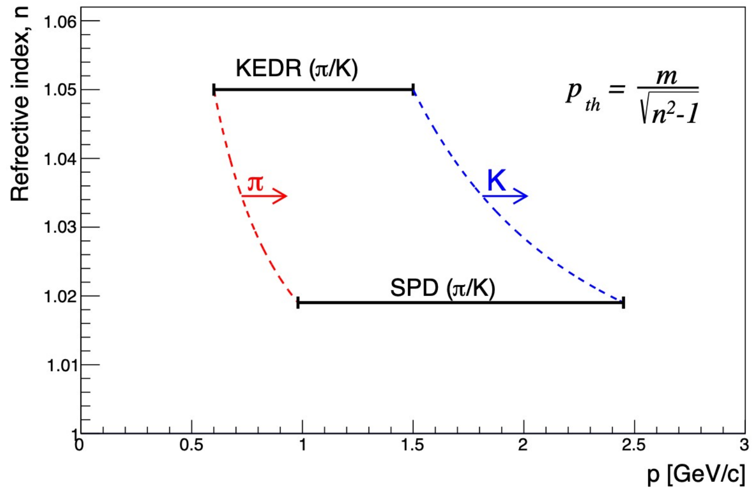


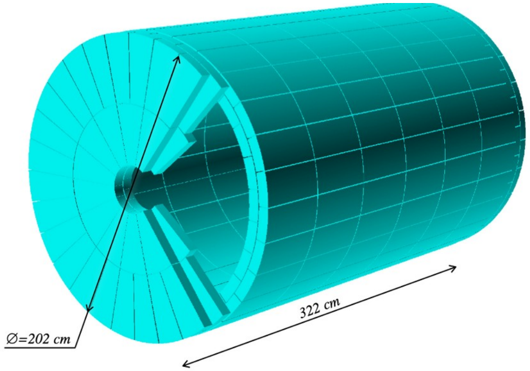
Goals:

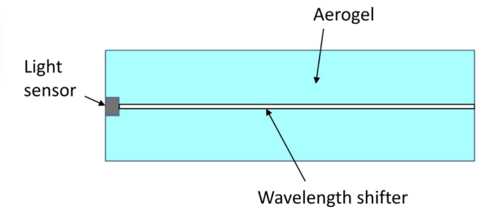
- *π/K separation up to ~1.5 GeV*
- *K/p separation*
- *t_0 determination*

Requirements:

- *Time resolution ~60-70 ps*







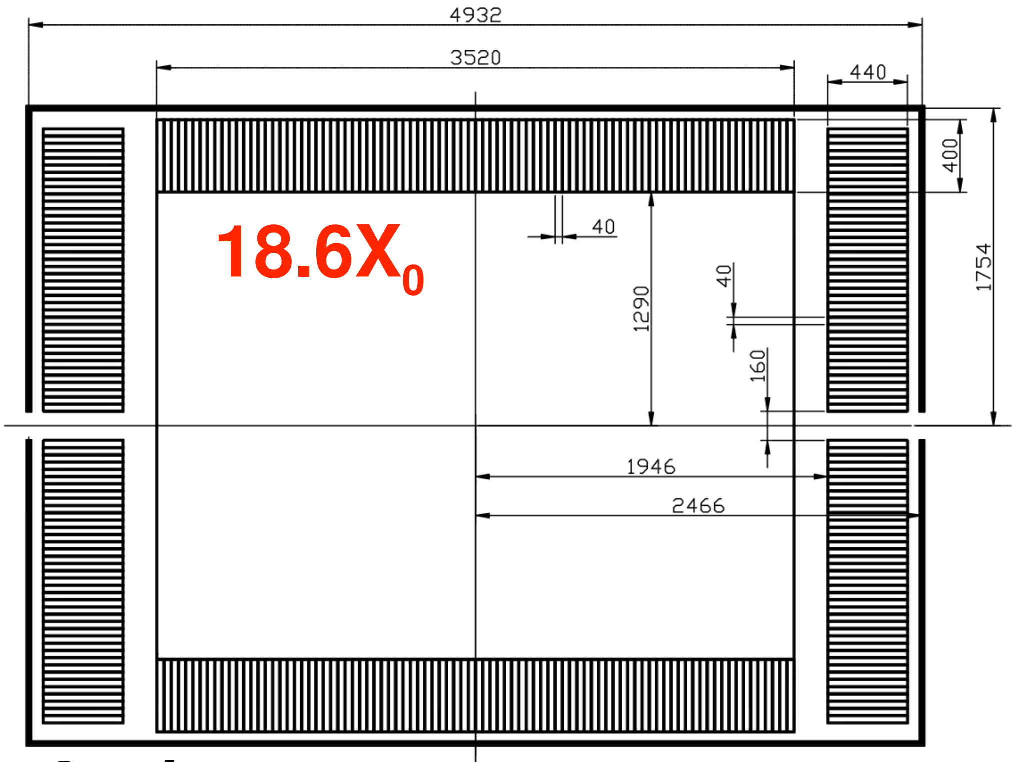
Goals:

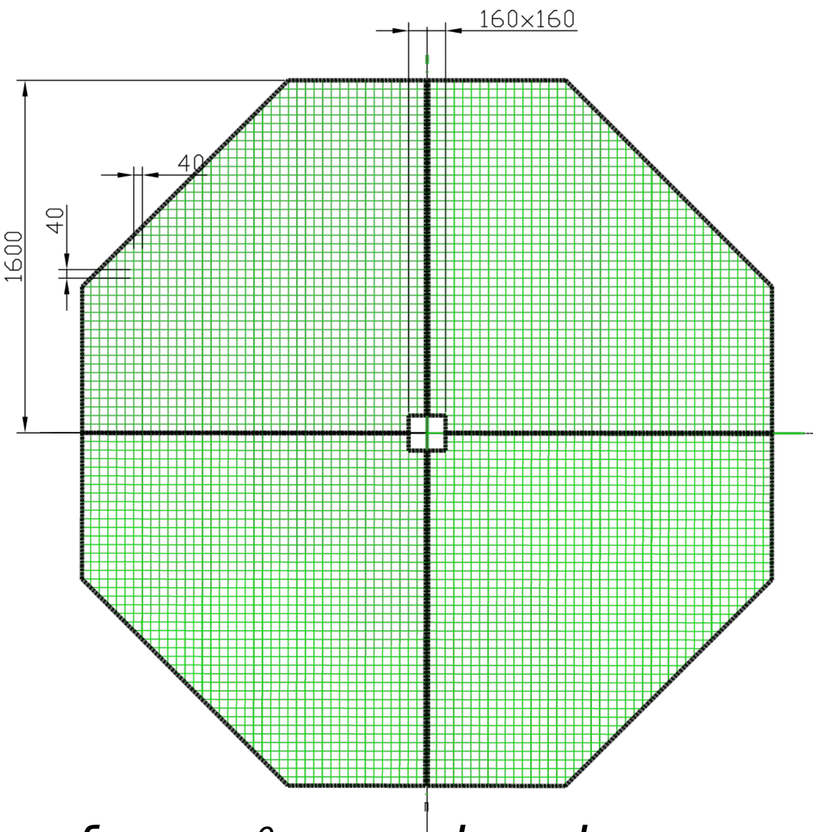
- *π/K separation up to 2.5 GeV range*

Requirements:

- *We should have enough light!*

Electromagnetic calorimeter





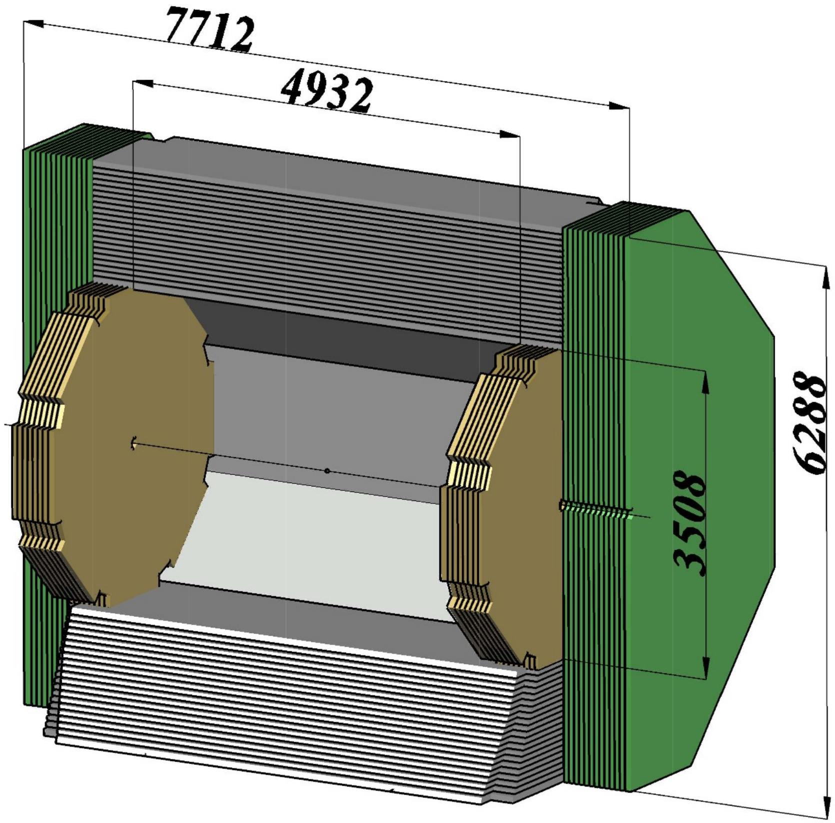
Goals:

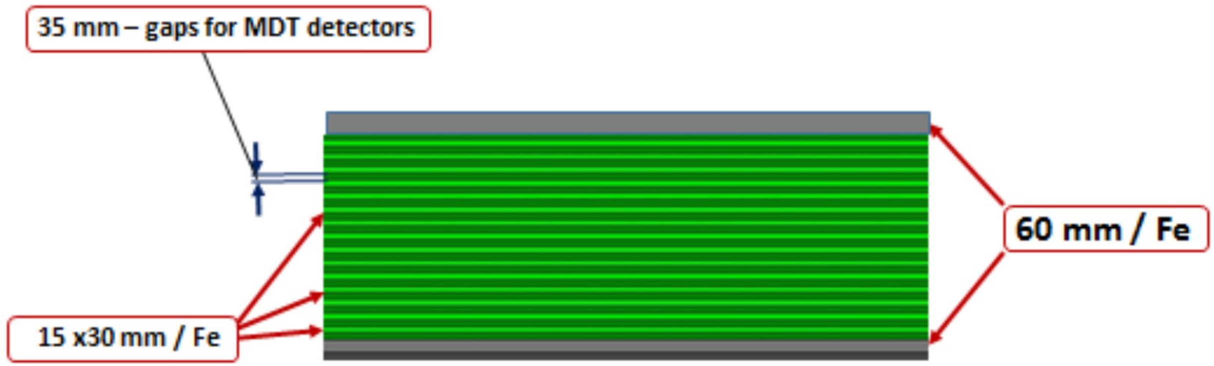
- *Detection of prompt photons, photons from π^0, η and χ_c decays*
- *Identification of electrons and positrons, participation in muon identification*

Requirements:

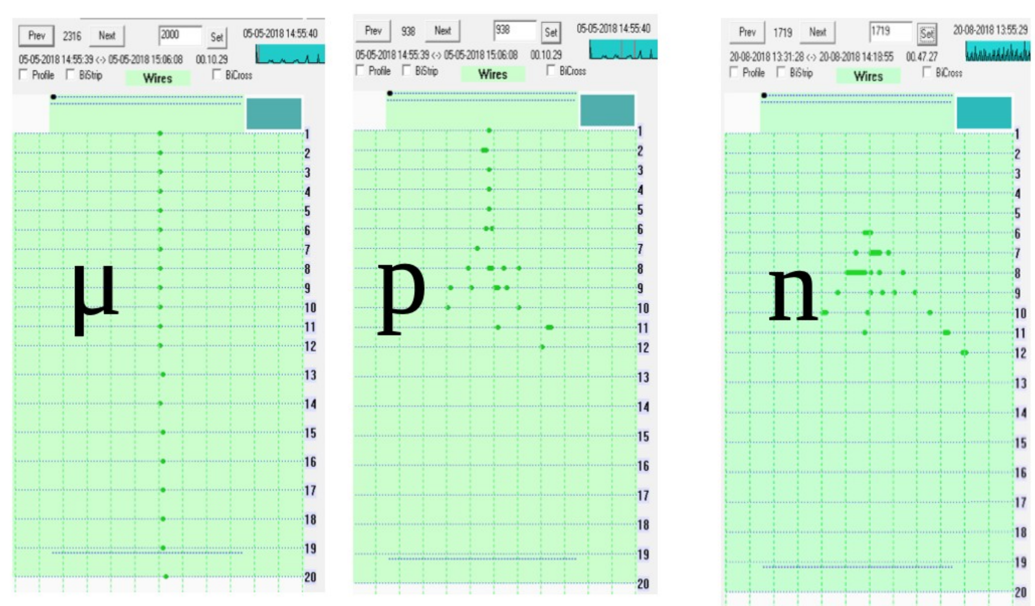
- *Granularity ~4 cm*
- *Low energy threshold (~50 MeV)*
- *Energy resolution*

Range (muon) system





Event examples at 5 GeV/c



Goals:

- *Muon identification*
- *Rough hadron calorimetry*

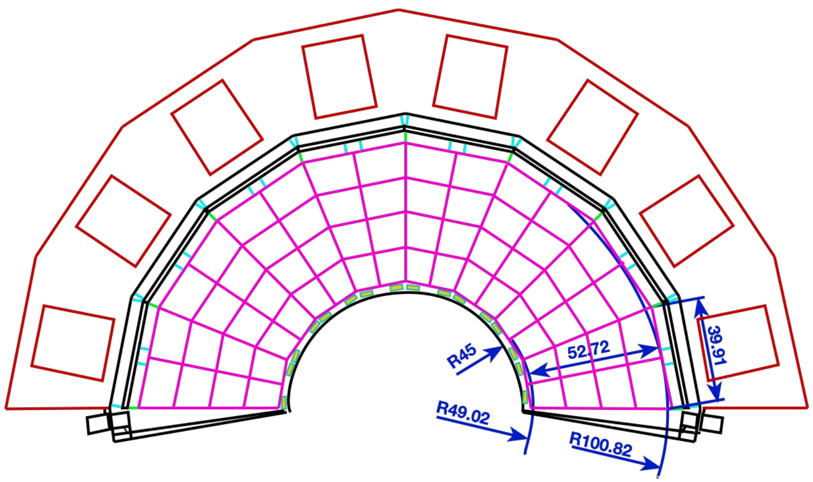
Requirements:

- *should have at least 4λ_I*

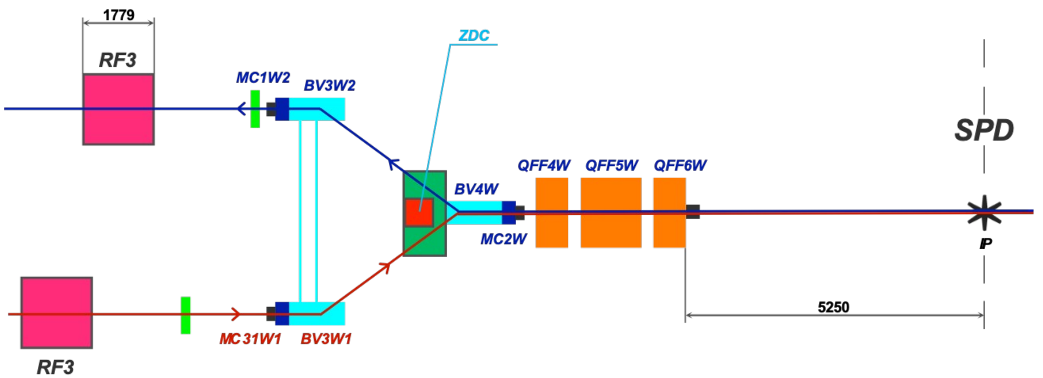
Local polarimetry and luminosity control

- *Charged particles in BBC*
- *π^0 in the end-cap part of ECAL*
- *Neutrons in ZDC*

Beam-Beam Counter



Zero Degree Calorimeter



***Requirements**:*

- *Operation inside the beam pipe (inner part)*
- *Time resolution ~1 ns (inner) and ~400 ps (outer part)*