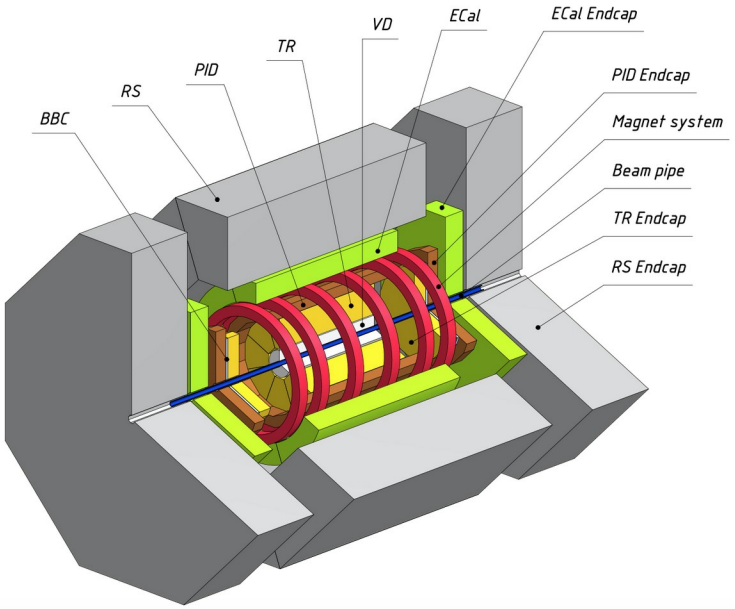
Offline Software and Computing for the SPD experiment

Alexey Zhemchugov
JINR

SPD Collaboration Meeting
8 June 2021

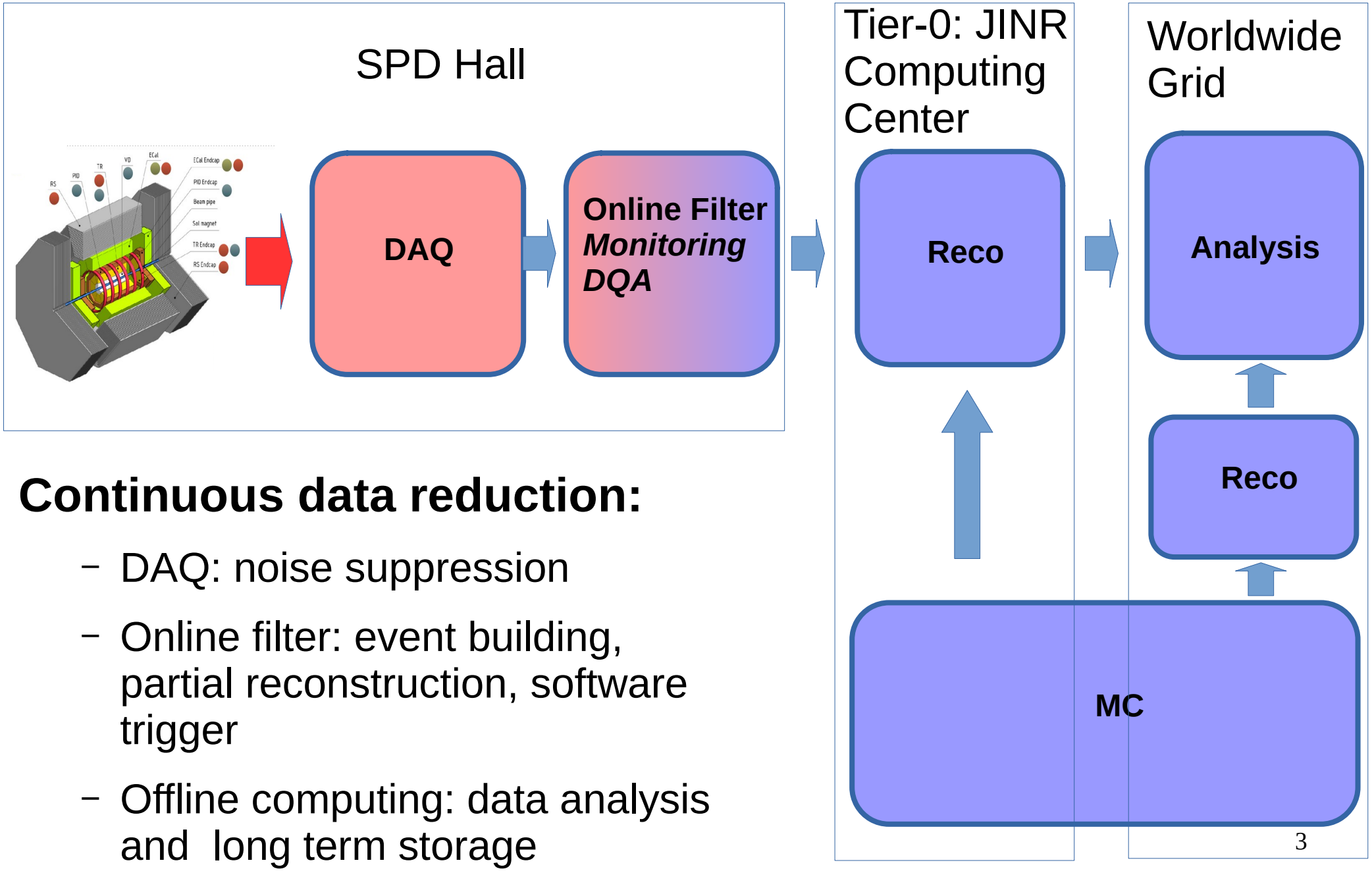
SPD as a data source



- 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

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

Data workflow



Continuous data reduction:

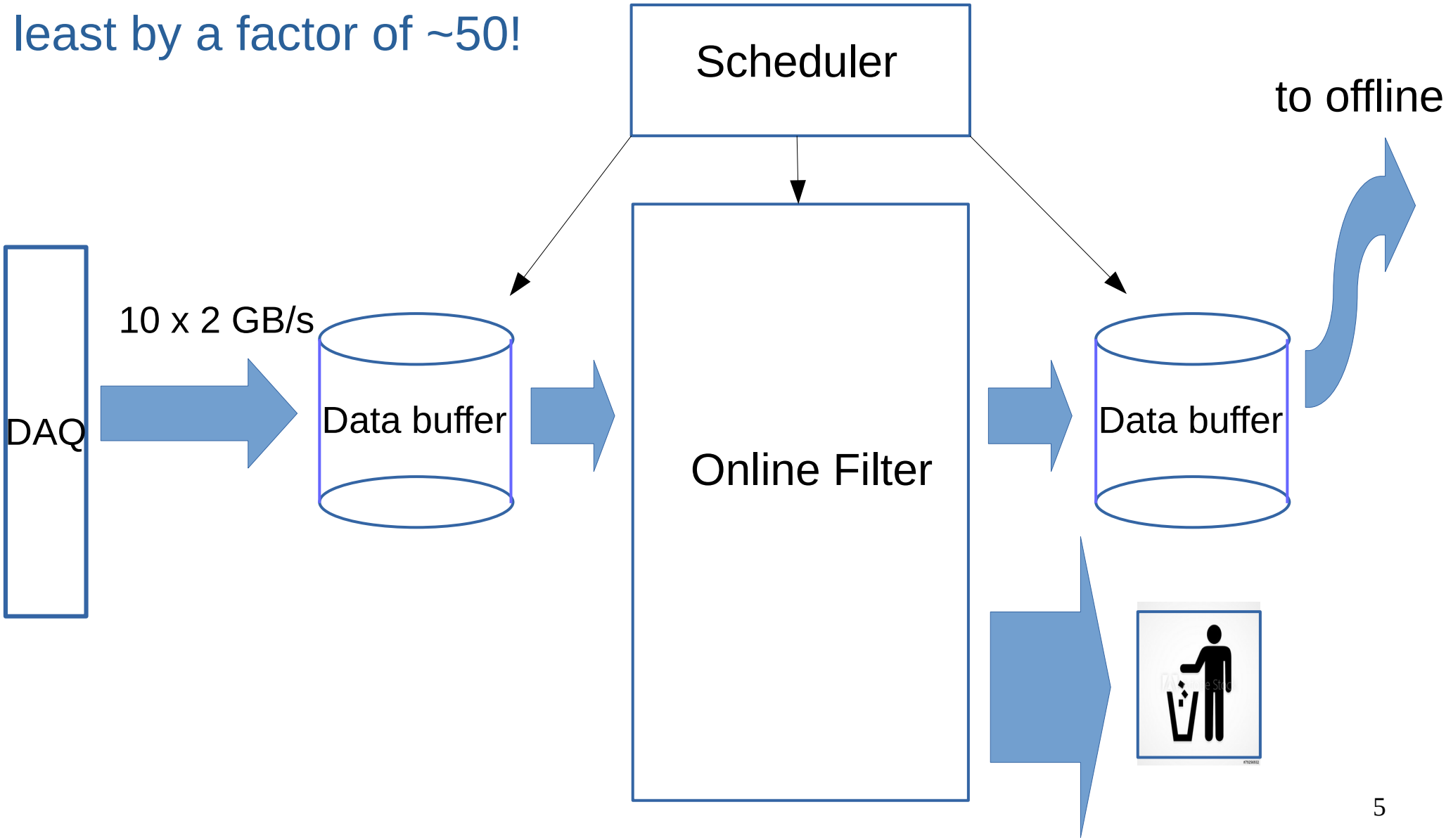
- DAQ: noise suppression
- Online filter: event building, partial reconstruction, software trigger
- Offline computing: data analysis and long term storage

Free-running DAQ

- 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
- No global trigger => No classical events anymore. Event building have to unscramble events from a series of time slices.
- *More details in the talk of Leonid Afanasyev on Thursday*

Online Data Filter

We need a data reduction at least by a factor of ~50!



Online Data Filter ingredients

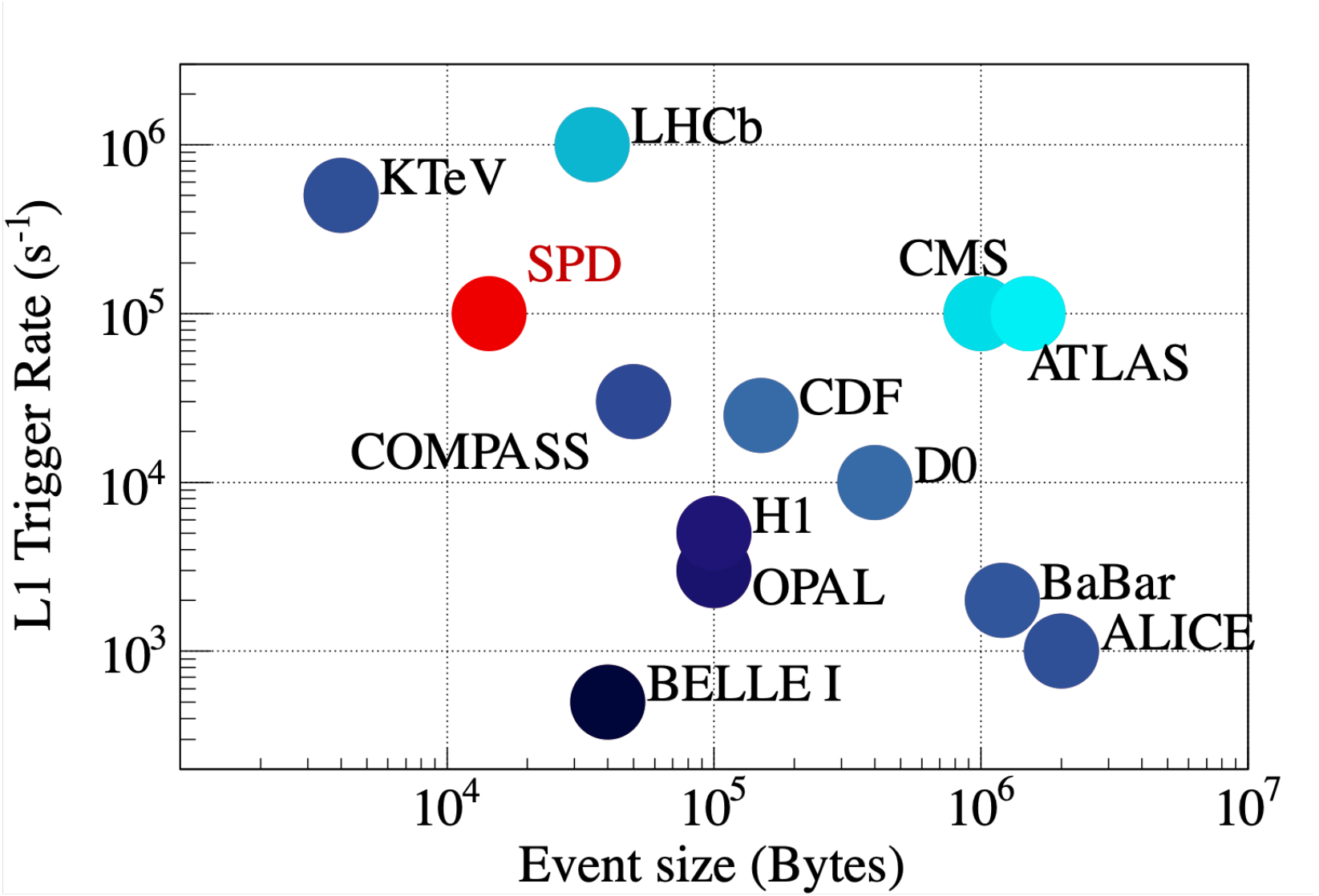
High-performance heterogeneous computing cluster

- Partial reconstruction
  - Fast tracking and vertex reconstruction — *more details in the talk of Egor Shchavelev tomorrow*
  - Fast ECAL clustering — *more details in the talk of Dimitrije Maletic tomorrow*
- Event unscrambling
- Software trigger
  - several data streams

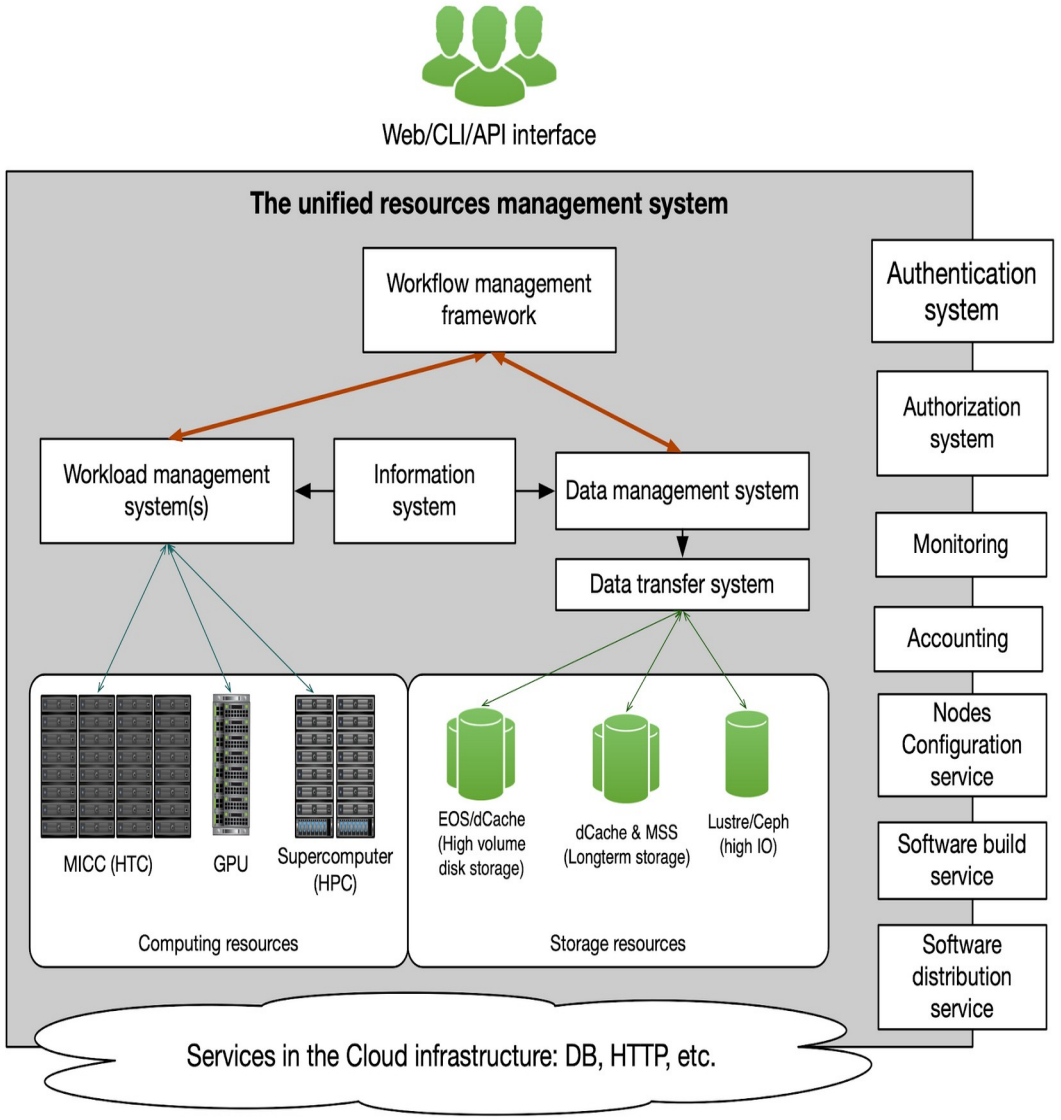
Machine learning is a key technology

- Monitoring and Data quality assessment
- Local polarimetry
- *More details in my talk tomorrow*

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
- *More details in the talk of Artem Petrosyan tomorrow*

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

SPD Offline Software

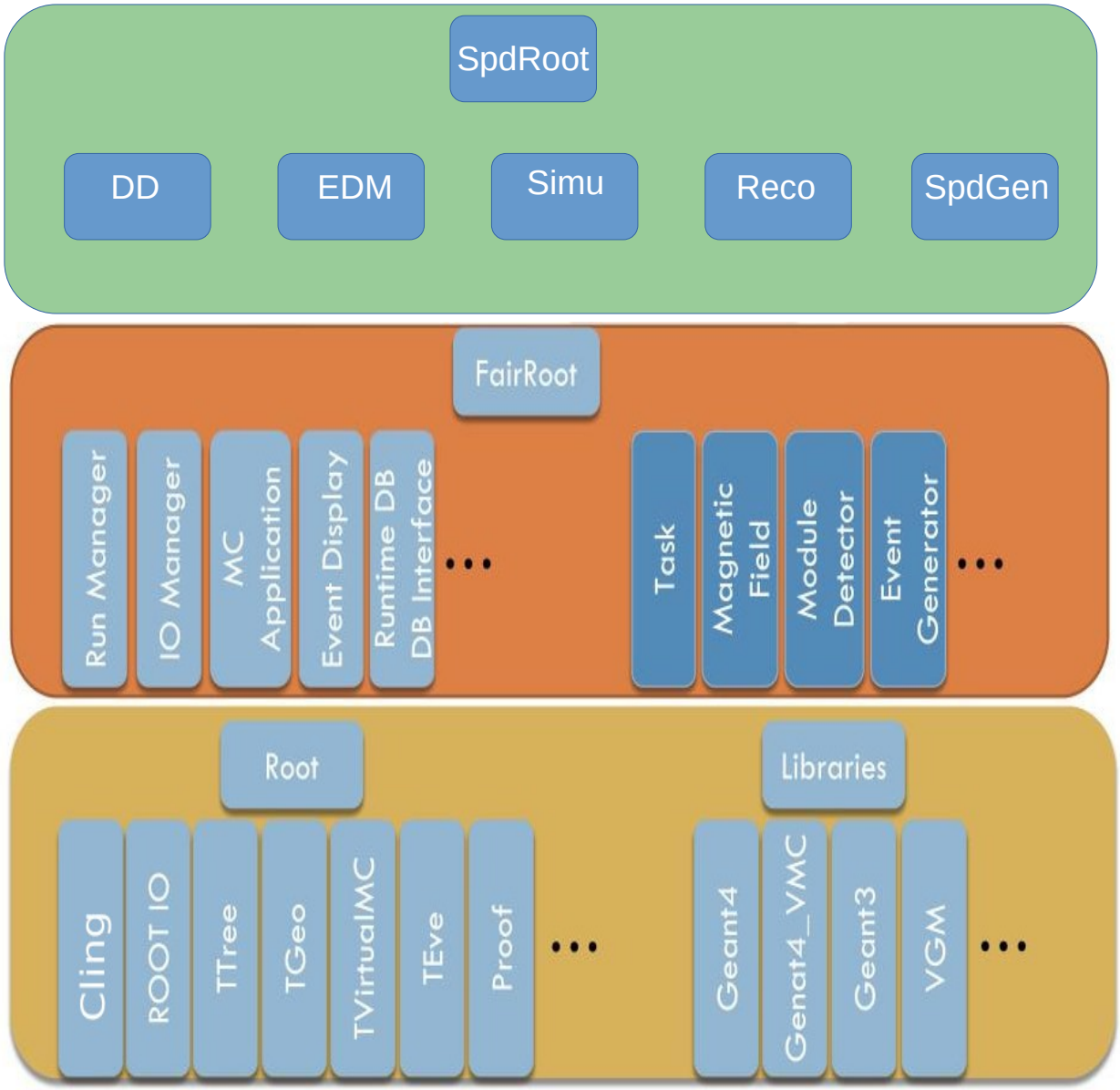
- Core Framework
- Detector Description
- Event Generators
- Simulation
- Reconstruction

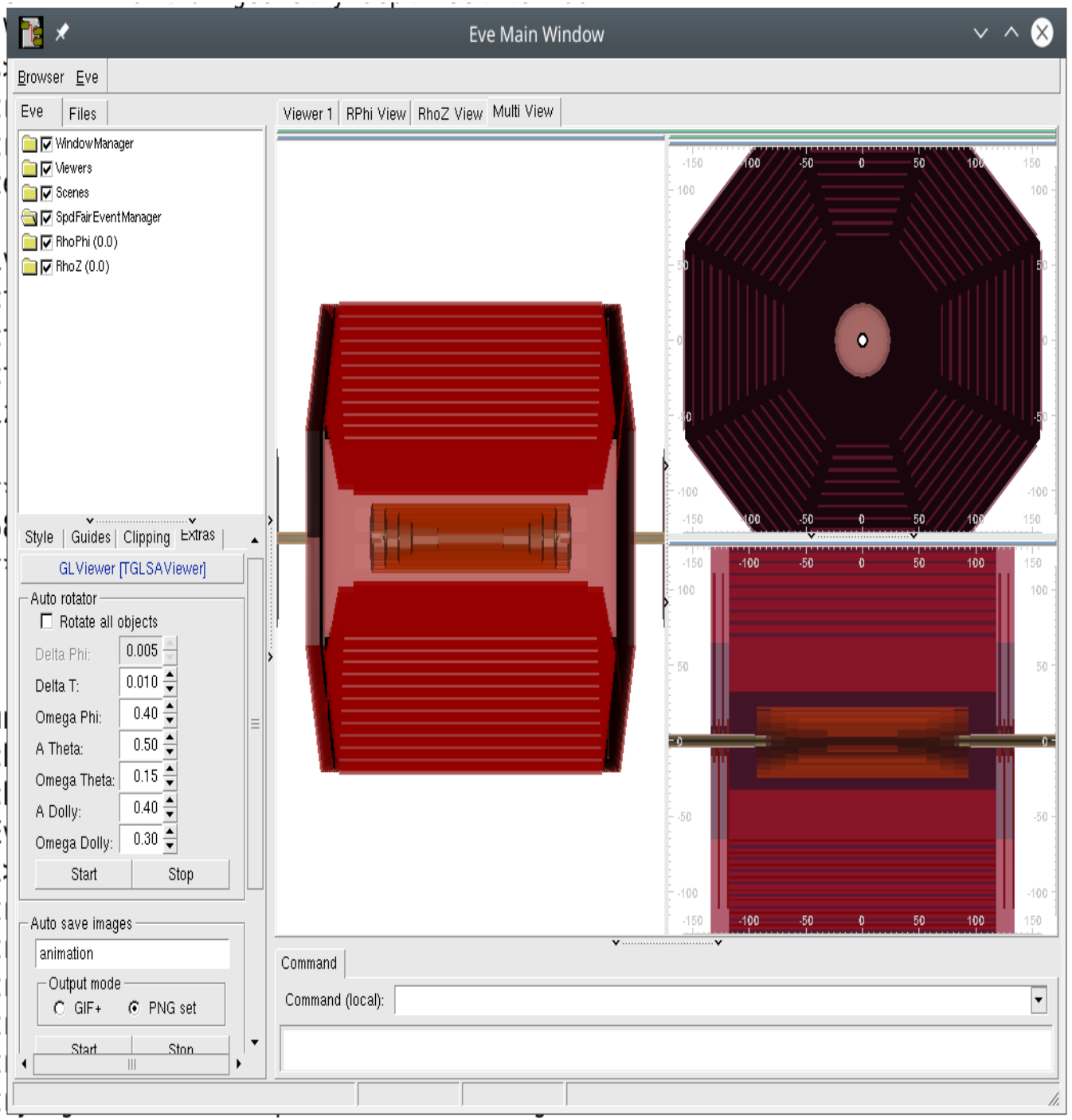
Git repository: http://git.jinr.ru/nica/spdroot

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

More details in the talk of Artur Tkachenko tomorrow

Core Framework: SpdRoot

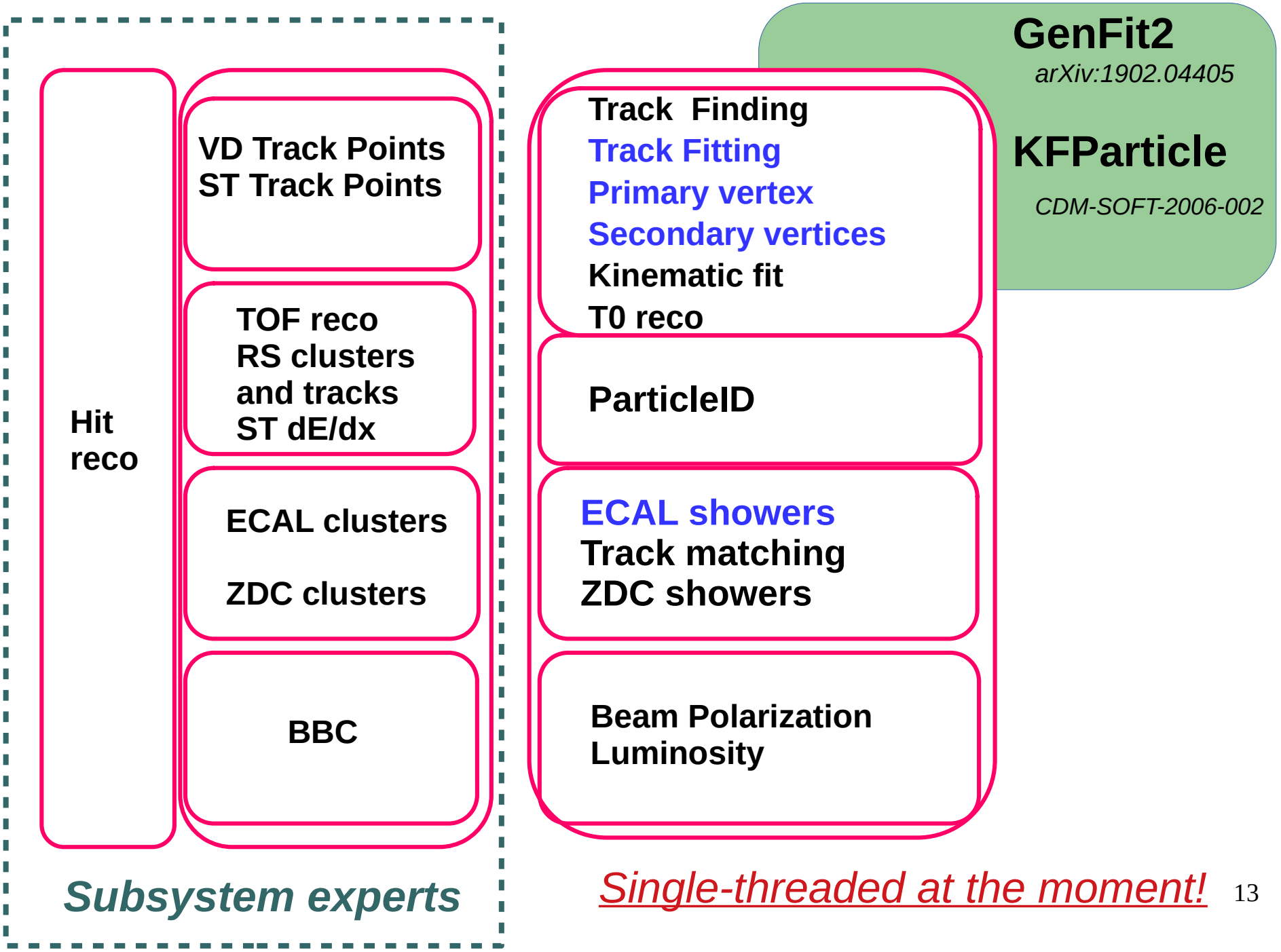



Eve Main Window
Browser
Eve
Eve
Files
WindowManager
Viewers
Scenes
SpdFairEventManager
RhoPhi (0.0)
RhoZ (0.0)
Style
Guides
Clipping
Extras
GLViewer [TGLSAViewer]
Auto rotator
Rotate all objects
Delta Phi: 0.005
Delta T: 0.010
Omega Phi: 0.40
A Theta: 0.50
Omega Theta: 0.15
A Dolly: 0.40
Omega Dolly: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

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



Software & Computing R&D in scope of the TDR preparation

- 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

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

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

Summary

- Preparation of the SPD experiment is making a 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 the offline software.
- An efficient online filter and a distributed computing system are the two keys to the success of the SPD data processing.
- 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

You are welcome to join any of (or all) these activities!