EIC Event Display

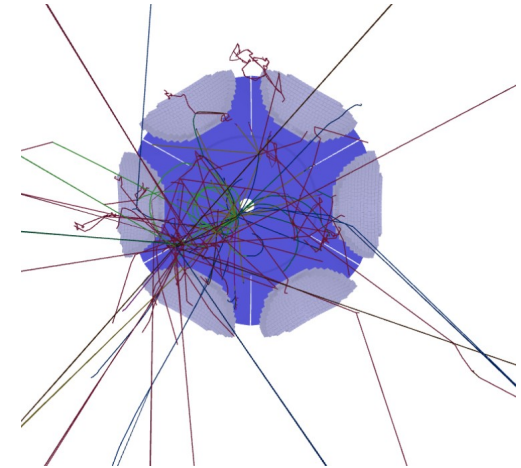## Phoenix based event display for EPIC

Dmitry Romanov

Event display requirements

Requirements for event display

Two primary purposes:

1. Outreach through enhanced visualization of detector and physics data

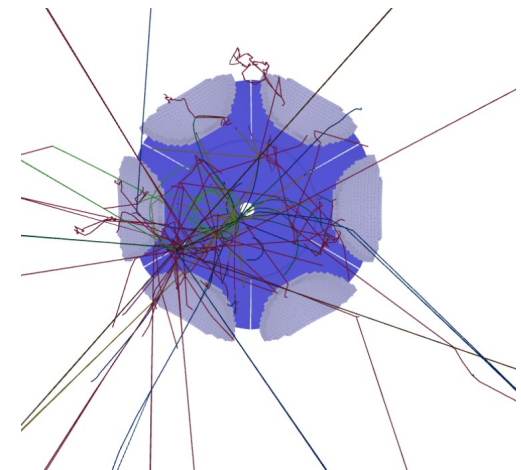2. **(!) Detailed debugging of simulation and reconstruction**
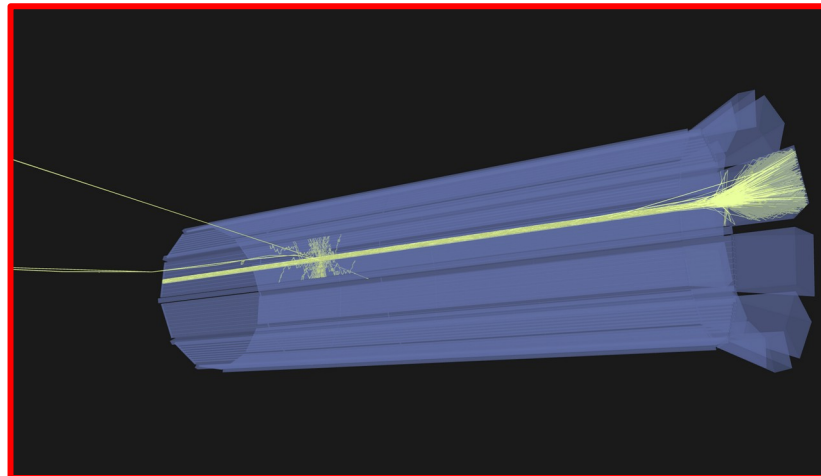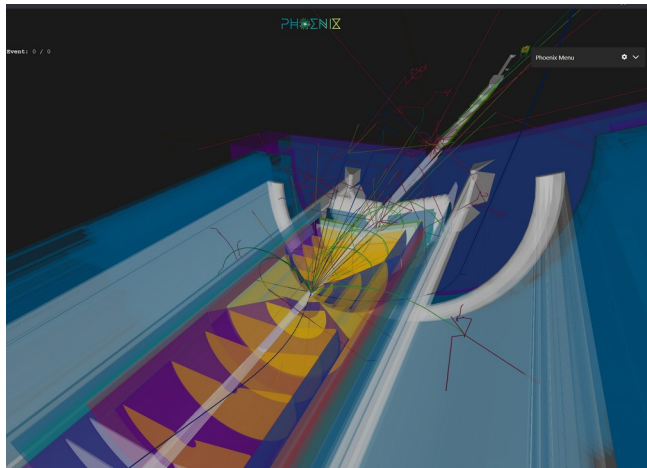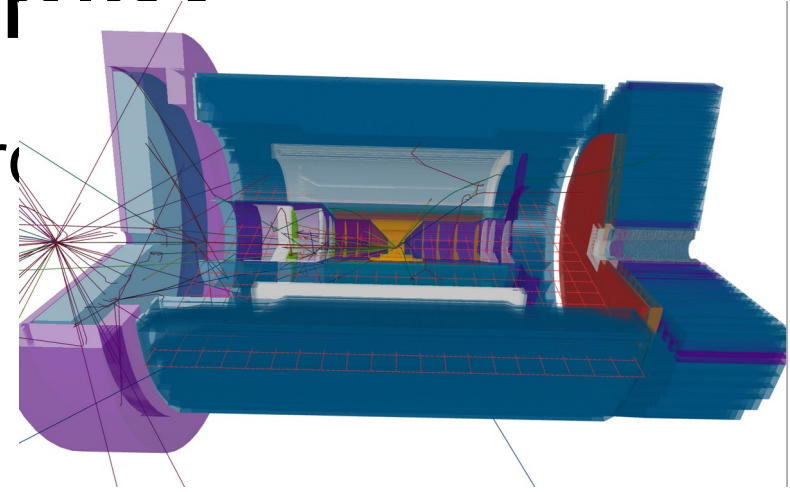
What is there so far

- DD4Hep plugin saving tracks to phoenix for
- Hits in EDM4Hep (json, + root in work)
- Load current geometries
- Can handle far-forward
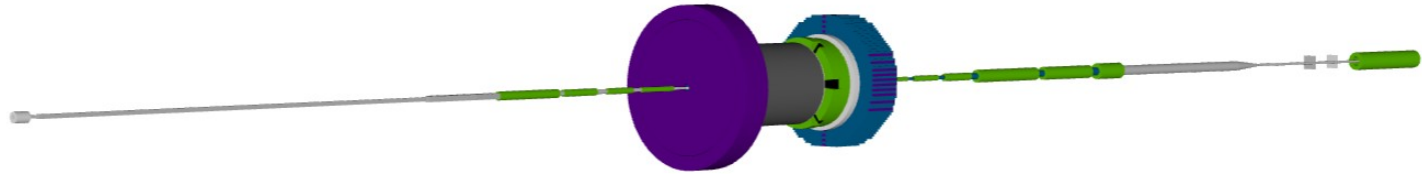- Geometry optimization (being in progress)
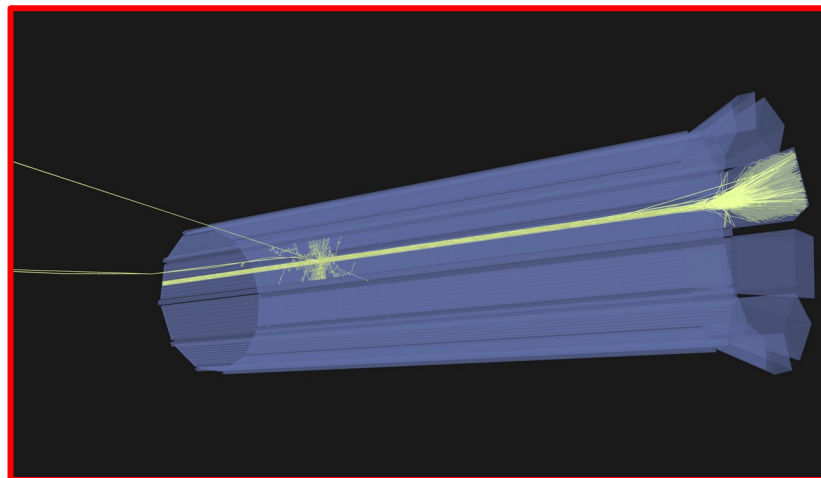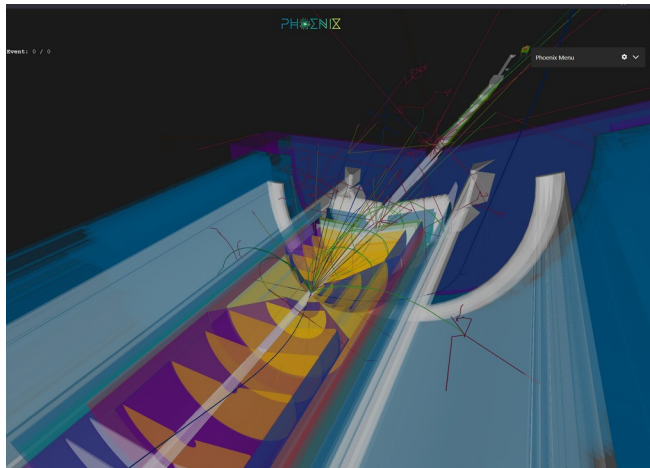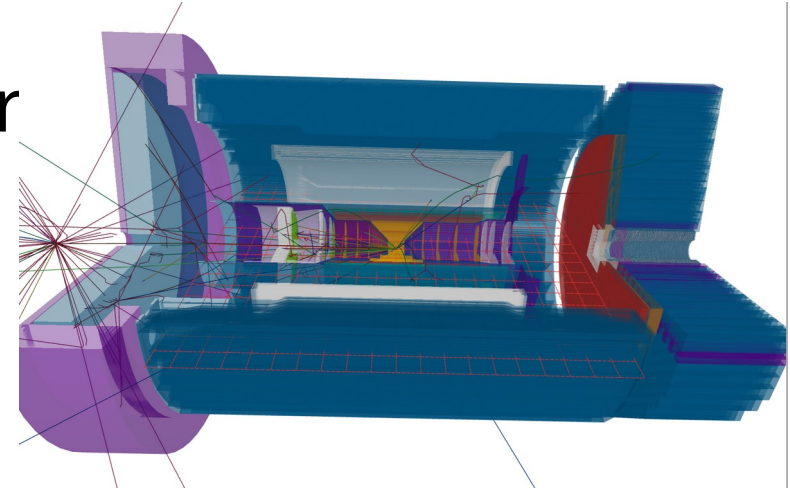
EIC Phoenix based event display

- Together with HSF Phoenix event display gr
- https://github.com/eic/firebird
- Static website using GH pages
- https://eic.github.io/firebird

Geometry time!

Geometry

Geometry Node

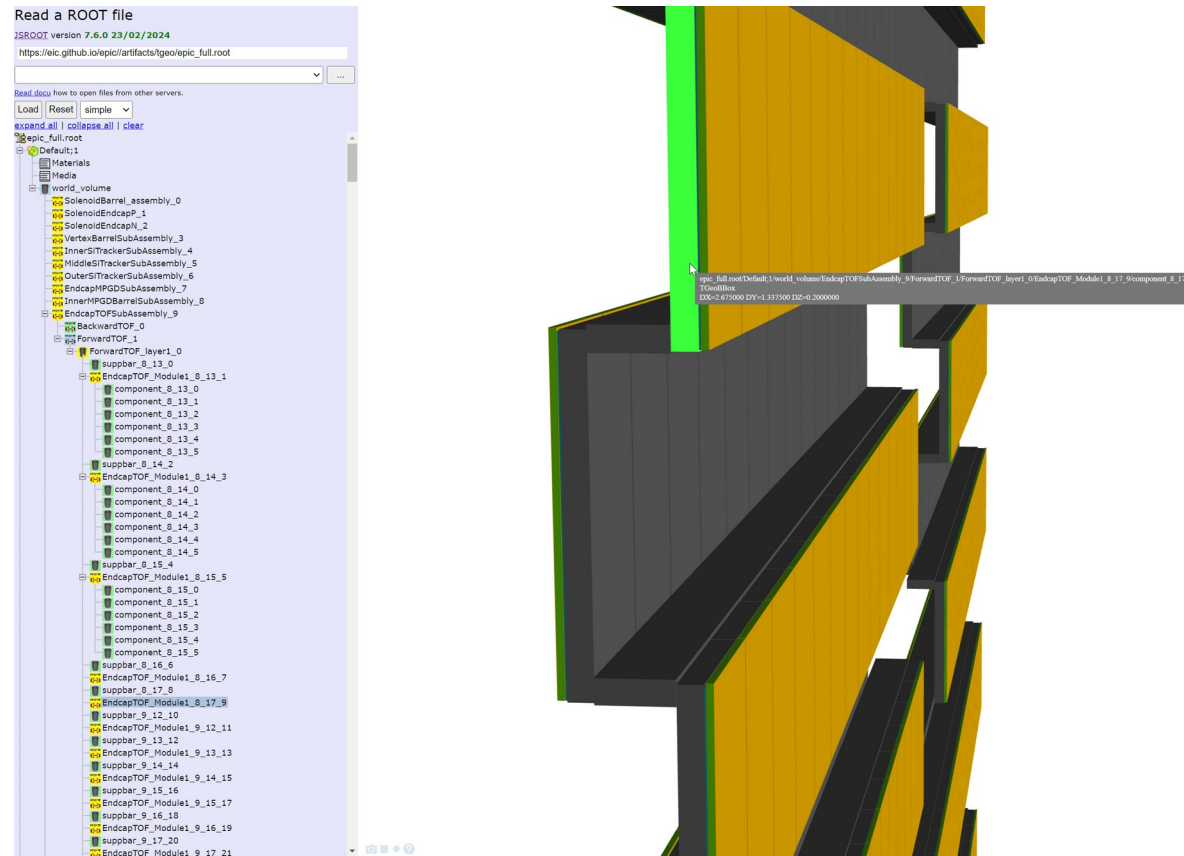| Node Count	Geometry Name
2,007,793	EcalBarrelScFi
1,816,561	EcalBarrelImaging
1,185,962	LFHCAL
324,163	LumiSpecCAL
32,872	EcalEndcapN
25,211	DRICH
7,716	HcalBarrel
3,860	EndcapTOFSubAssembly
2,311	EndcapMPGDSubAssembly
1,795	InnerMPGDBarrelSubAssembly
1,537	DIRC
1,443	BarrelTOFSubAssembly
1,151	OuterSiTrackerSubAssembly
910	RICHEndcapN
802	ZDC_Crystal
773	VertexBarrelSubAssembly
588	HcalEndcapPInsert
579	HcalFarForwardZDC_SiPMonTile
471	MiddleSiTrackerSubAssembly
455	ForwardRomanPot_Station_1

Distribution of Geometry Nodes
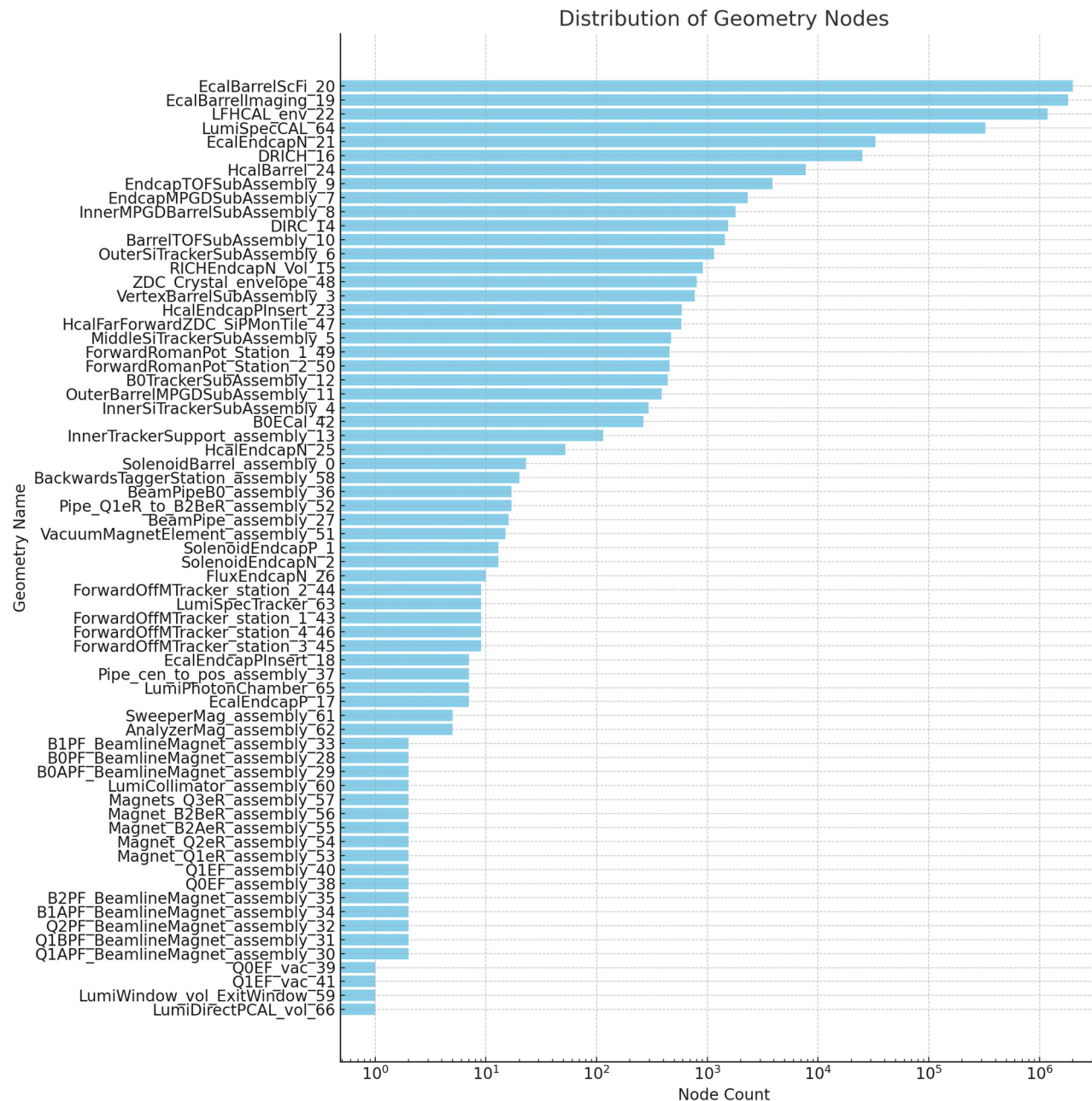
ECAL Barrel

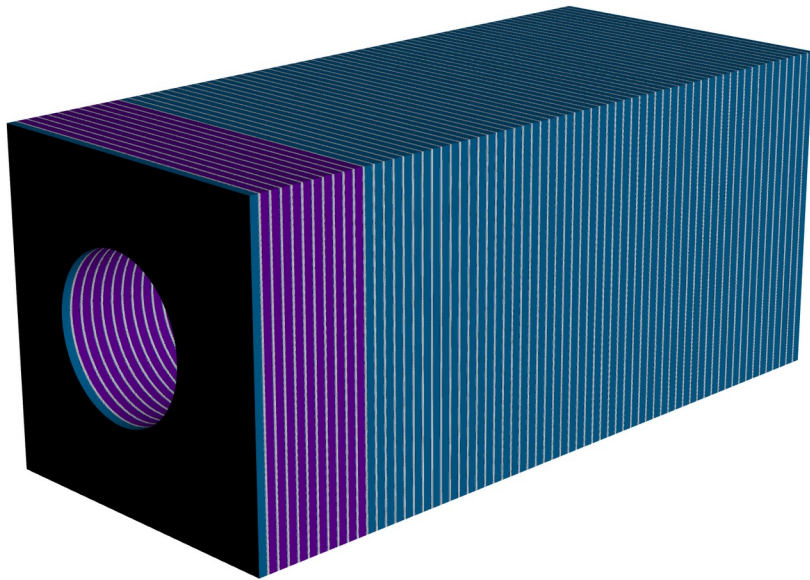# Hcal Barrel

- Small number of parts, but active use of tessellated objects
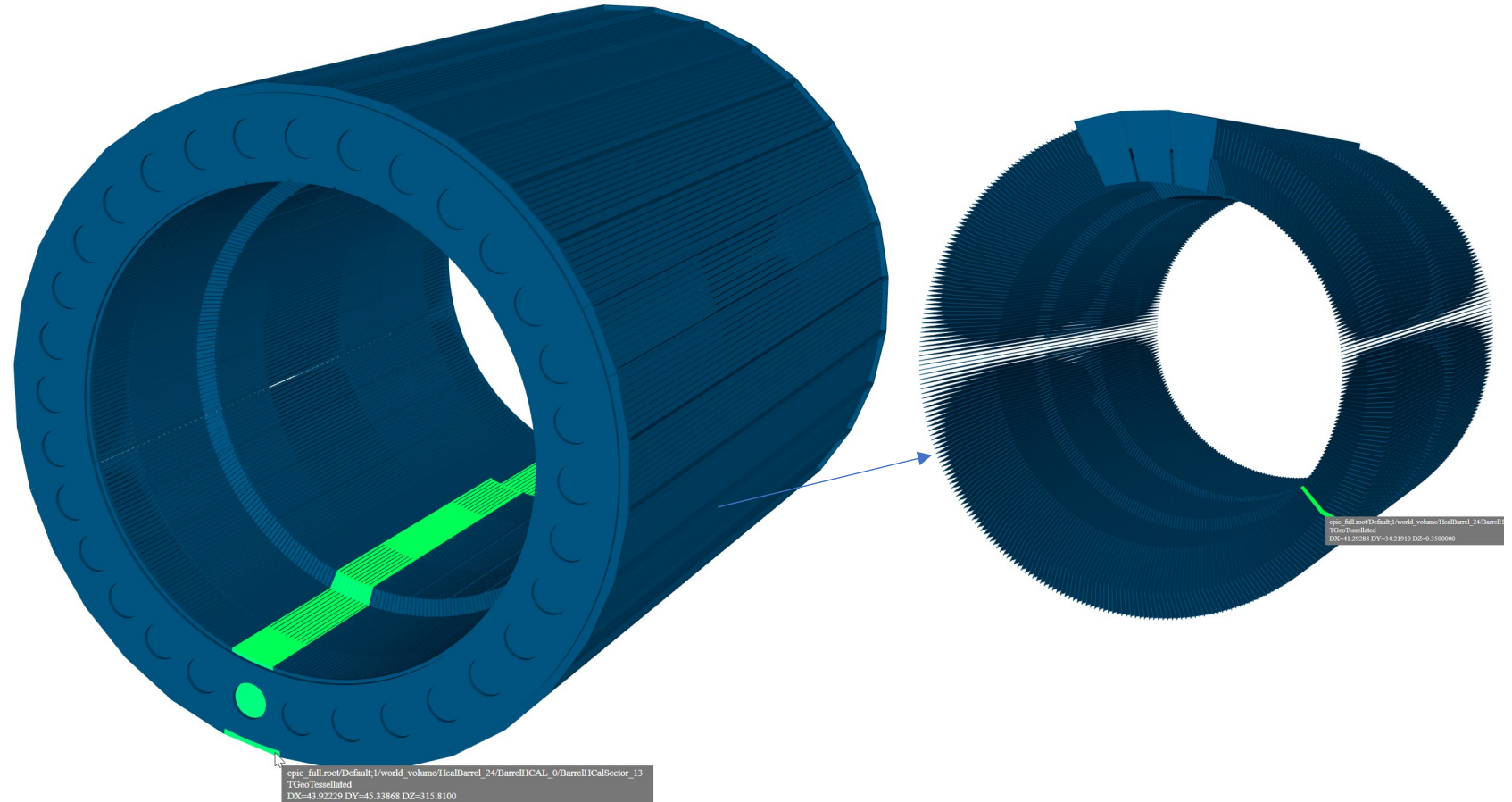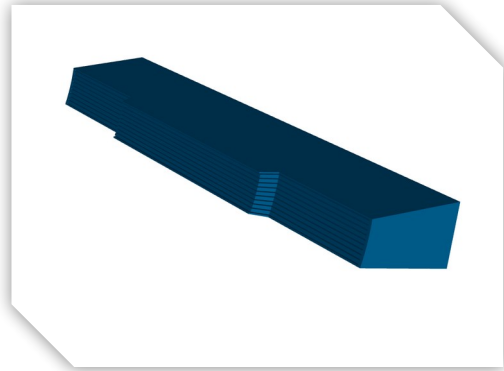
Inconsistent naming

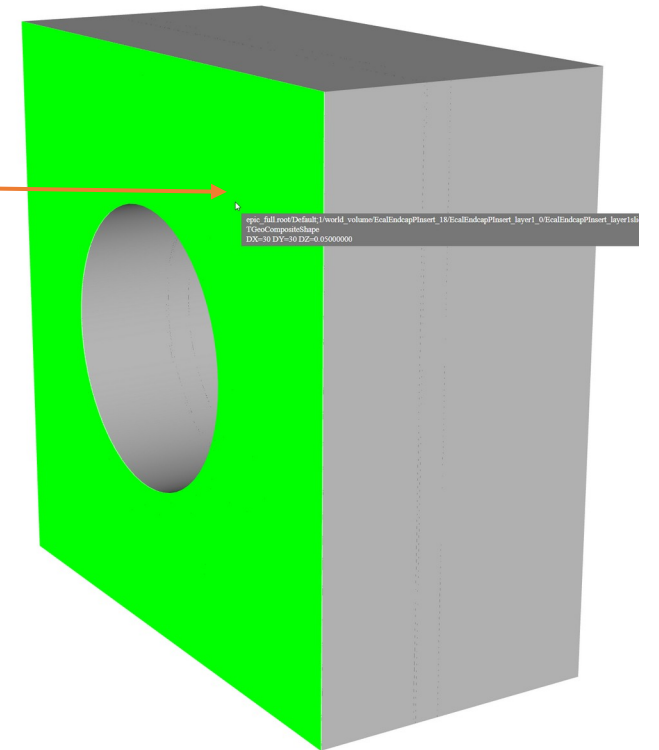- Many geometry elements named assuming no one ever will use names

Same name? EcalEndcapPInsert

Performance Inner Detector

epic_inner_detector.root

(All numbers are [ms])

- Open root file: 678
- Reading geometry from file: 993
- Go over all nodes: 132
- Build geometry (for WebGL): 2002
- Convert to JSon geometry (Phoenix): 20

Total visible nodes 15902 numfaces 412466

So now it works fine?

WebGL performance

29964 DRAW CALLS

WebGL performance

29964 DRAW CALLS

What is good drawcals number?

Can we do the same?

- Kind of but...
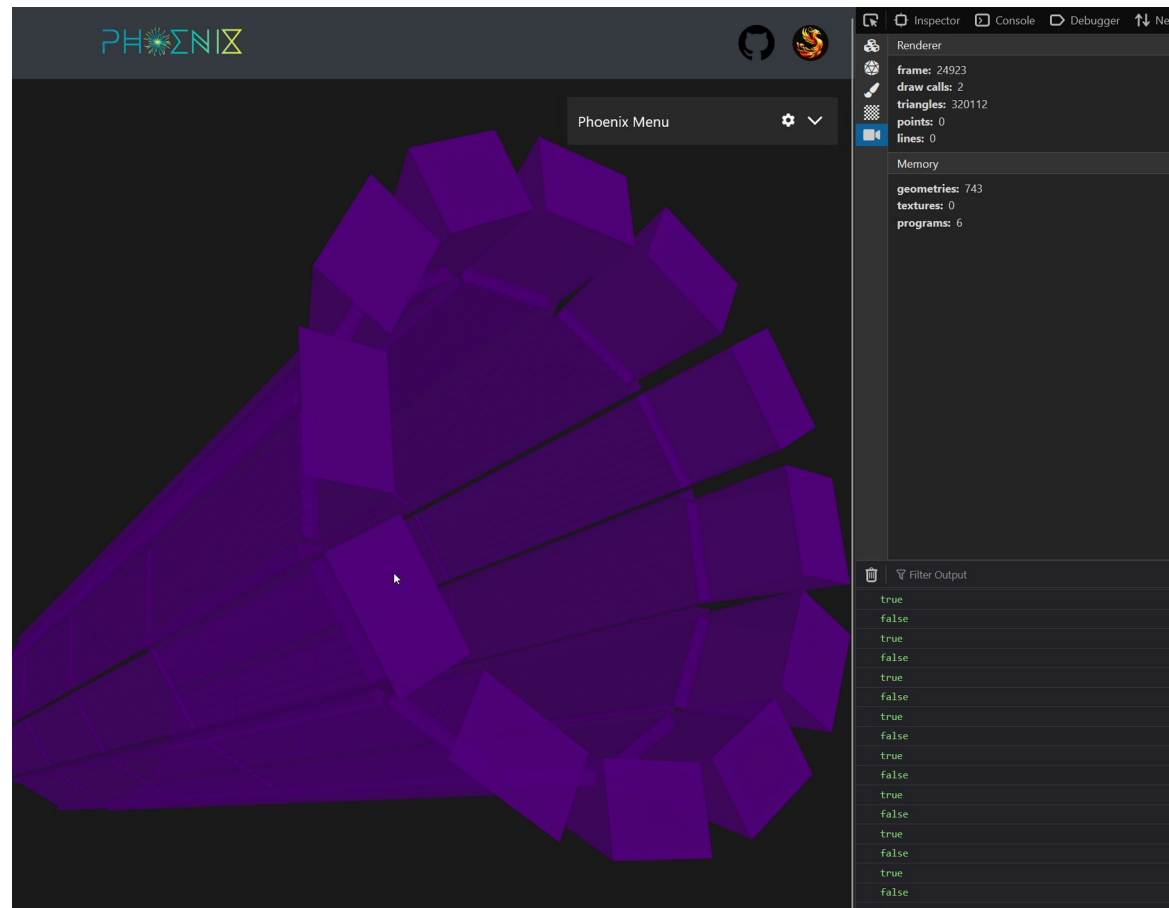
Data processing chains

- Geometry pipeline

| Server-side Processed | Read ROOT file | Geometry PRE-process | **BUILD** root->three | Geometry POST-process | Display webGL |

Current results

- ~100 Drawcals and 1000 geometries
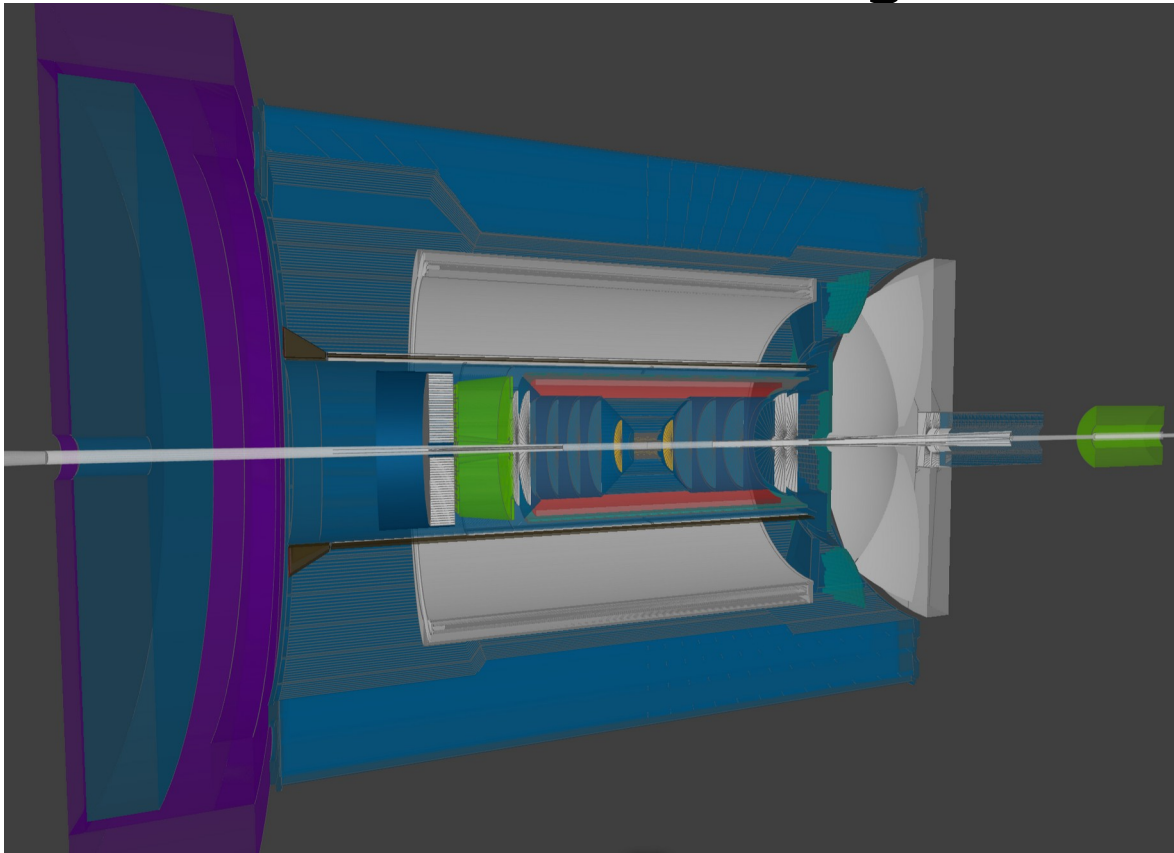
Renderer

frame: 5352
draw calls: 116
triangles: 799026
points: 0
lines: 307594

Memory

geometries: 752
textures: 0
programs: 10

What about hits?

Hits rendering is not optimized and in TODO
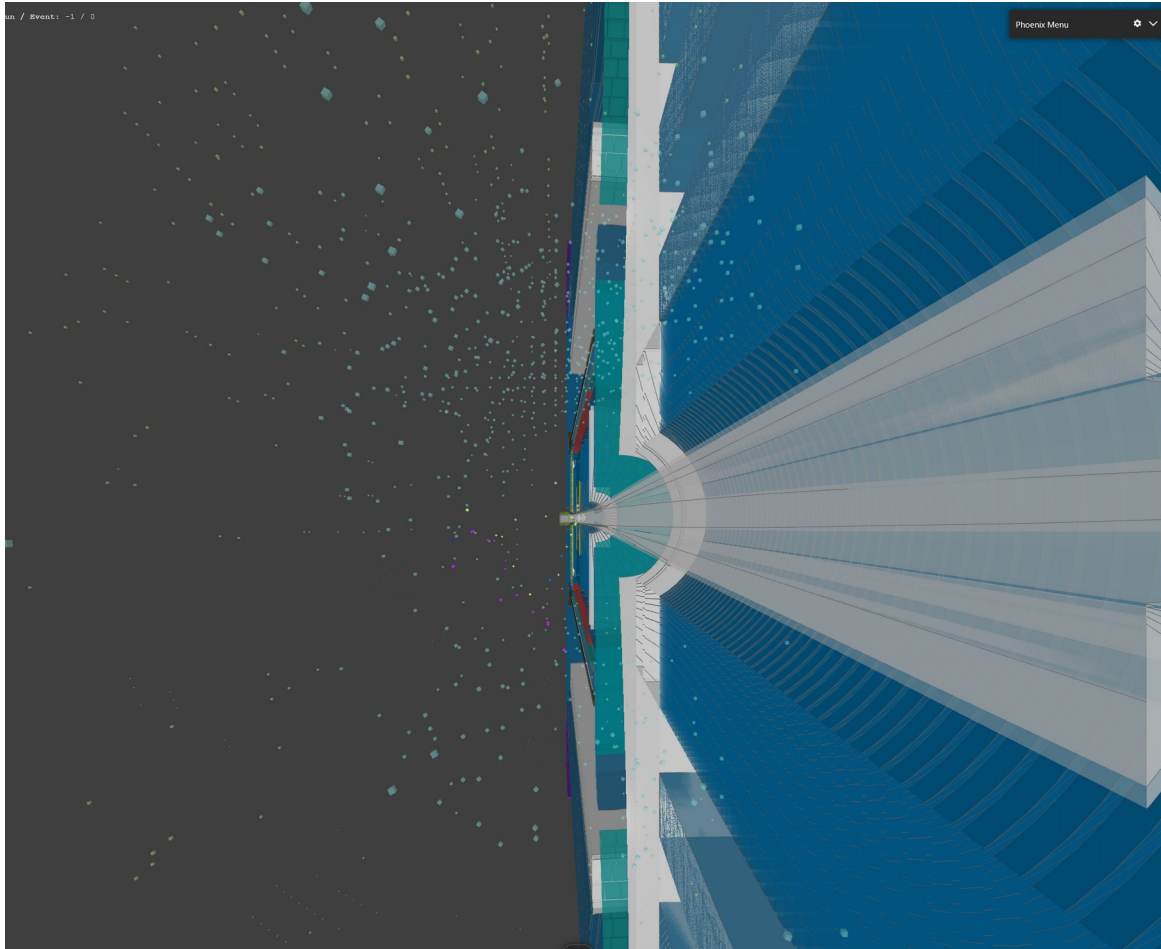
Renderer	
frame:	2376
draw calls:	17327
triangles:	981722
points:	69
lines:	281672
Memory	
geometries:	23879
textures:	14
programs:	10

Future plans

User local CLI (visioning)

- Vision:

  pip install pyrobird

```
> fbd display -g geometry -I input1 …
```

```
(base) romanov@SmallBrave:/mnt/c/eic/firebird$ pip install pyrobird
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pyrobird in /home/romanov/.local/lib/python3.10/site-packages (0.0.5)
Requirement already satisfied: click in /home/romanov/.local/lib/python3.10/site-packages (from pyrobird) (8.1.7)

[notice] A new release of pip is available: 23.0 -> 24.0
[notice] To update, run: /usr/bin/python3 -m pip install --upgrade pip
(base) romanov@SmallBrave:/mnt/c/eic/firebird$ fbd display --geo='epic_full.root' -i sim.edm4hep.root -i reco.edm4eic.root
```

Better visual
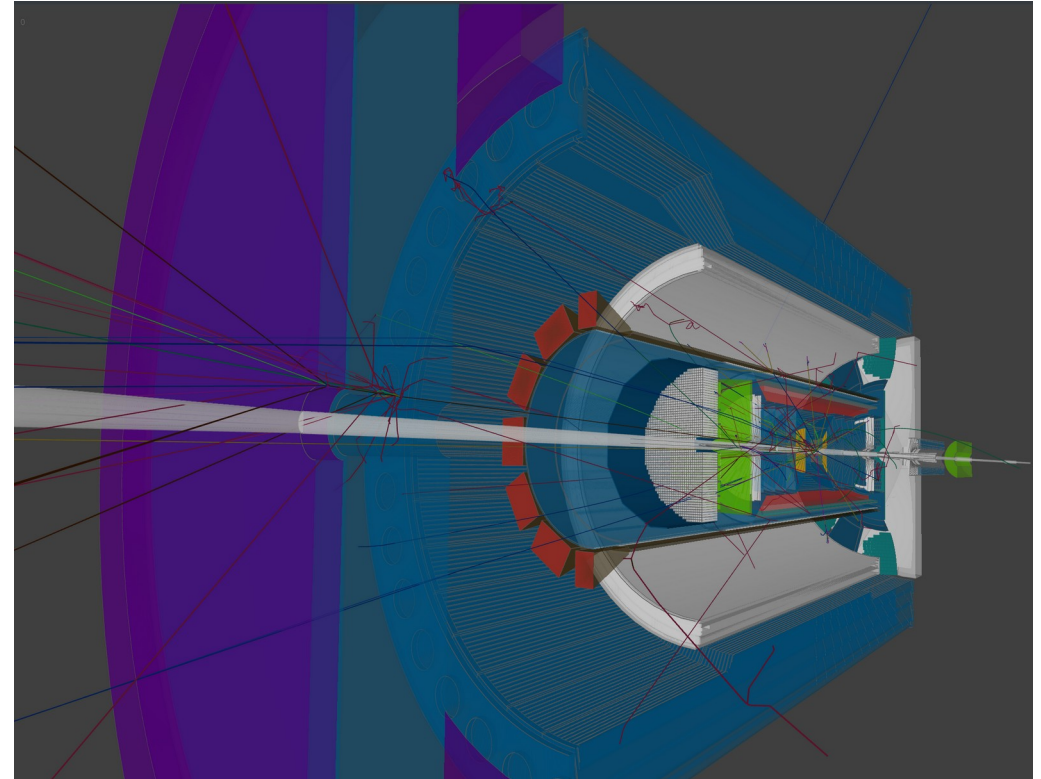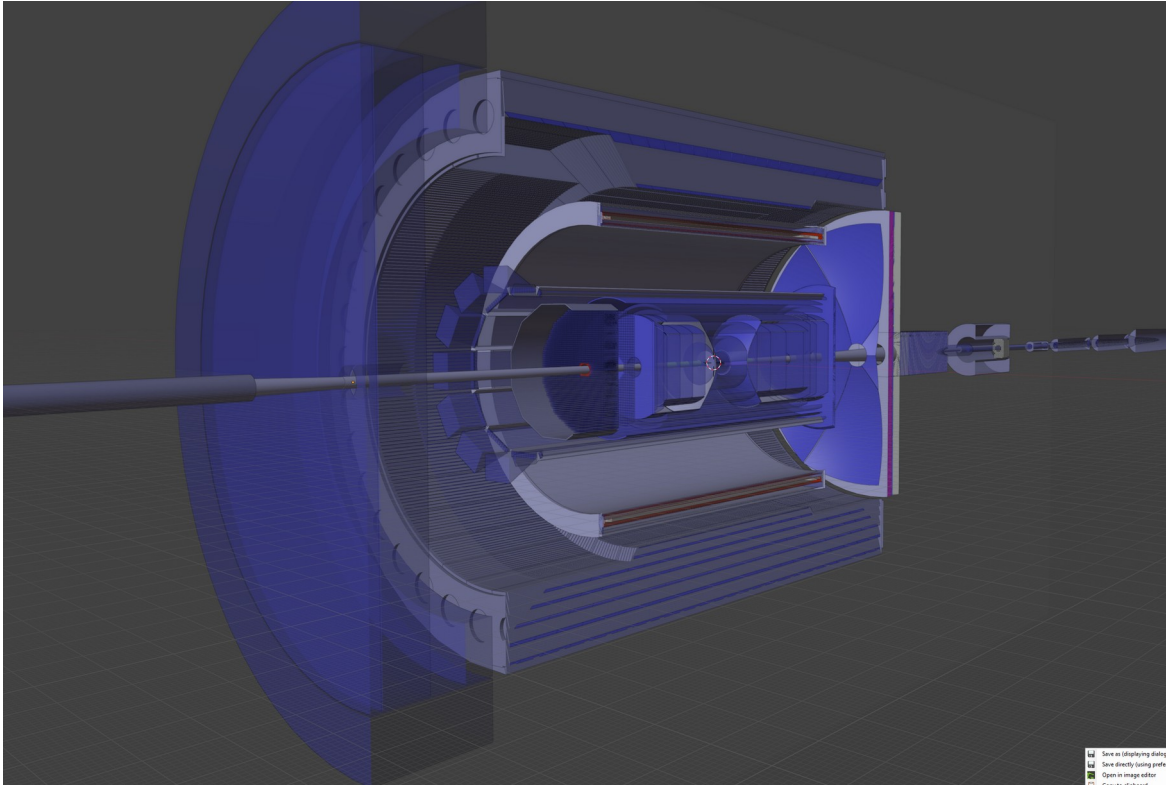
Geometry rules, community contributions

- All geometry manipulations done by set of rules

- Anticipated detector groups to contribute

- No javascript coding for rules
(as it seems now)

```
 99          {
100            namePattern: "*/EcalEndcapN*",
101            editRules: [
102              {pattern: "*/crystal*", prune:PruneRuleActions.RemoveSiblings},
103            ]
104          },
105          {
106            namePattern: "*/HcalEndcap*",
107            editRules: [
108              {pattern: "*/*layer?slice1_?", prune:PruneRuleActions.RemoveSiblings},
109            ]
110          },
111          {
112            namePattern: "*/HcalBarrel*",
113            editRules: [
114              {pattern: "*/Tile*", prune:PruneRuleActions.Remove},
115              {pattern: "*/ChimneyTile*", prune:PruneRuleActions.Remove},
116            ]
117          },
118          {
119            namePattern: "*/EndcapTOF*",
120            editRules: [
121              {pattern: "*/suppbar*", prune:PruneRuleActions.Remove},
122              {pattern: "*/component*3", prune:PruneRuleActions.RemoveSiblings},
123            ]
124          }
125
```
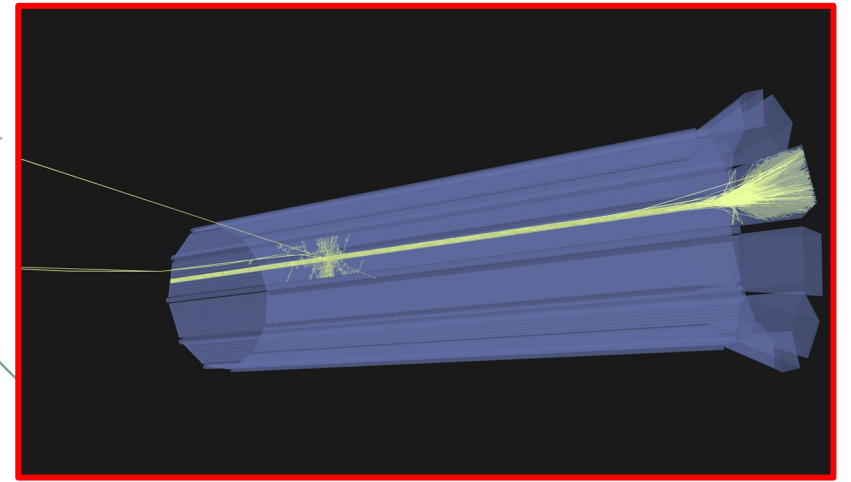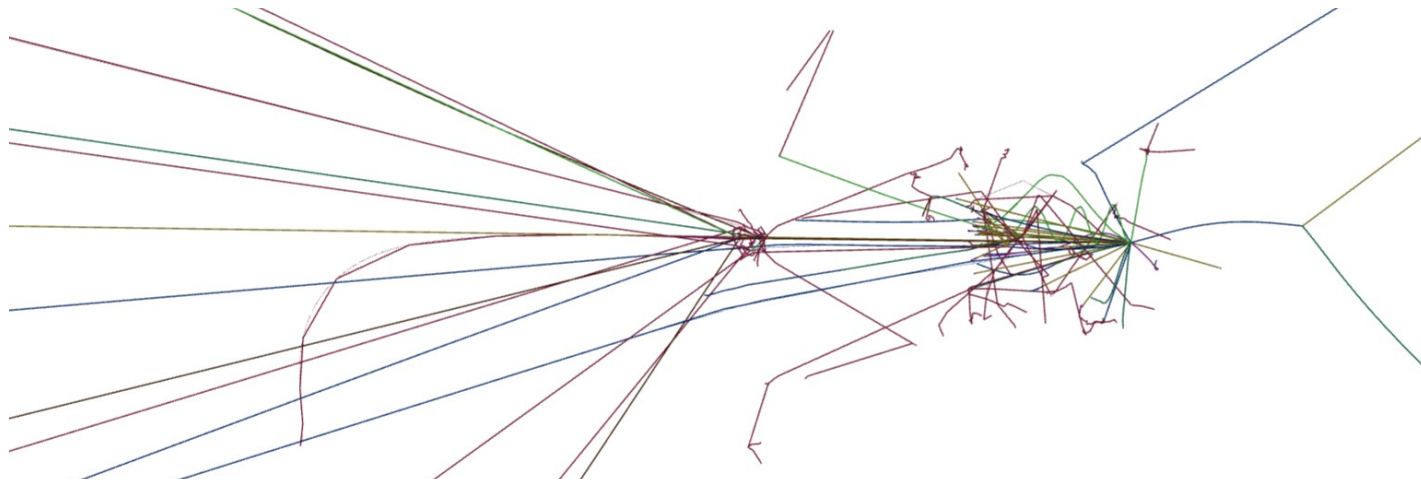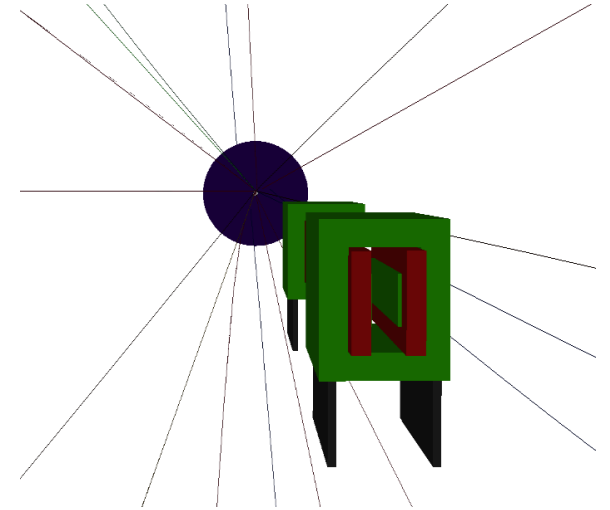
DD4HEP integration

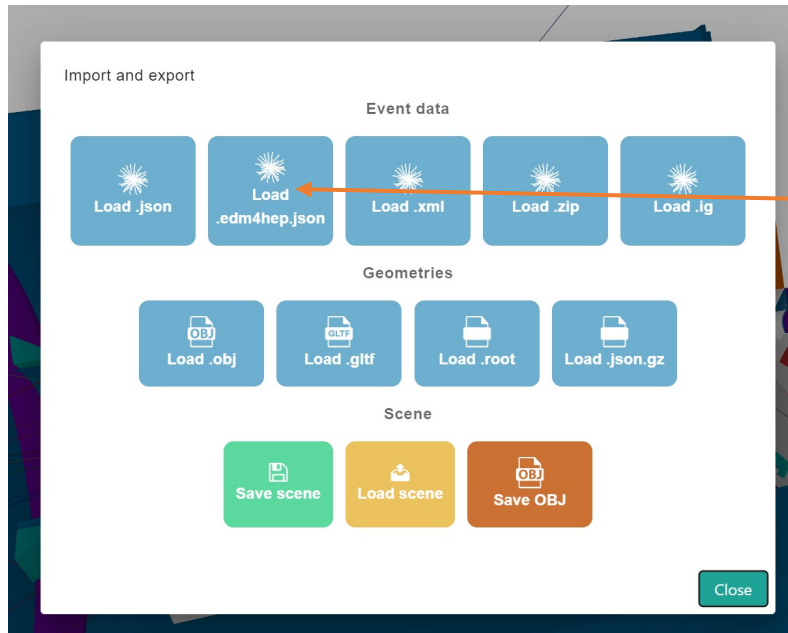- Using non-intrusive DD4Hep plugin
- Currently is a proof of concept
- Vision / possibilities:
  - Full "what is happening" (easily GBs for event)
  - Cuts by region/energy/vertex/detector, …
  - Integration with npsim
  - Write to edm4hep
  - Far forward debug

Import export

- Load edm4hep from root (not json as of now)
- Load edm4eic (planned by June 2024)
- Load ACTS reconstructed tracks and info

Deliverables June

Deliverables will depend on community. Core plans:

- **Software perspective:**
  - Initial integration with campaigns
  - Improve on modularity, making more convenient
    for groups to work on their parts of interest
  - Improve on cli part, make convenient for users and in batch operation
  - Implement particular event objects if available (recoil electron, jets, connected flows)
  - **Acts data source**
  - **EDM4EIC data source**

- **Users' perspective:**
  - Make tutorials on how to contribute to event display software
  - Collect and analyze feedback from the community to focus on further priorities
  - Can select runs and events from the latest simulation campaign reconstructed files
  (more on planning on the next page)

More on planning and deliverables

- [Implementation plan for event display on google docs](#) (written internally 2 months ago but is actual)

- [Requirements for event display](#)