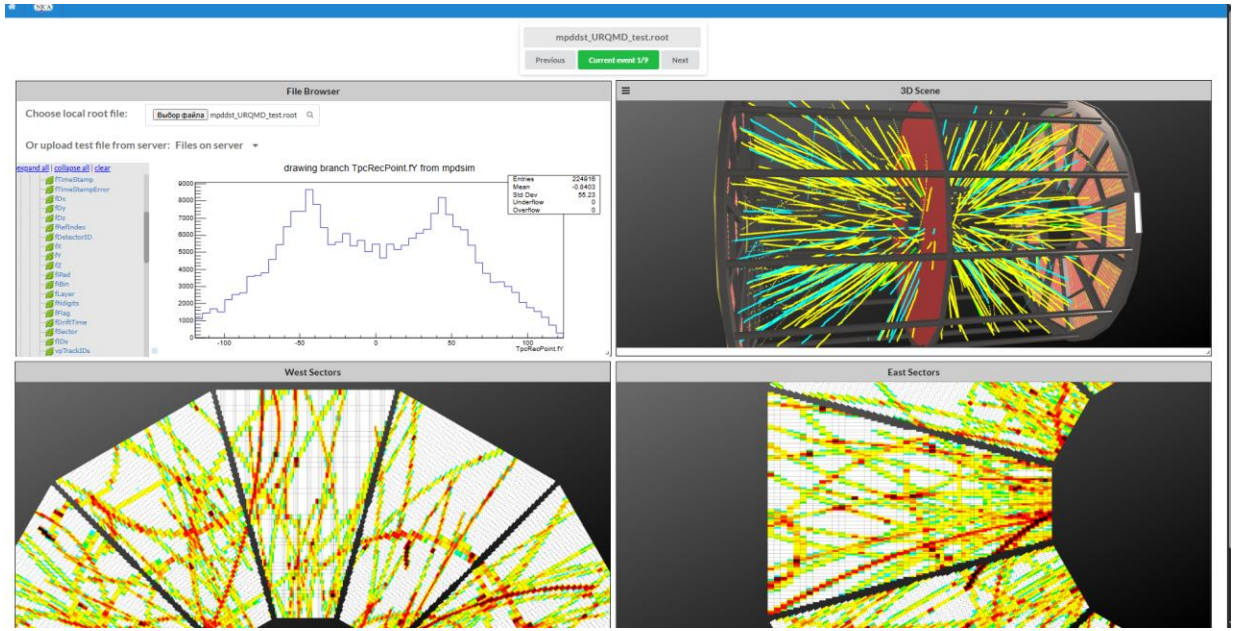
TPC detector DST event viewer

ALEXANDER KRYLOV (LHEP)
EMAIL: AVKRYLOV@JINR.RU
OLEG ROGACHEVSKY (LHEP)
VIKTOR KRYLOV (LNP)

Introduction

There are various methods for monitoring the engineering, network and computer systems of the experiment. As a rule, they have a common name for all - **Event Display** . The operator of the facility must receive comprehensive information about the progress of the experiment in a clear and intuitive form.



Introduction

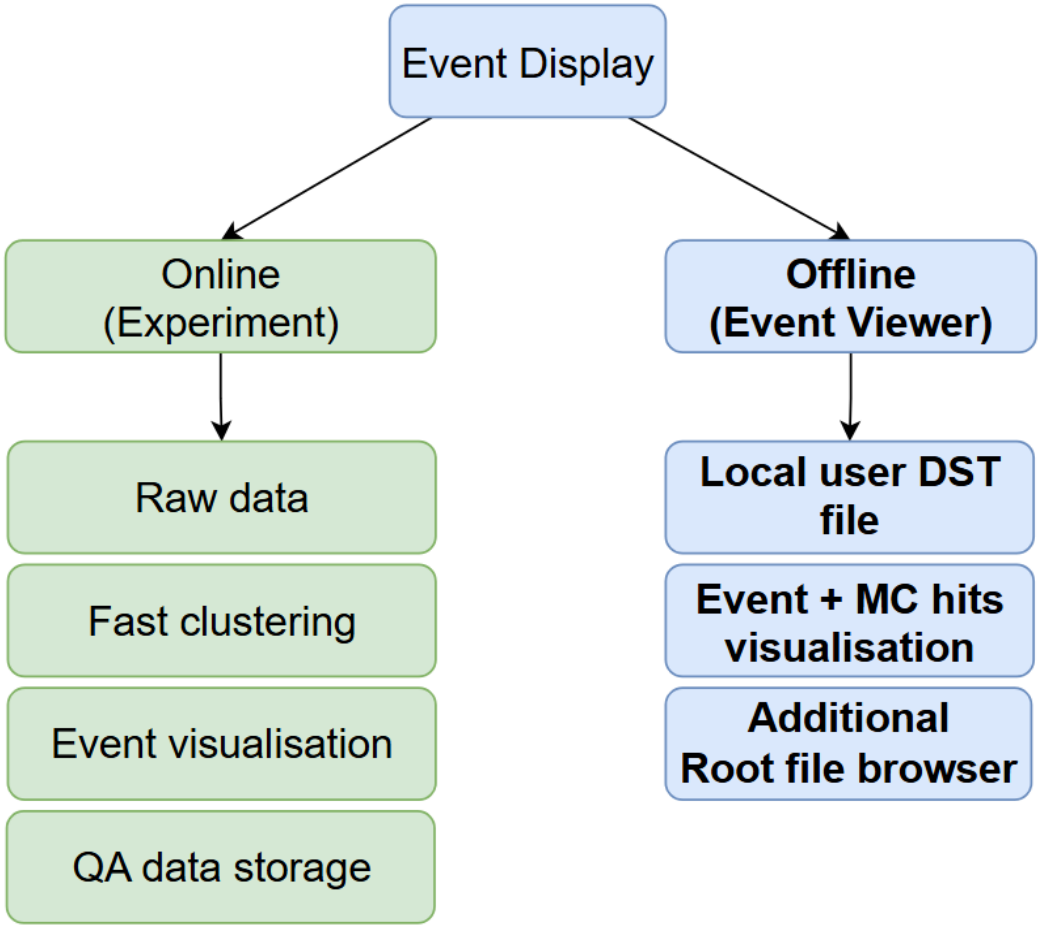
Event Display will work during experiment and also in offline mode.

Experiment (online):

- Raw experiment data as input
- Fast clustering
- Current event visualisation
- Storting QA data histograms

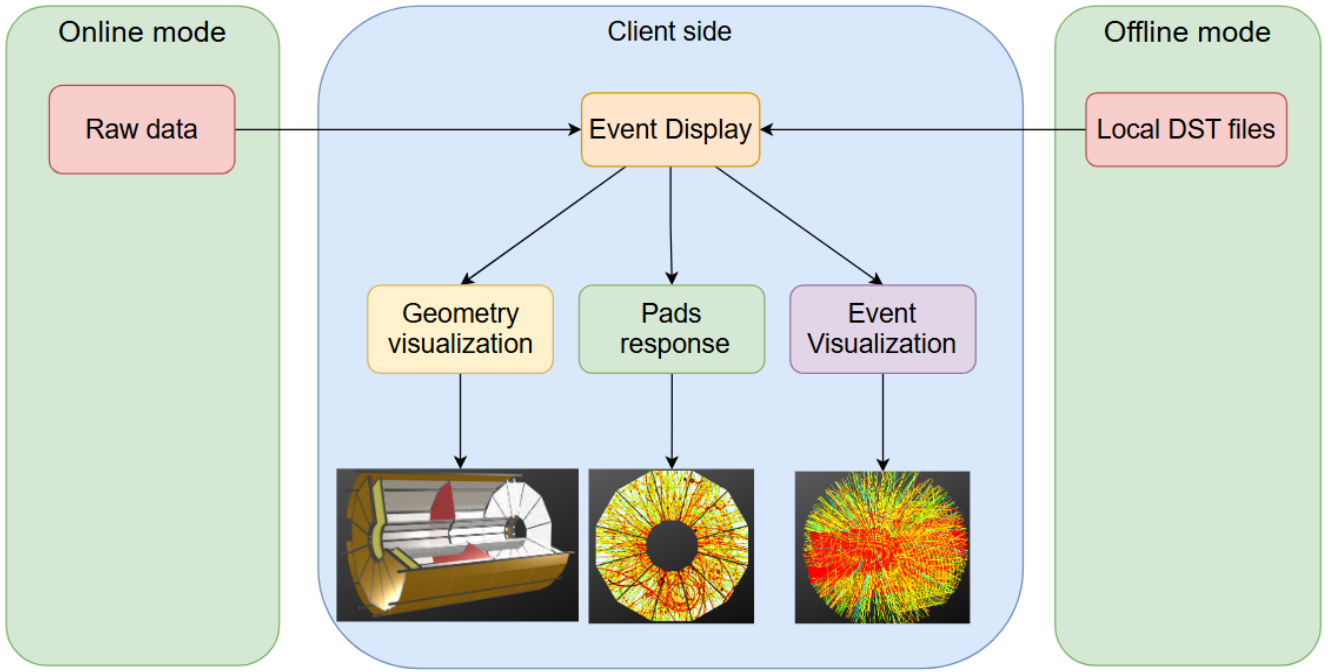
Offline:

- Local DST files as input
- Choosed event visualisation
- Additional Root File Browser

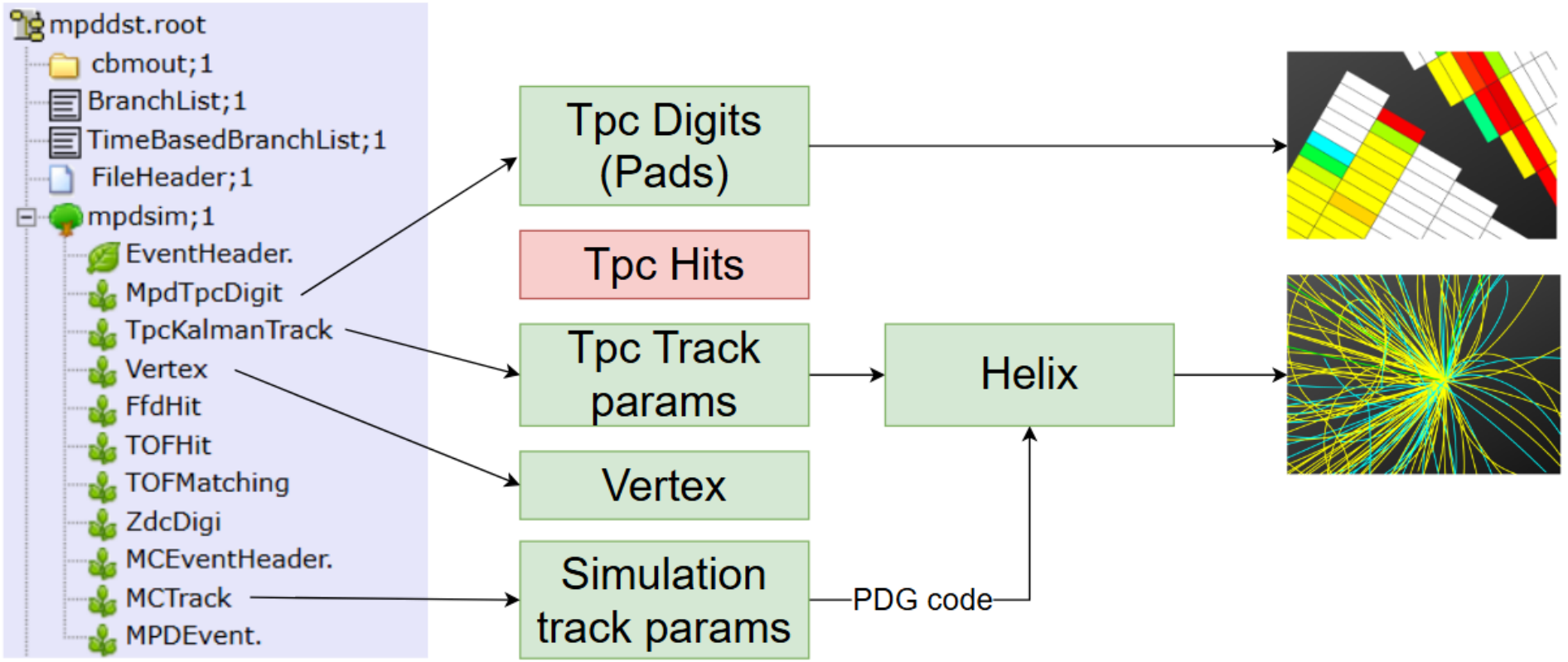


Event Display client side

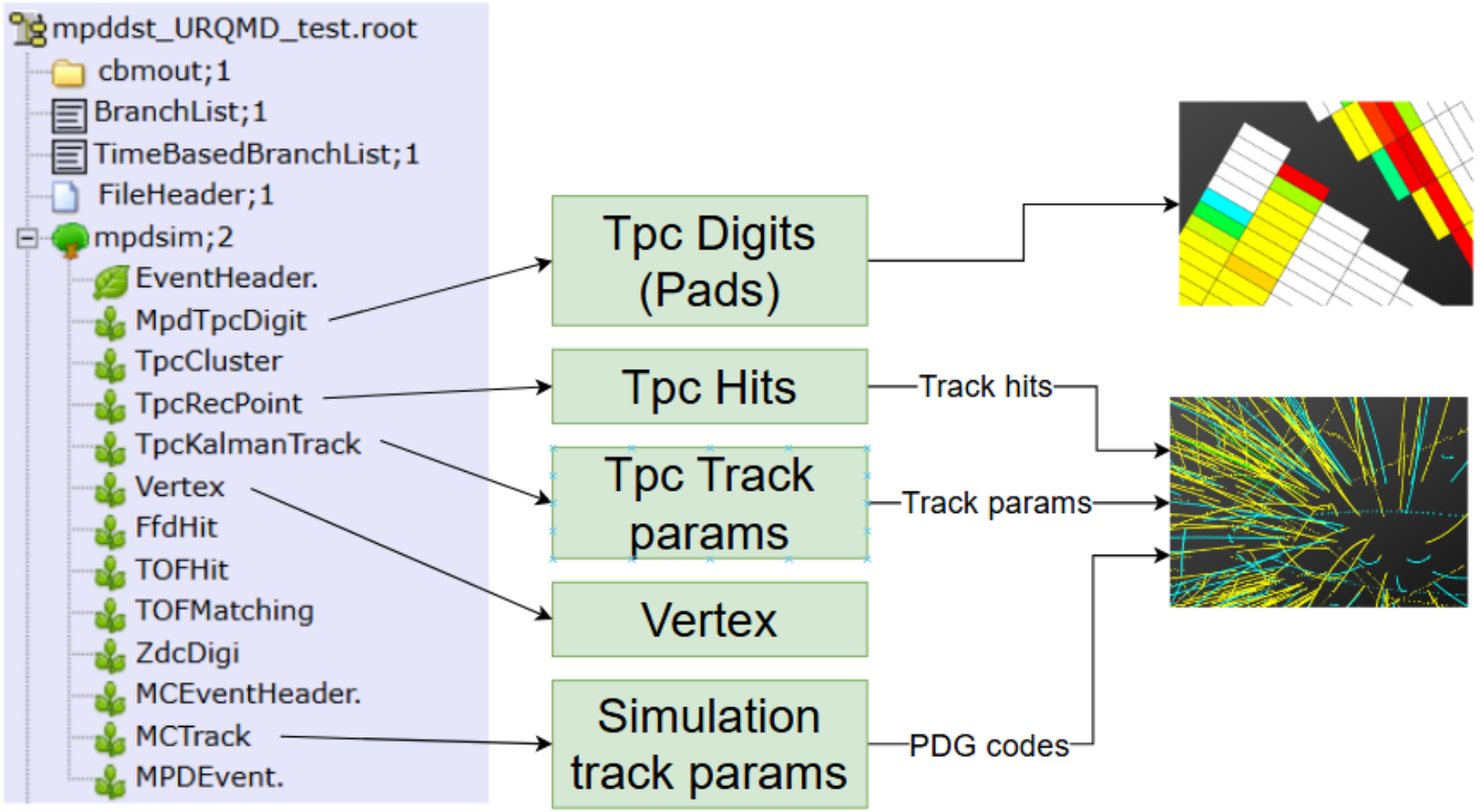
- All the geometry and event visualisation is done **on the client side**;
- Client Side presents this data through interactive components like sensor information and event visualization.
- As a result, during offline work, the client does not have to wait for the file to be uploaded to the server and can start working with it immediately



Current mpddst format



Full mpddst format



Add reconstructed hits

To add TPC reconstructed hits add line

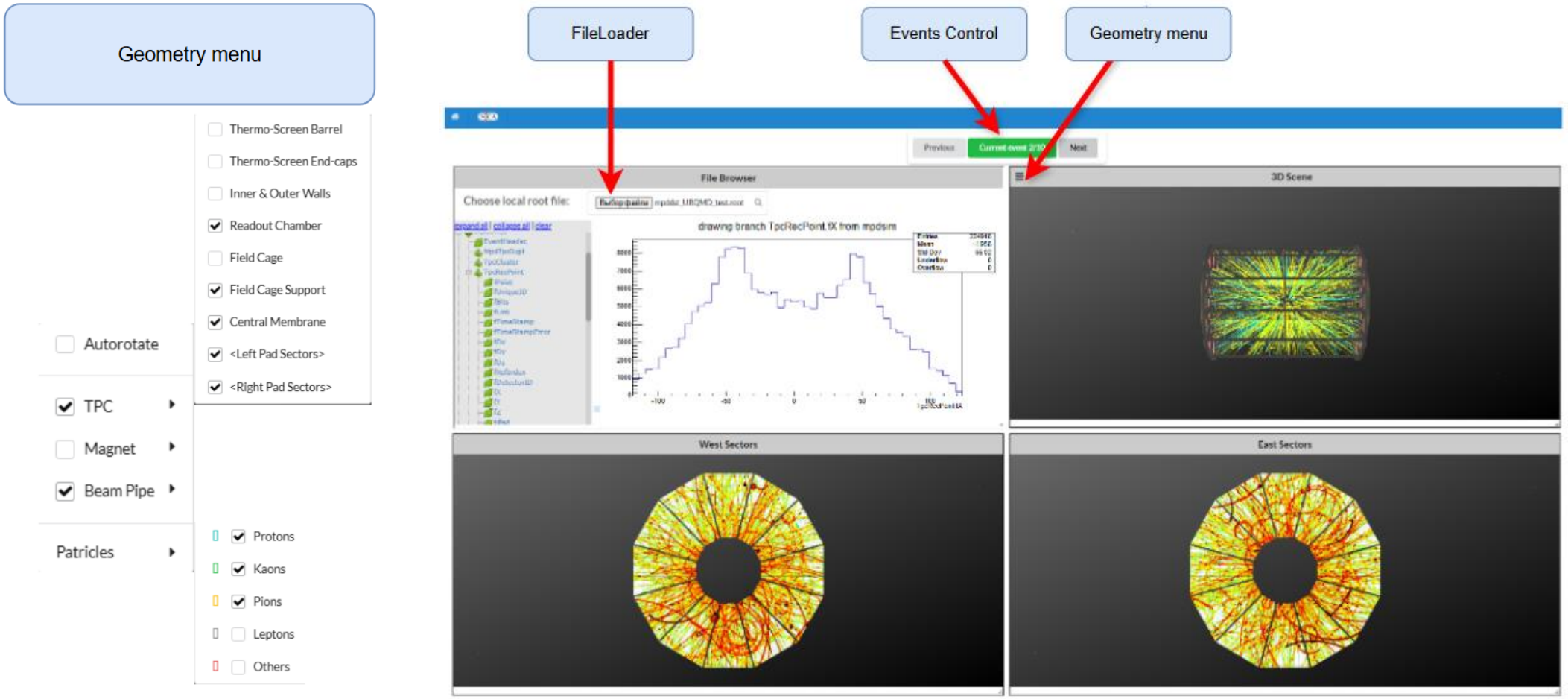
tpcClus->SetPersistance(kTrue)

In **runReco.C** macros

```
AbstractTpcClusterHitFinder *tpcClus;
switch (tpcClusteringModule) {
   case ETpcClustering::MLEM: {
      tpcClus = new TpcClusterHitFinderMlem(*secGeo, qaObject);
      break;
   }
   case ETpcClustering::FAST: {
      tpcClus = new TpcClusterHitFinderFast(*secGeo, qaObject);
      break;
   }
   default: {
      std::cerr << "Error. You've set the non-existing Tpc Clustering Module.\n";
      return;
   }
}
tpcClus->SetPersistence(kTRUE);
fRun->AddTask(tpcClus);
}
```

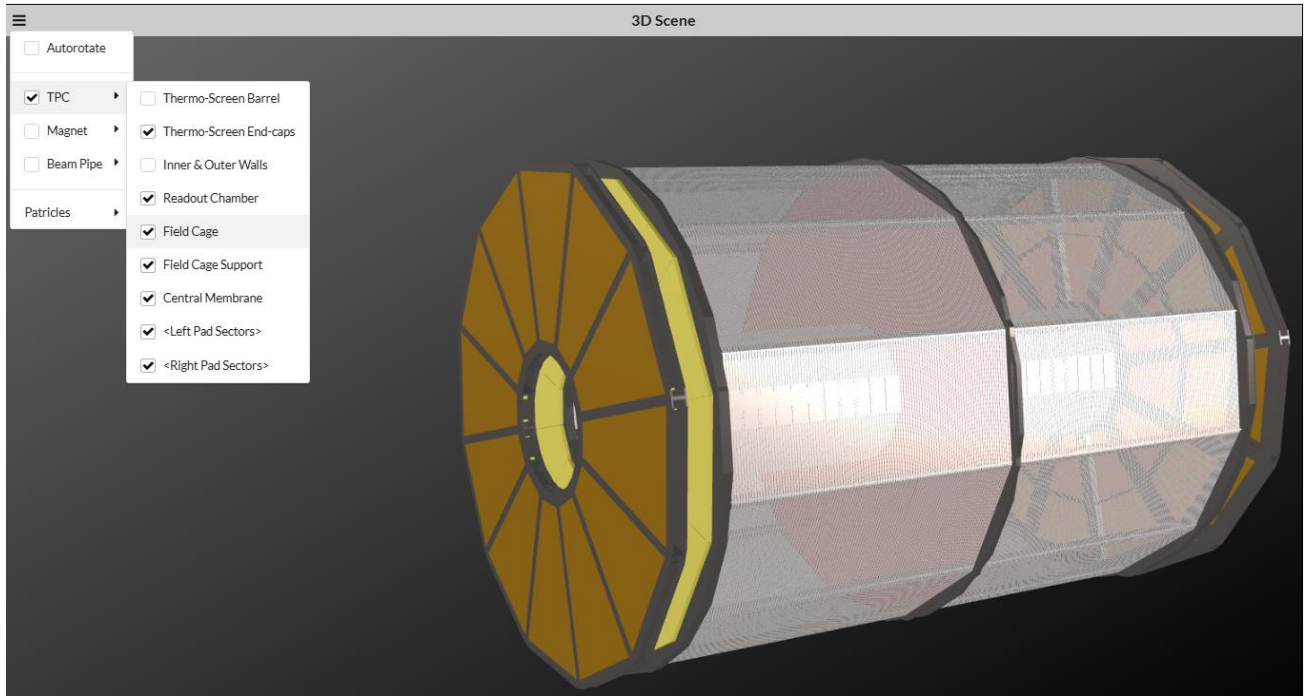
runReco.C macros for mpdroot v25.03.25

Event Viewer design



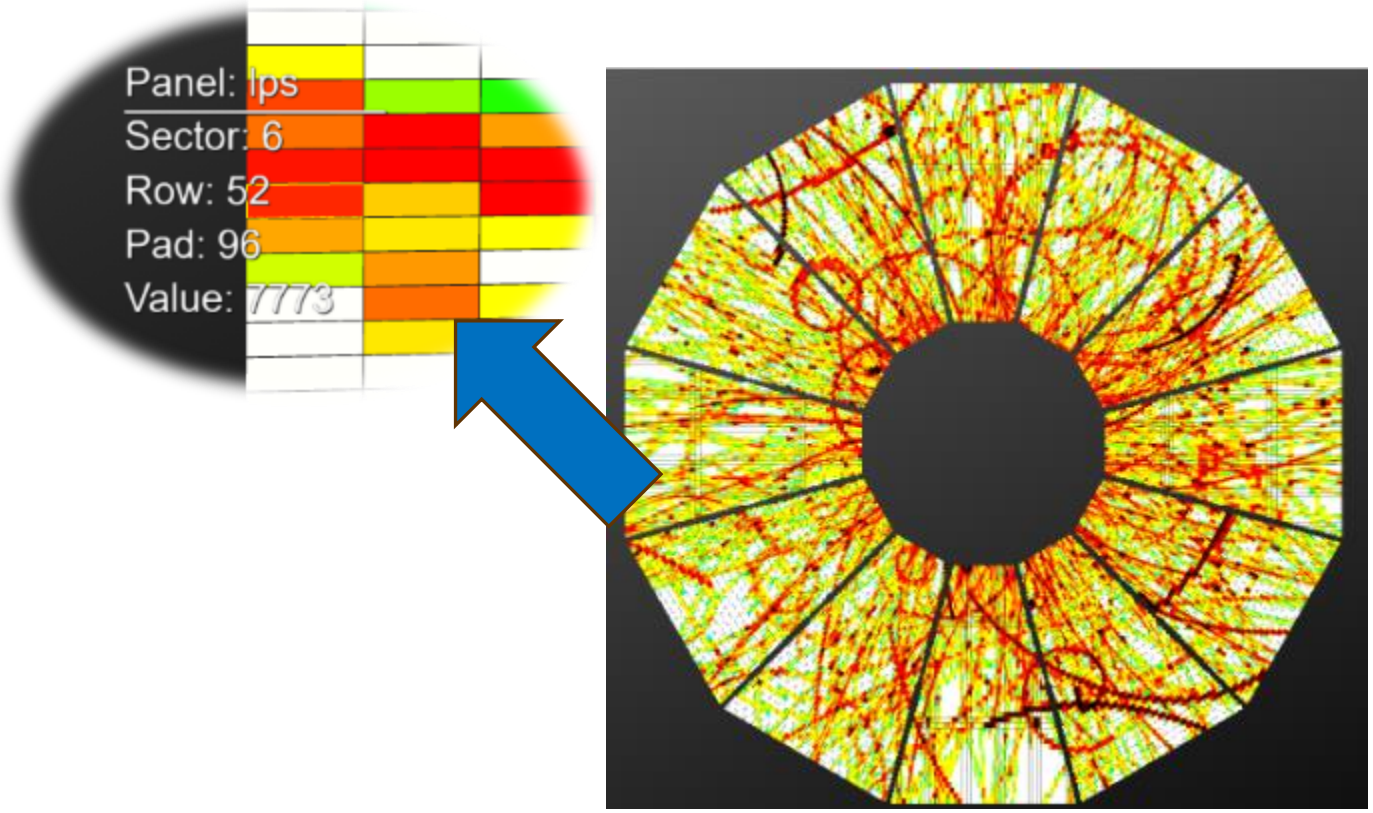
Detector geometry visualisation

- Detector geometry stores on server side and is received by the client upon the first connection.
- Geometry are simplified for drawing speed purposes.
- Different part of detector can be enabled and disabled at the user request.
- Also 3D scene has a possibility to move and rotate viewpoint in all 3 axes.



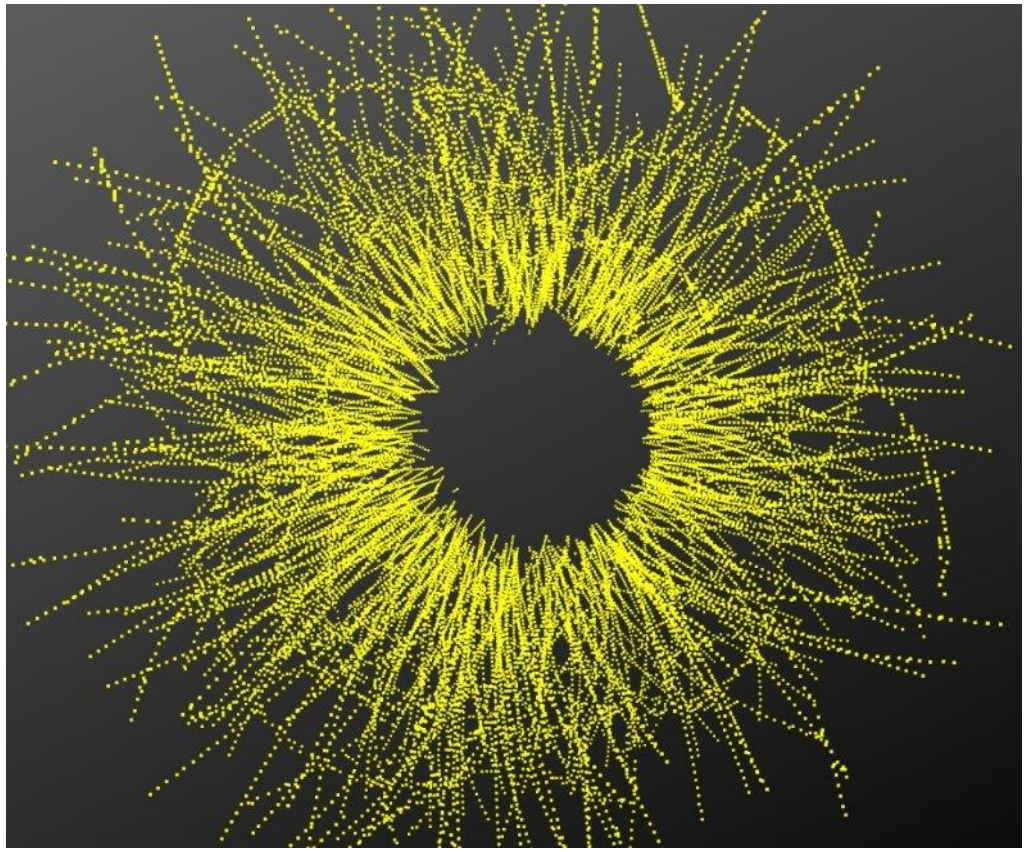
TPC pad response

- TPC pads for west and east endcup sectors.
- Each TPC pad has cumulative charge value for the whole event.
- User can zoom in (wheel) and move (right mouse pressed) through the different sectors and see the choosed pad information (left ctrl pressed)



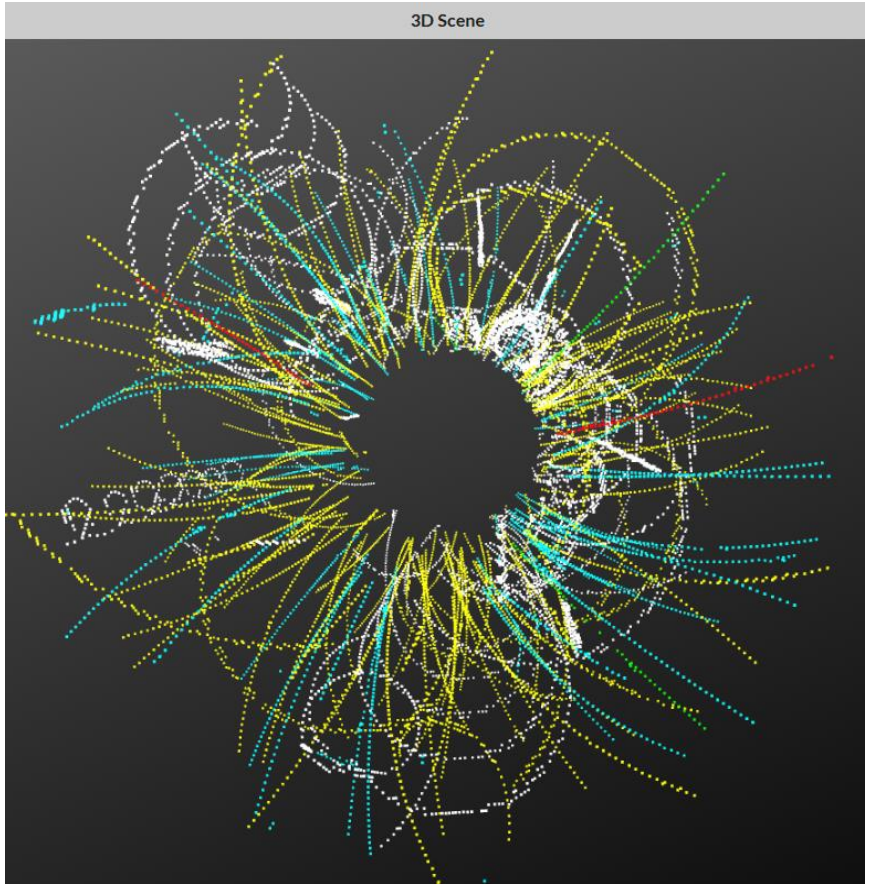
TPC Hits visualisation

- TPC Hits – reconstructed 3D points inside TPC detector
- Can be visualised ONLY if **TpcRecPoint** branch are in DST file.
- Hits are visualized for one event.



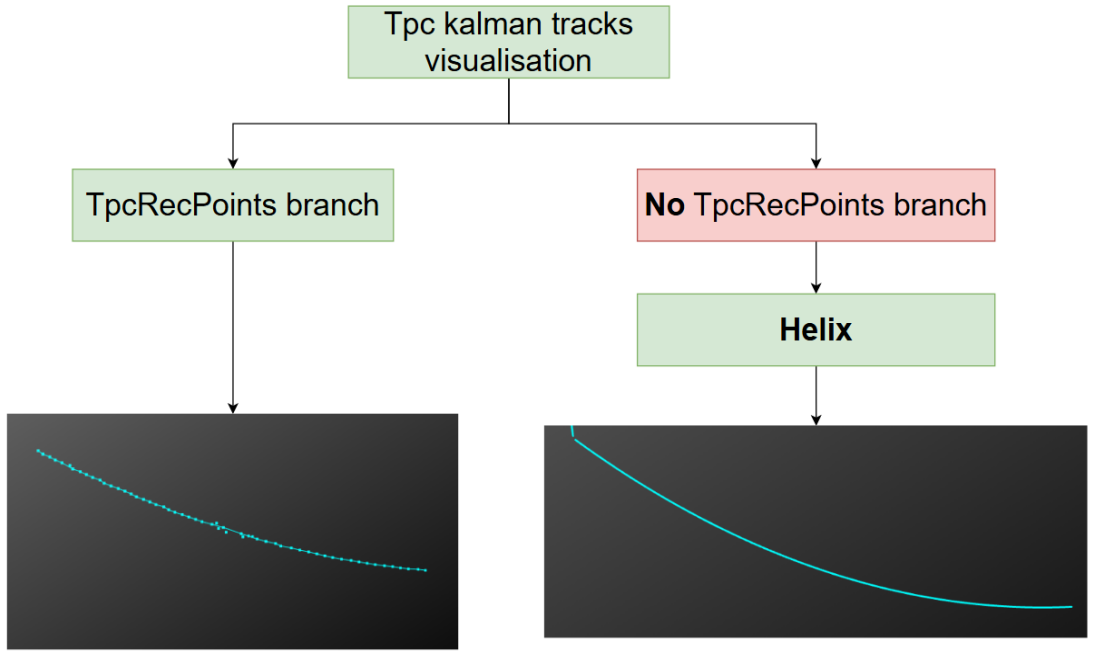
TPC Hits additional MC information

- While using simulation data reconstructed TPC hits has reference to MonteCarlo track (MCTrack branch) they are from.
- Using that information it is possible to distinguish different patricles by its pdg code and visualize it in different colors.
- Used colors:
  - Protons (2212) -> cyan
  - Pions (211) -> yellow
  - Kaons (321) -> green
  - Leptons (11, 13) -> red
  - Other -> white

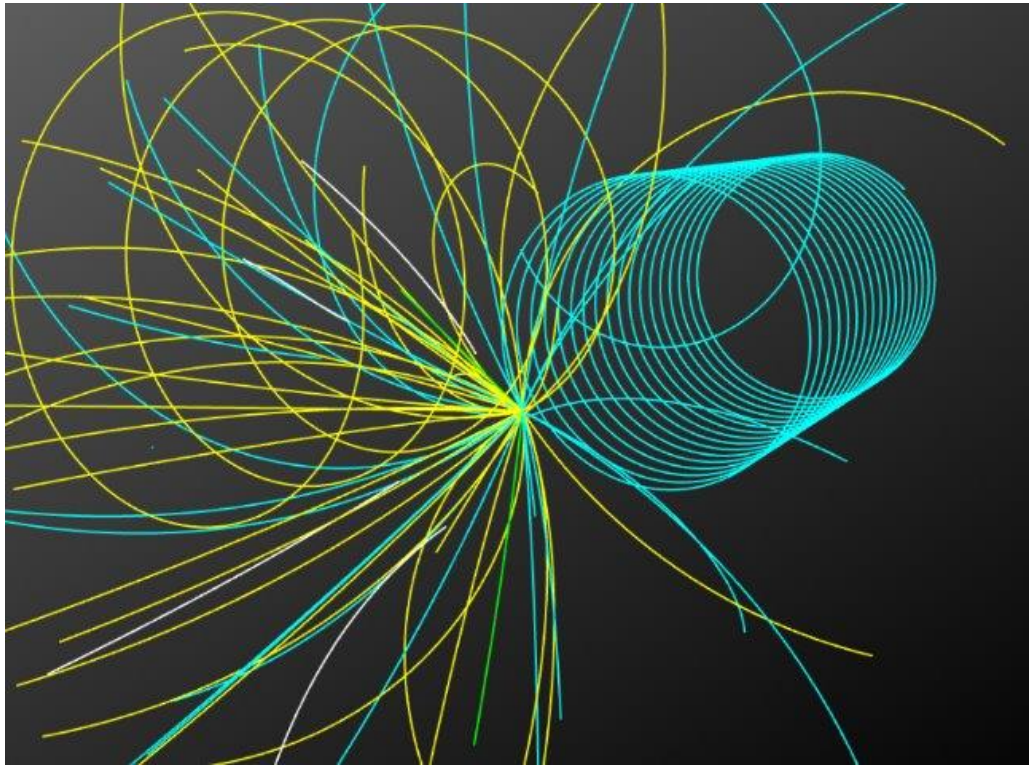


TPC Tracks visualisation

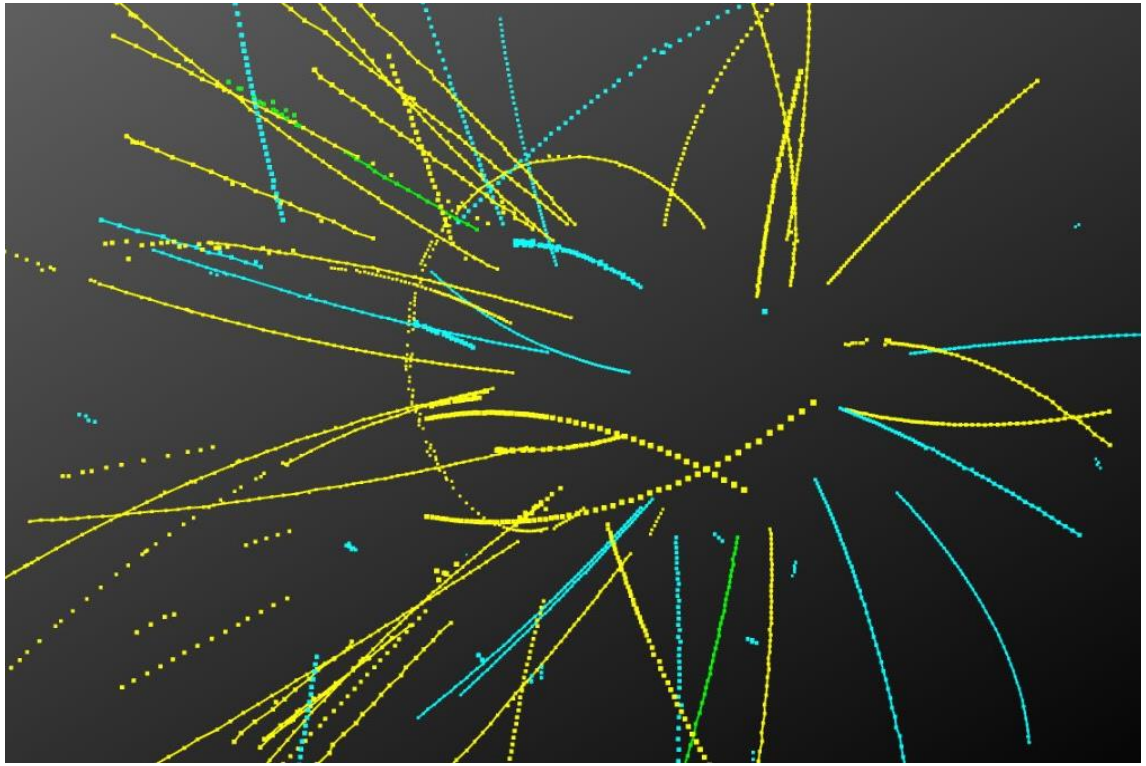
- TPC tracks – reconstructed kalman tracks inside TPC detector (**TpcKalmanTrack** branch).
- Two methods to draw tracks:
  - If TPC hits (the TpcRecPoints branch) are enabled, then each Kalman track has its own hits, which are used to construct it as a line;
  - If no TPC hits included, to visualize each Kalman track its Helix representation is used.
- During offline work there are information about pdg code of the reconstructed track. That information is used to distinguish different particles by colors (same as for TpcHits).



TPC Tracks visualisation



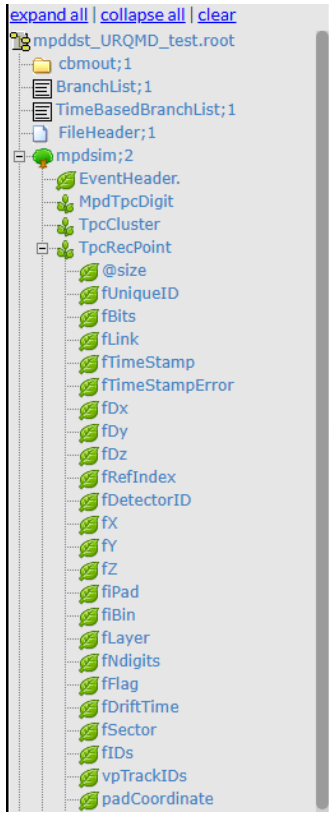
TPC tracks using Helix

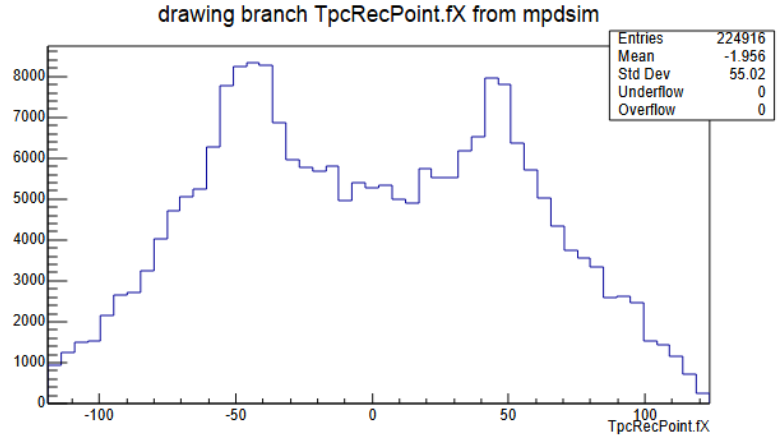


TPC tracks by reconstructed TPC Hits

Root browser usage

- **JSROOT** library has the possibility to create root type FileBrowser for local client files;
- Browser can visualise histograms and TTree branches information;
- It is done with the help of **HierarchyPainter** and **d3.js** library.

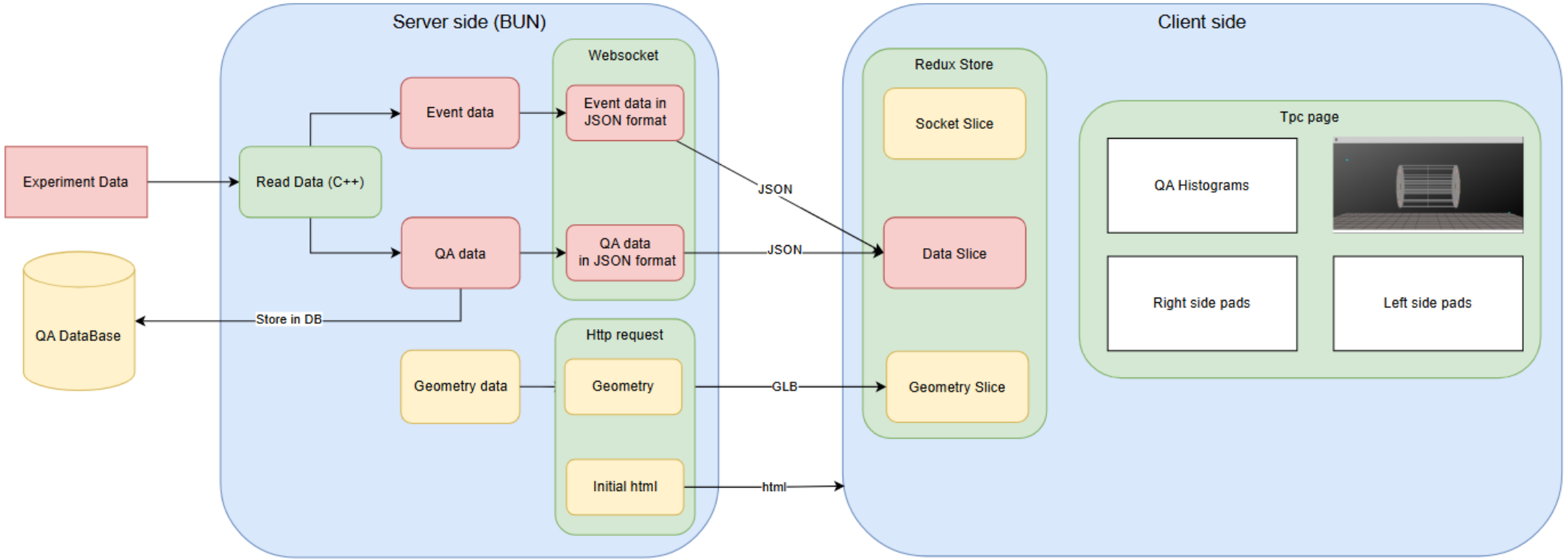




Used web technologies

	New	Old
FrontEnd	React + TypeScript	React + JavaScript
BackEnd	Bun	NodeJs
Server	Bun.serve	Express
Bundler	Bun	WebPack
DataBase	Bun:sqlite	None
Connection	Bun.ws	WebSocket
Store	Redux toolkit	Redux
Graphics	ThreeJS + JSROOT	ThreeJS

Server side data handling (next step)



Summary

- The client side of the event display has been updated to the newest technologies: BUN, Typescript and Modern Redux.
- At the moment it was implemented fully for the TPC detector. Other detectors will be added later.
- Now the event display works with local DST files, using the JSROOT.
- It is available from the mpdroot site (mpdroot.jinr.ru -> MPD visualization -> MPD Event Viewer).

MPD Software Development & Computing Team

Rogachevsky O. Coordinator

Krylov V., Krylov A. Online MPD Event Display

Moshkin A., Pelevanyuk I. Mass Production

Bychkov A. Detector Simulation

Kuzmin V. Detector Alignment

Podgainy D., Zuev M. Supercomputing

Alexandrov E., Alexandrov I. Databases

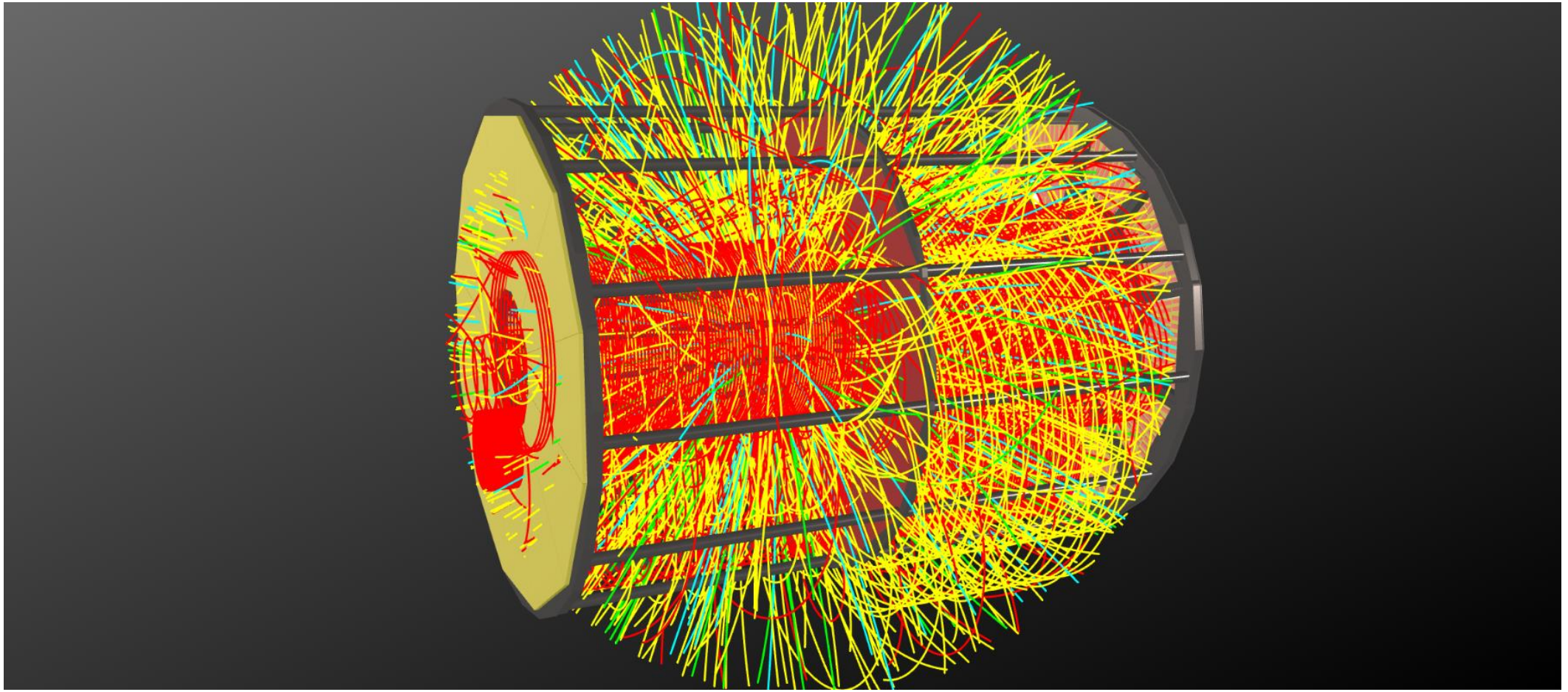
Balashov N. Gitlab Support

Belecky P., Kamkin A. Acts Tracker

Busa J. Build System

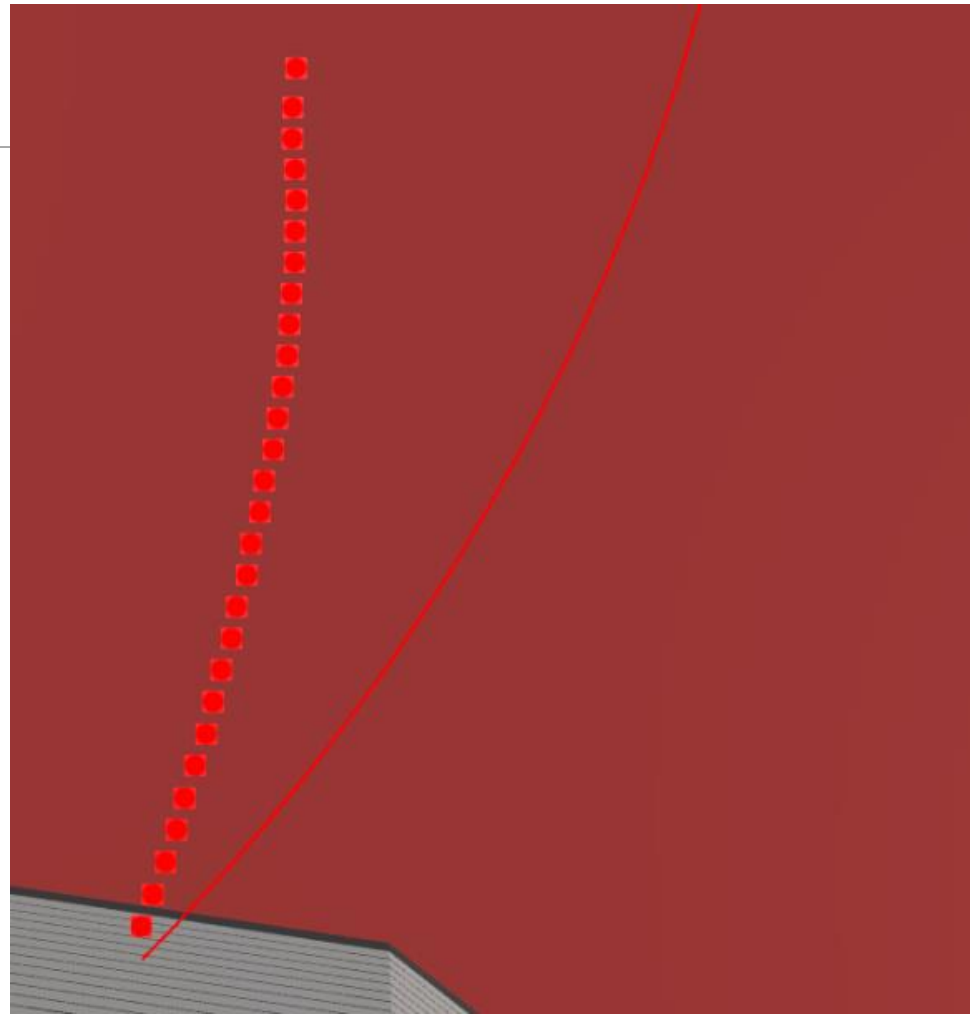
Hnatic S. Architecture

Thank you for your attention!

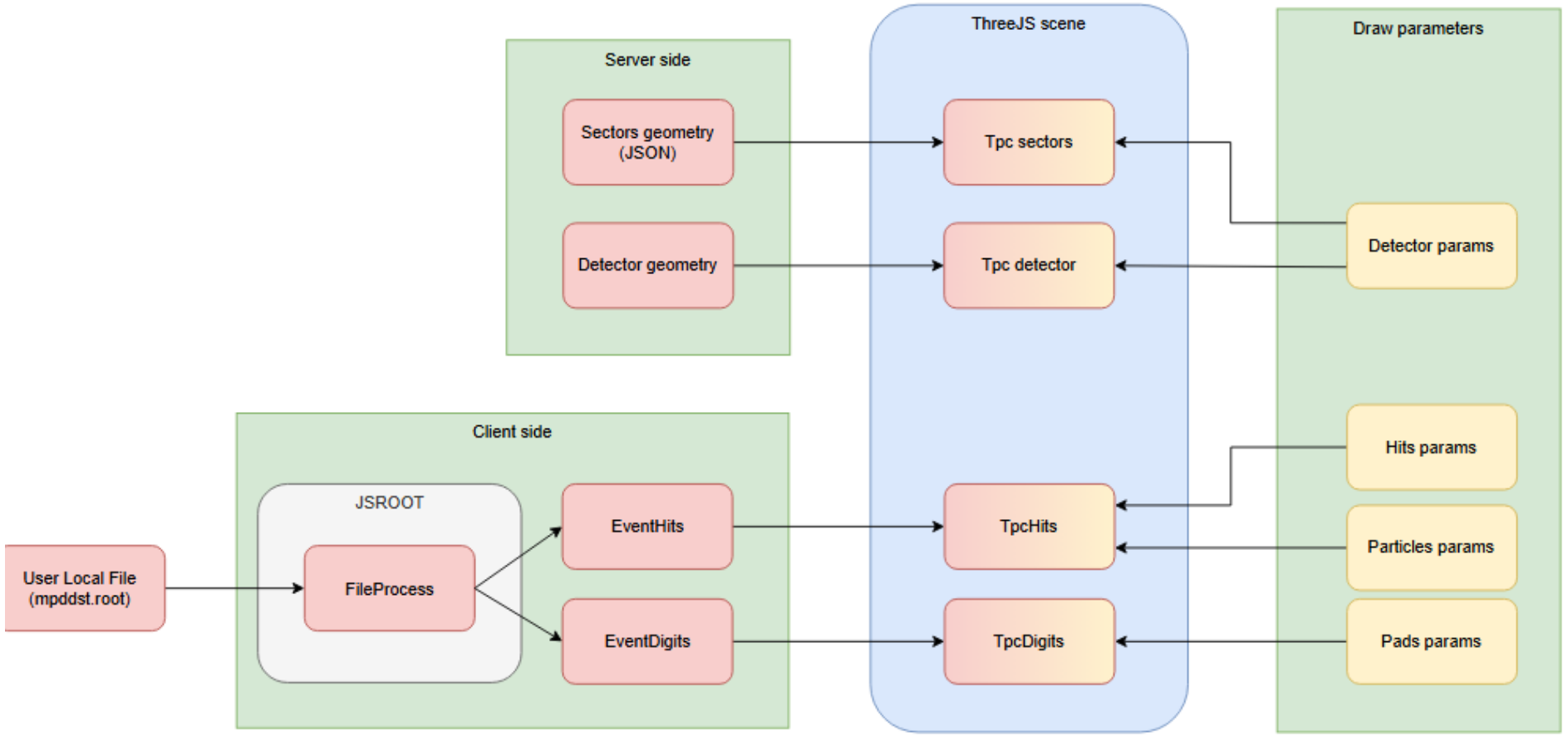


Helix inaccuracy

- Helix representation of track does not take into account perticles ionization energy loss in TPC gas;
- It means that track built by Helix function will not match with the original track exactly.
- Sometimes that discrepancies can be quite big.



Client data flow scheme



Event Broker