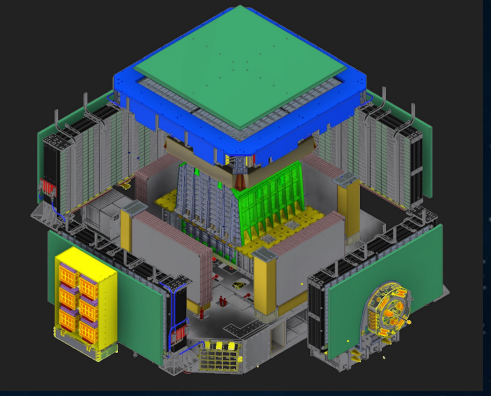
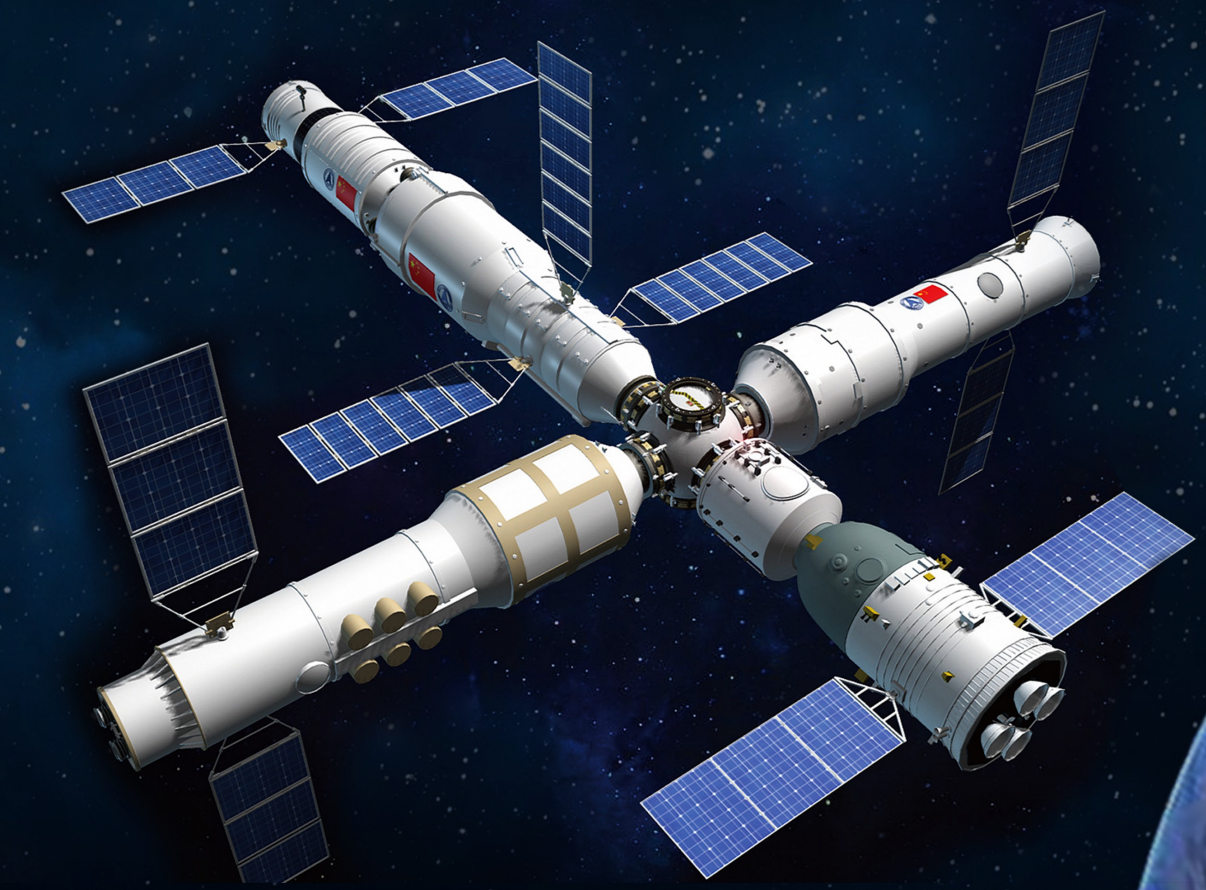
Data Reversion based on HERDOS

Zhaoyi Qu / SDIAT

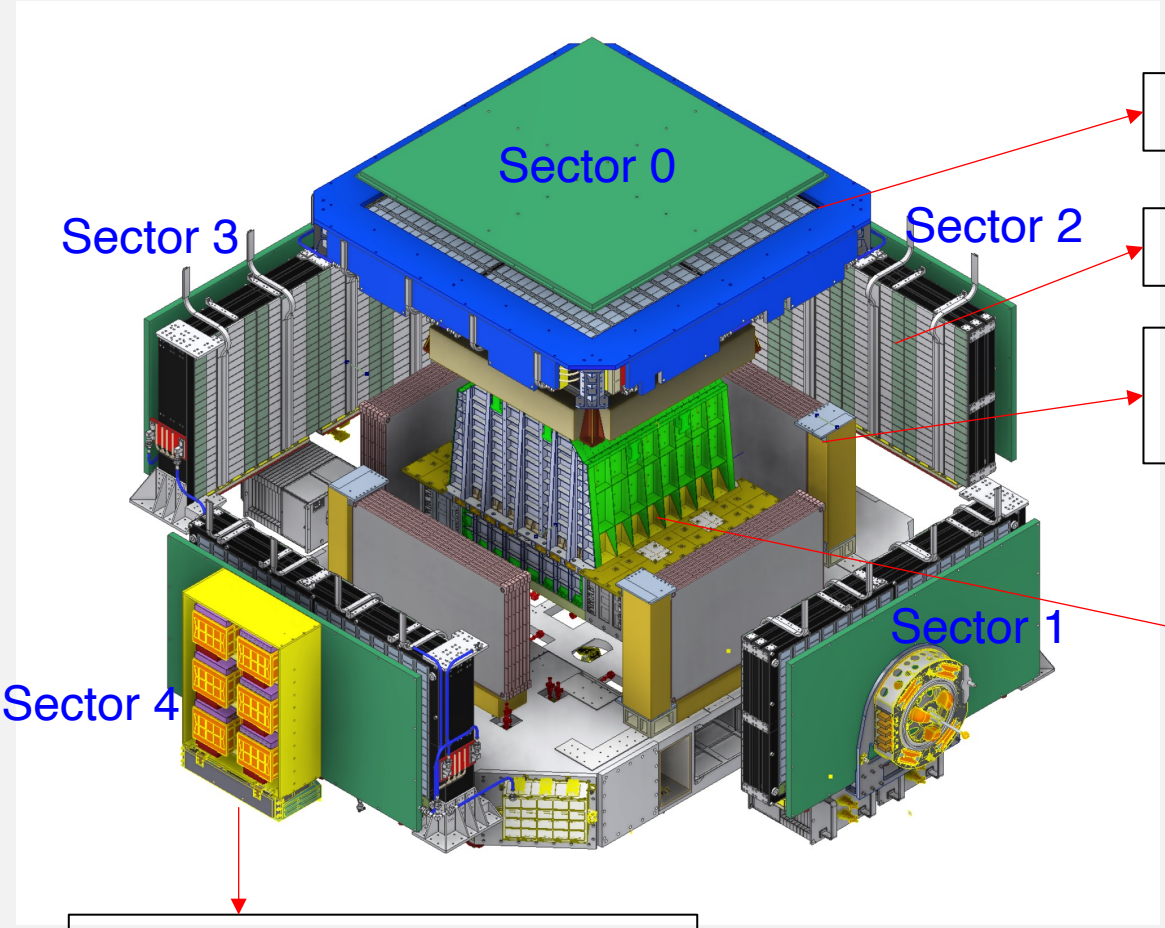
May-8-2025





- General requirement
- Algorithm design
  - Geometry
  - Event data
- WebGL design
  - Geometry rendering
  - Physics data display
- Additional information
  - Reconstruction objects
  - RTI data
- Future Compatibility

HERD Detector



Silicon Charge Detector(SCD) — **Position, Charge**

Plastic Scintillator Detector(PSD) — **Charge, Trigger**

Fiber Tracker(FIT) / Silicon Tungsten Tracker(STK) — **Position, Charge**

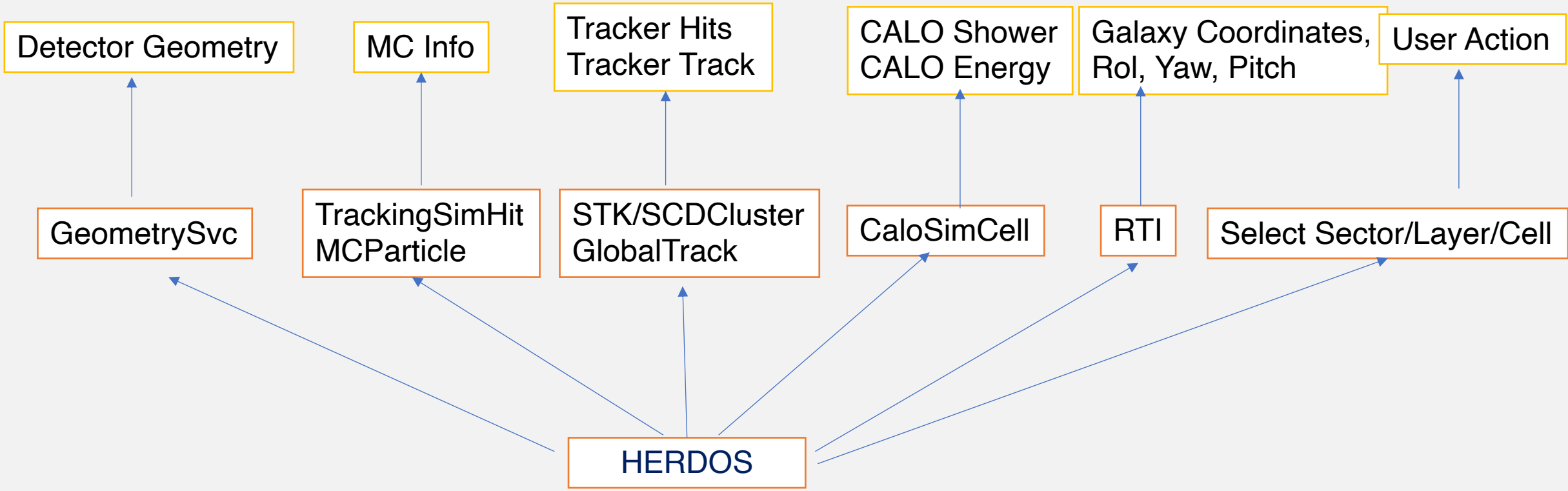
Calorimeter Array (CALO) — **Energy, direction, e/p, trigger**

Transition Radiation Detector (TRD) — **Calibration, e/p**

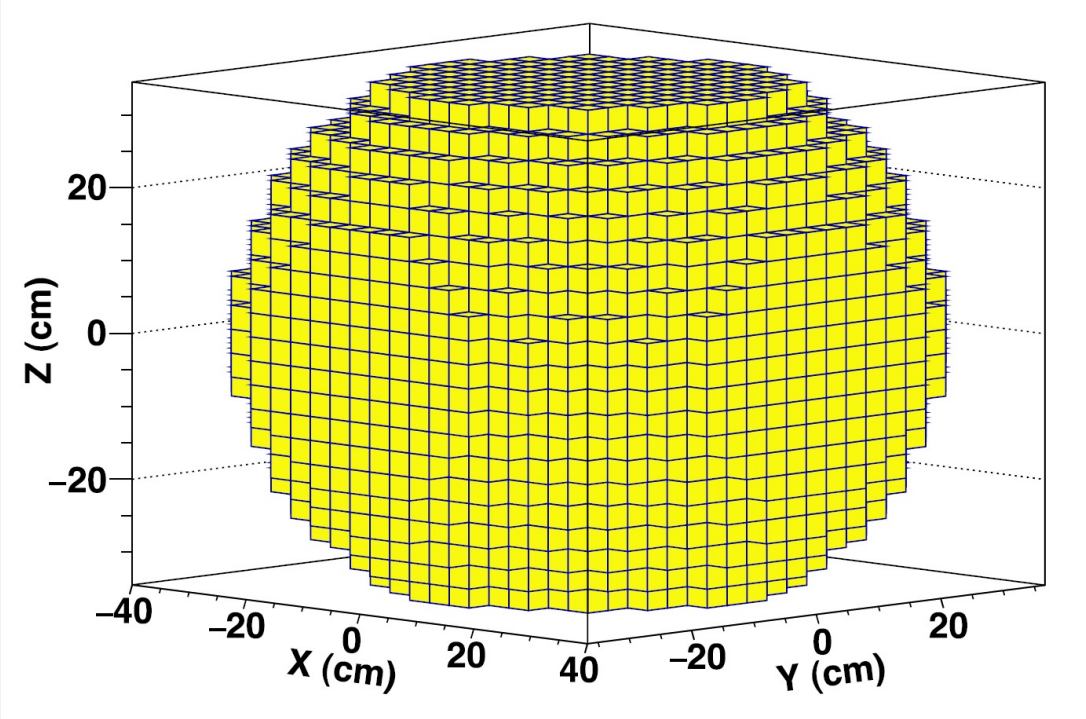
HERD is a compact detector with 5 sensitive sectors.

Algorithm Design

General Requirement



Algorithm design – Geometry-CALO



Schematic of 3-dimensional imaging calorimeter(CALO)

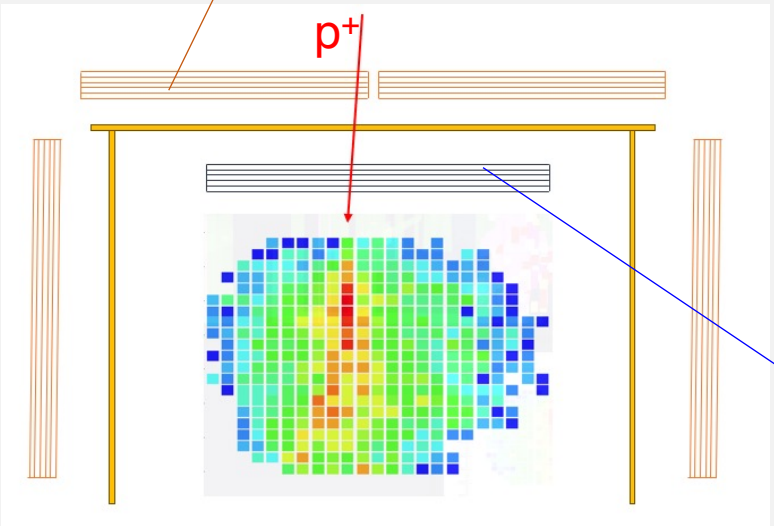
```
auto codemap = geosvc->getCodemap();
for(auto map: codemap){
  for(auto &submap :map.second){
    int sid = map.first;
    int cellCode = submap.first;
    dd4hep::Position position = geosvc-
>getPosition(map.first, cellCode);
    std::vector<double> dim0 = geosvc-
>dimension(map.first, cellCode);
  }
}
```

For CALO, sid = 0.

- 3×3×3cm LYSO cubes
- 23(x)×23(y)×21(z) layers
- Total 7489 cubes

Algorithm design – Geometry – SCD/STK

SCD:
- **5 - sectors,**
- **8 - layers each sector,**
- **Single side readout each sensor.**
- **4-x-readout, 4-y readout.**



STK:
- **top - sector.**
- **14 - layers top sector.**
- **Single side readout each sensor.**
- **7-x-readout, 7-y-readout.**

```
auto codemap = geosvc->getCodemap();
for(auto map: codemap){
  for(auto &submap :map.second){
    int sid = map.first;
    int cellCode = submap.first;
    dd4hep::Position position = geosvc-
>getPosition(map.first, cellCode);
    std::vector<double> dim0 = geosvc-
>dimension(map.first, cellCode);
  }
}
```

For SCD, sid = 4
For STK, sid = 1

Algorithm design – Event Data- MC

- STK/SCD
  - TrackingSimHit
- FIT
  - TrackingSimHit
- PSD
  - TrackingSimHit
- CALO
  - CaloSimCell
- Particle
  - MCParticle

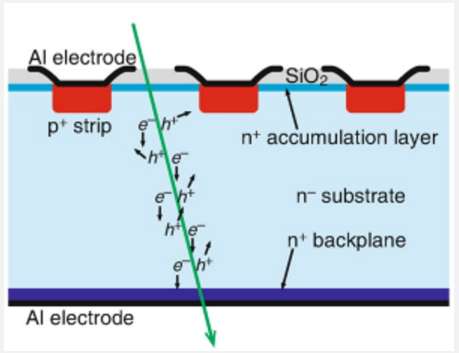
edep, position, direction,
trackID, pdgID, cellCode

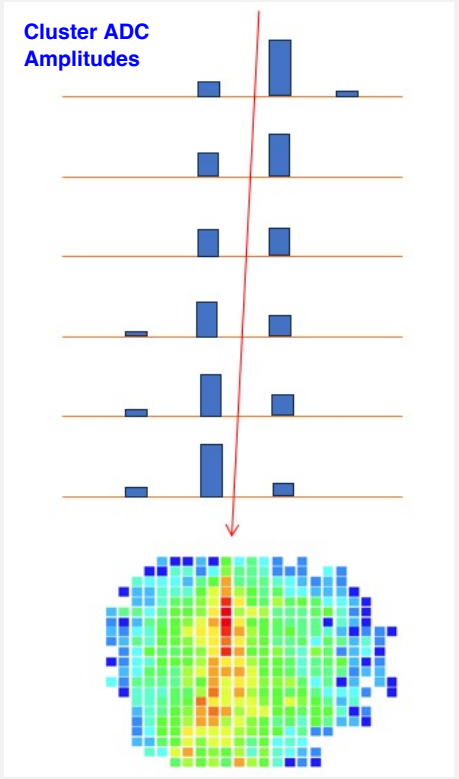
trackID,pdgID, parentID,
EntryPosition, momentum

Algorithm design - Event Data- Cluster

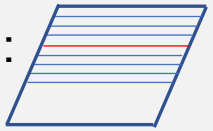
- SCD Cluster
  - "scdclusters
- FIT Cluster
  - "fitclusters"
- STK Cluster
  - "stkclusters"

- Readout info
  - position
  - cellCode
  - charge





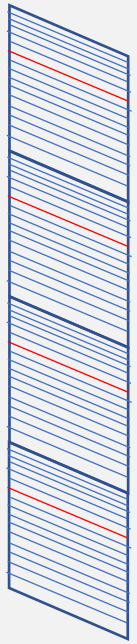
Algorithm design - Event Data - Cluster- Readout Direction

sensor(cell): 

- SCD can only measure one direction.
  - measurement(main) direction (y)
  - aux direction (x)
  - norm direction (z)
- One cluster scale contains
  - precise position of (y,z), main& norm direction
  - limited length of aux direction
  - multiple position for z-ladder

normal ladder:

z ladder:



Example to obtain direction from cellCode

```
SniperPtr<GeometrySvc> ddsvc(GetTask(), "GeometrySvc");
auto xdir = ddsvc->getMainDir(GeometrySvc::SubDetector::STK, 100*cellCode);
auto ydir = ddsvc->getAuxDir (GeometrySvc::SubDetector::STK, 100*cellCode);
auto zdir = ddsvc->getNormDir(GeometrySvc::SubDetector::STK, 100*cellCode);
```

Algorithm design – Event Data – CALO(MC)

Currenttly, only MC info is supported CaloSimCellCollection,
if data do not exist, algorithm will discard CALO part.

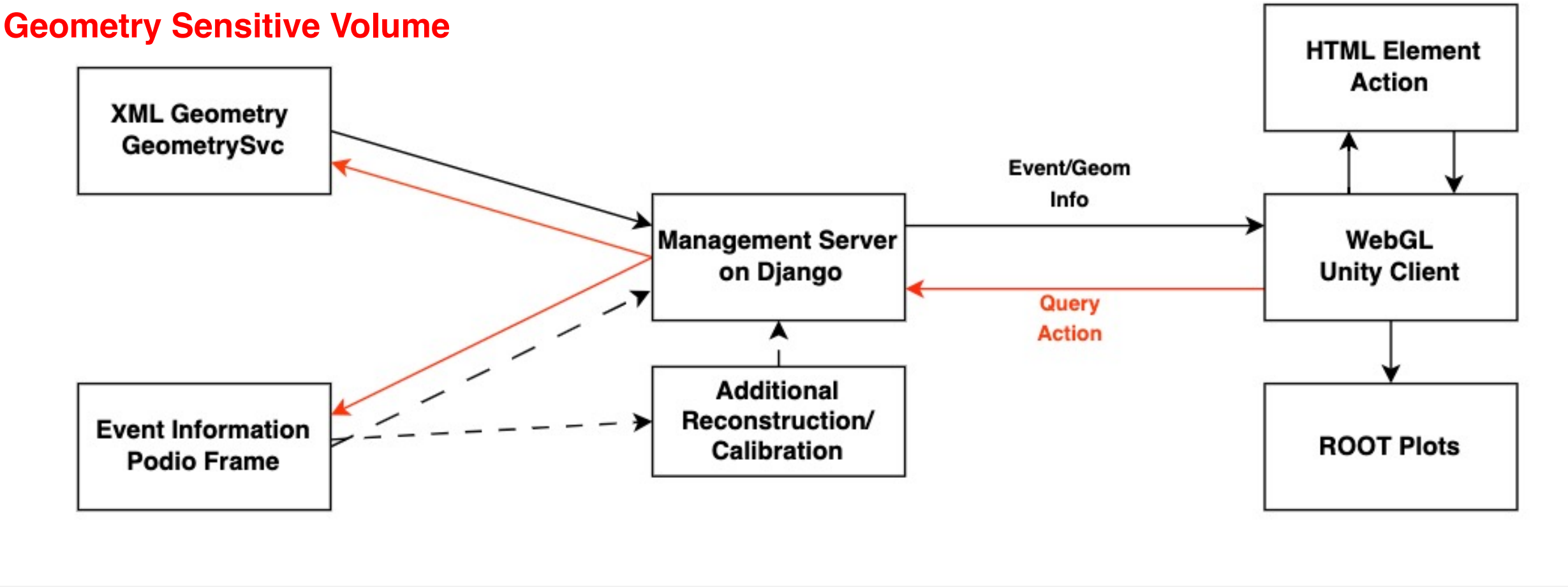
```
DataHandle<CaloSimCellCollection>*  mCALOHandle    = new DataHandle<CaloSimCellCollection> (
        "calohits", GetTask());
    try{
       CaloSimCellCollection  const* calohits = mCALOHandle->getCollRead();
        int ncalo = (*calohits).size();
        for(int ih=0;ih<ncalo; ih++){
            auto  calohit = (*calohits)[ih];
            auto  edep = calohit.getEdep();
        }
    }
    catch(...){
        if(++_ierrprint < 4)
            std::cout<< "..... "<<_ierrprint<<" no CALO MC info \n";
    }
```

Algorithm design – Event Data - Compitability

- All collection names are collected in ZDisplay, standard, non-standard.
- MC:
  - scdhits, scdhits2, stkhits, stkhits2
  - scdinfnhits, scdinfnhits2. (used in beamtest 2023)
  - psdhits, psdhits2, psdinfnhits, psdinfnhits2
  - TRDhits, TRDhits2
  - fithits, fithits2
  - calohits
  - mcparts
- Reconstruction
  - fitclusters, scdclusters, stkclusters, psdreco
  - scdcluster_INFN

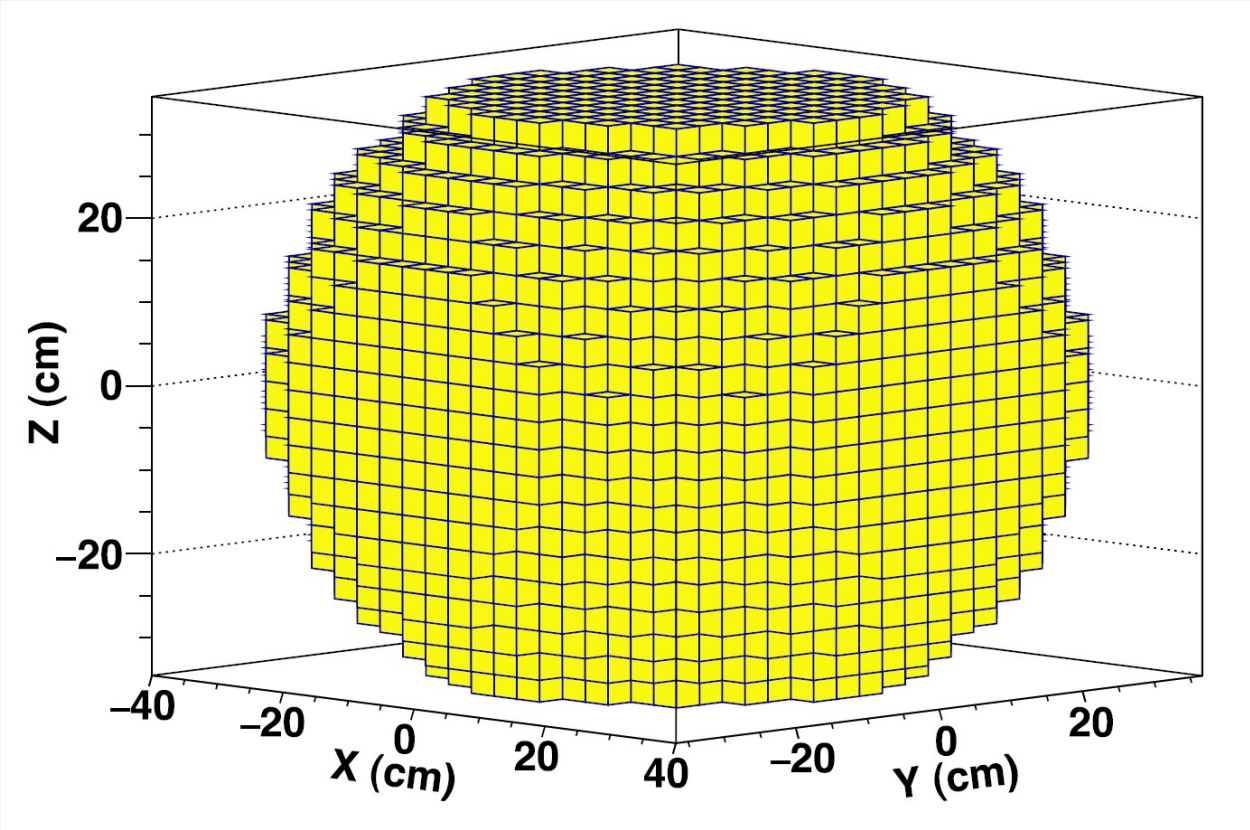
WebGL Design

Event/Geometry Data Flow

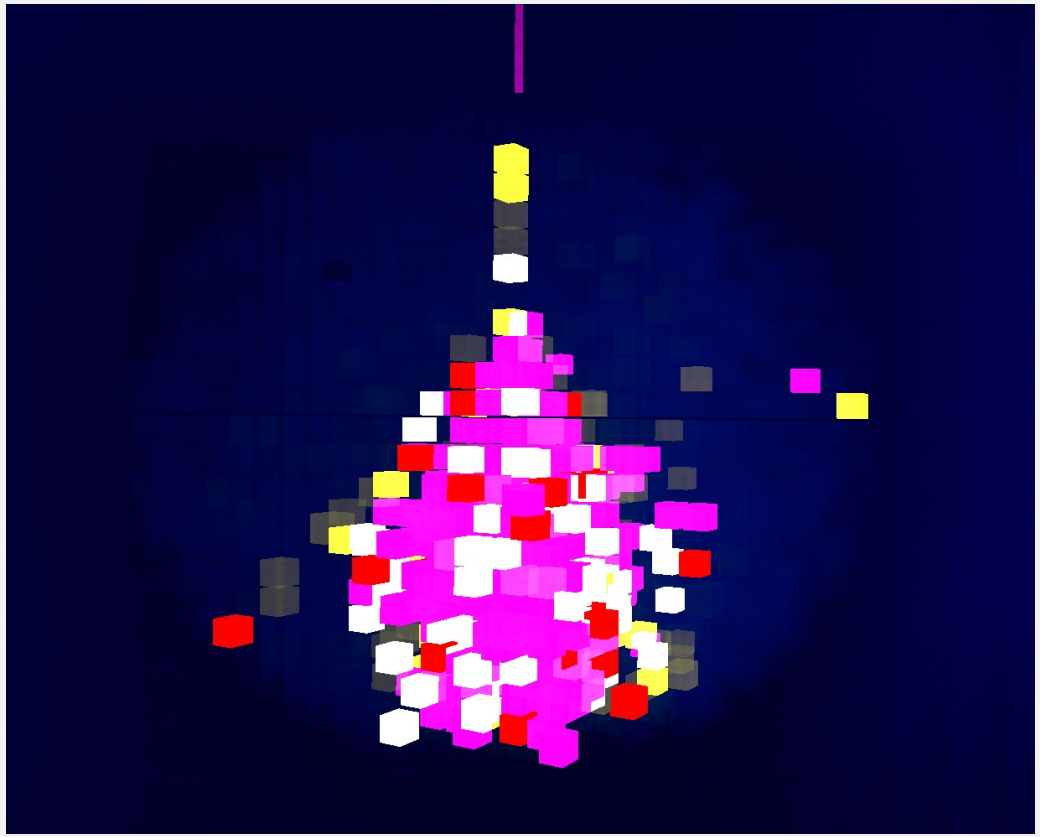


Geometry Rendering - CALO

default root root style

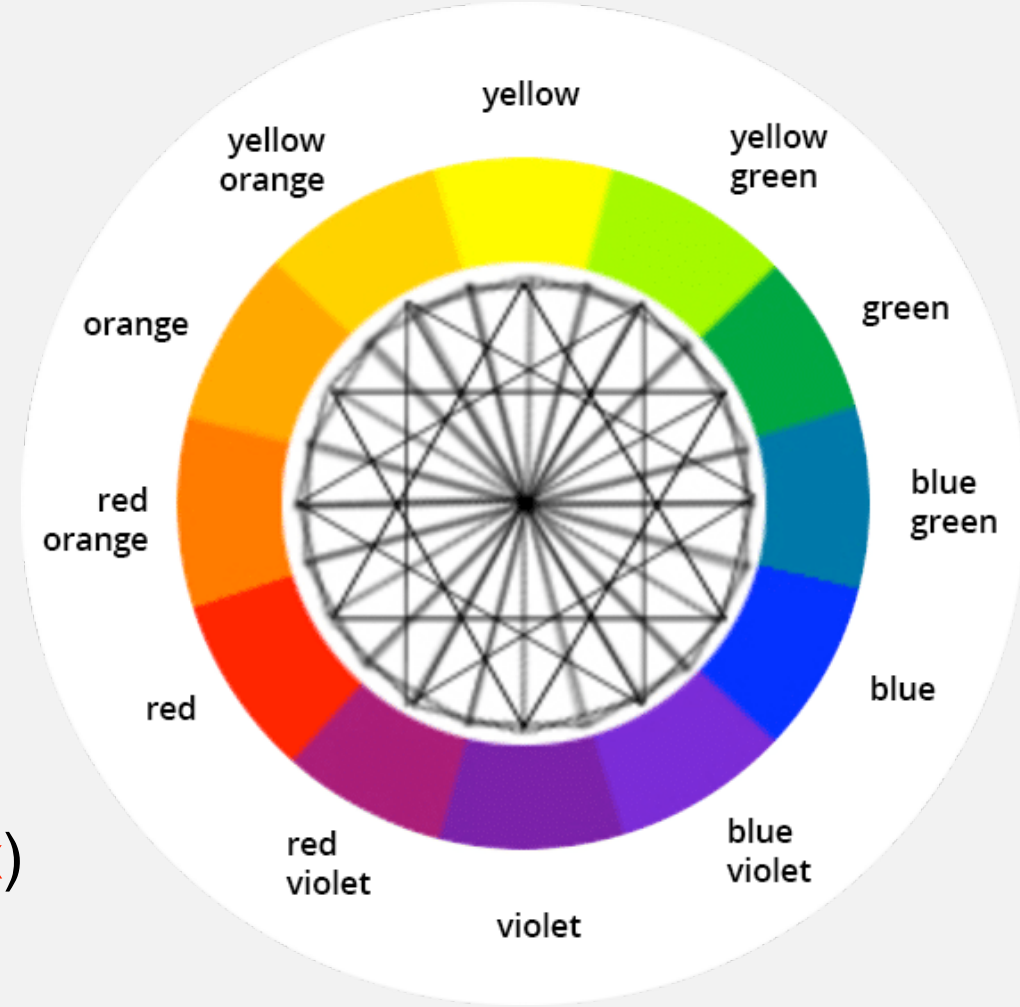


color scale based on energy



Geometry Rendering - CALO – Color

- CALO Cells (MC/Recon)
  - Cell code
  - Cell position
  - Cell energy
- Cell Color based on edep
  - EMin blue
  - EMiddle yellow orange
  - EMax red
  - Divide 10 bins between log(EMin)~ log(EMax)
    - Set corresponding color.
    - Set transparency base on k*log(E)



Unity - Clients

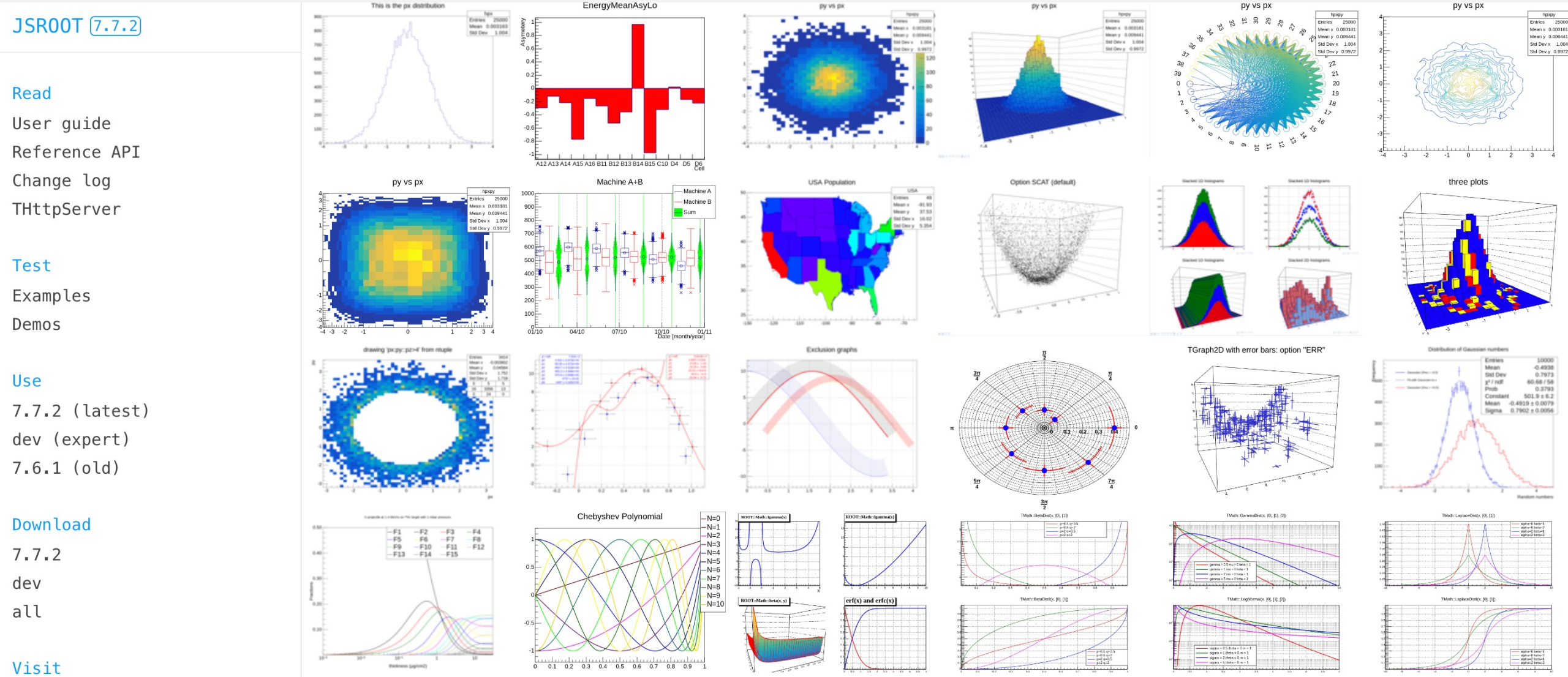
- Unity can be compiled into
  - Window
  - MacOS
  - Linux
  - WebGL
  - Android, iOS
- However, plots in unity is difficult.
  - Choose WebGL as main client.
    - rendering geom
    - display event info
    - json encode/decode
    - communication with django server
  - jsroot (root in javascript) as plots

Unity will generate wasm for WebGL:

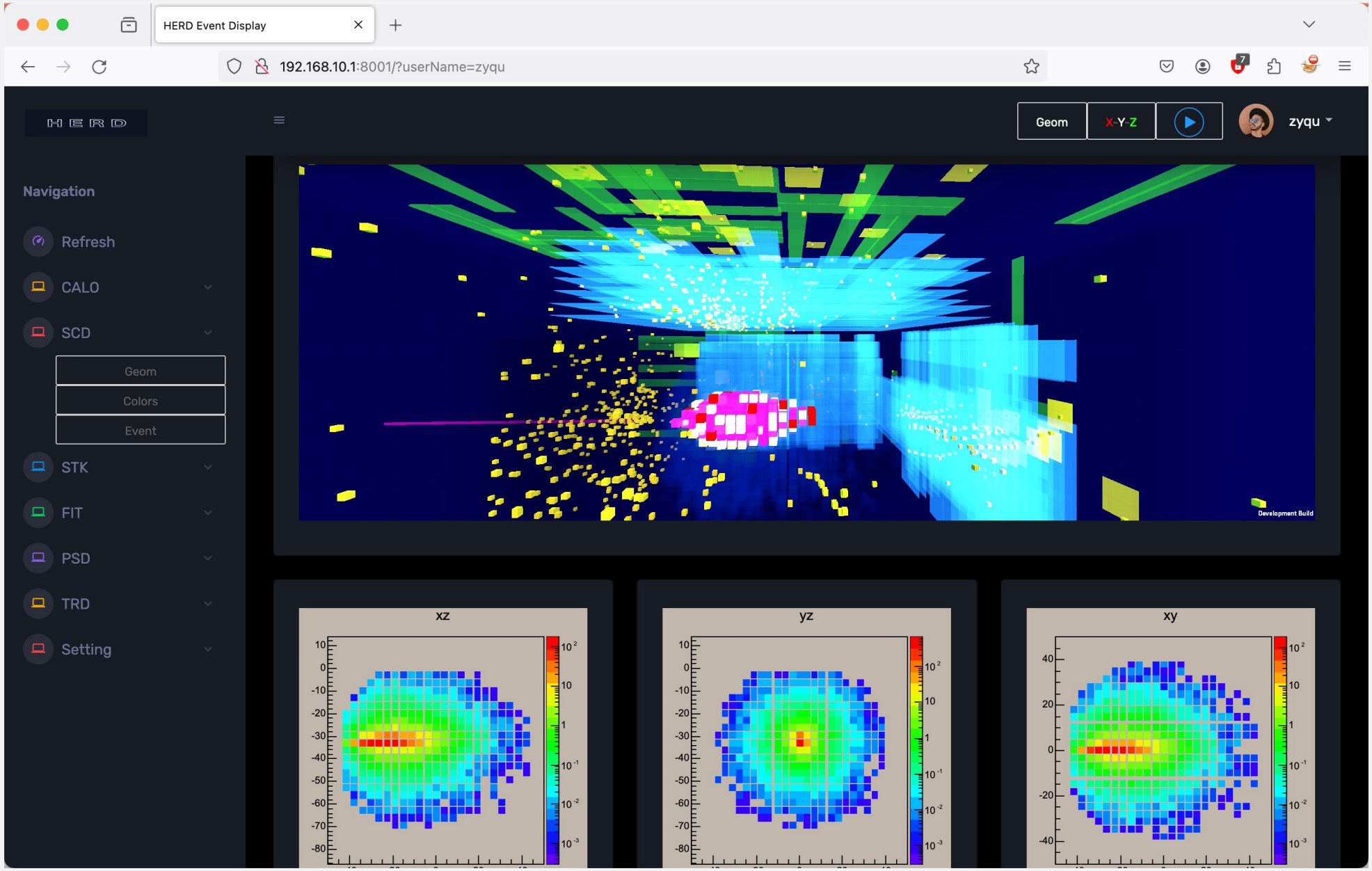
- Binary file
- Fast as original
- Use GPU automatically
- Optimized for CPU

WebGL & Plots

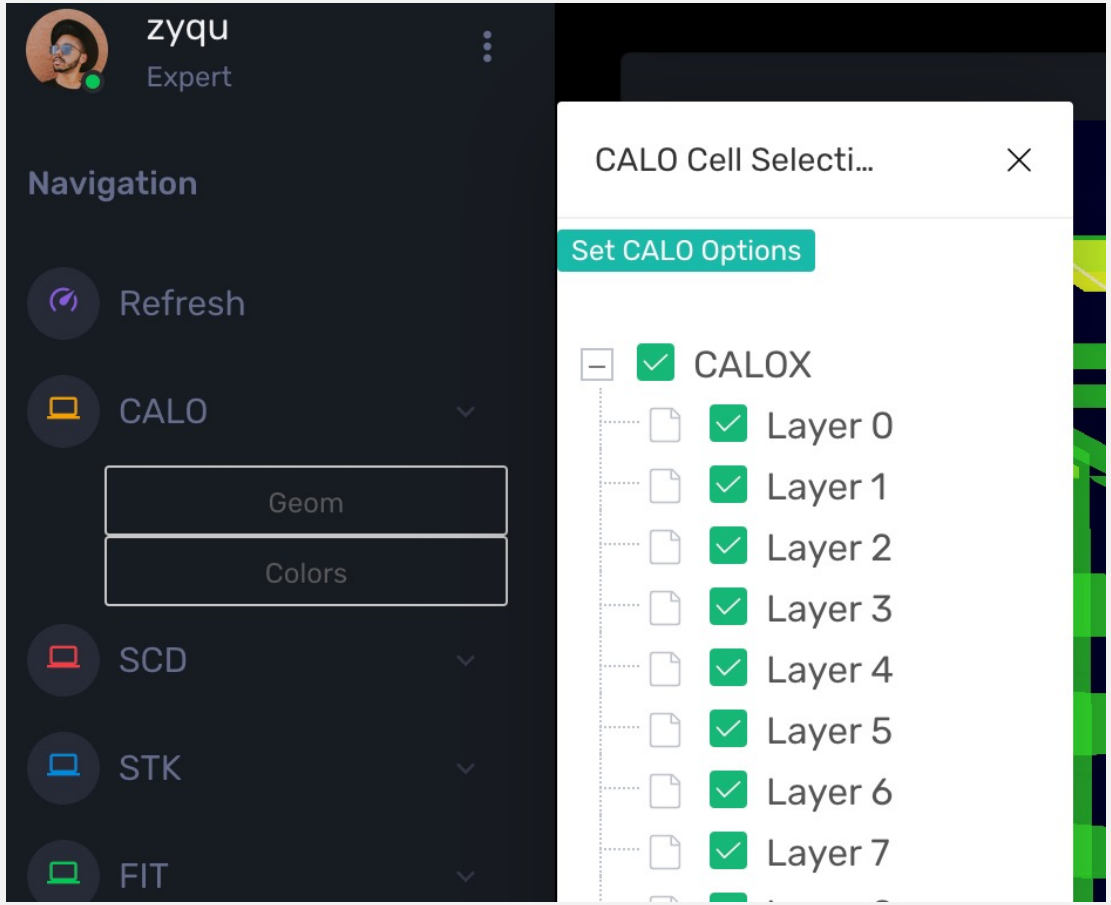
- Use jsroot to generate plots

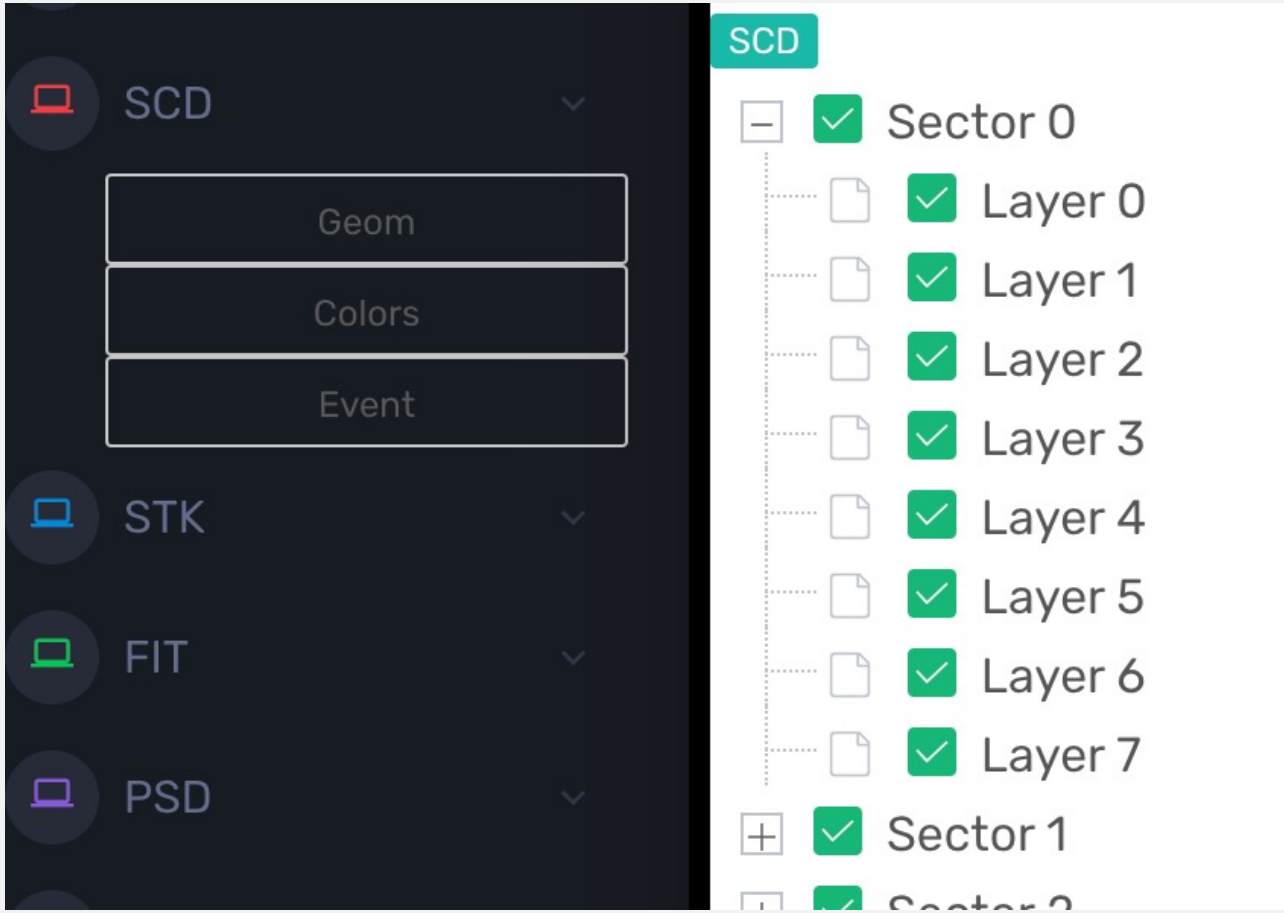


WebGL & Plots


HERD Event Display
192.168.10.1:8001/?userName=zyqu
HERD
Geom
X-Y-Z
zyqu
Navigation
Refresh
CALO
SCD
Geom
Colors
Event
STK
FIT
PSD
TRD
Setting
Development Build
xz
yz
xy

WebGL Example – Geom Selection

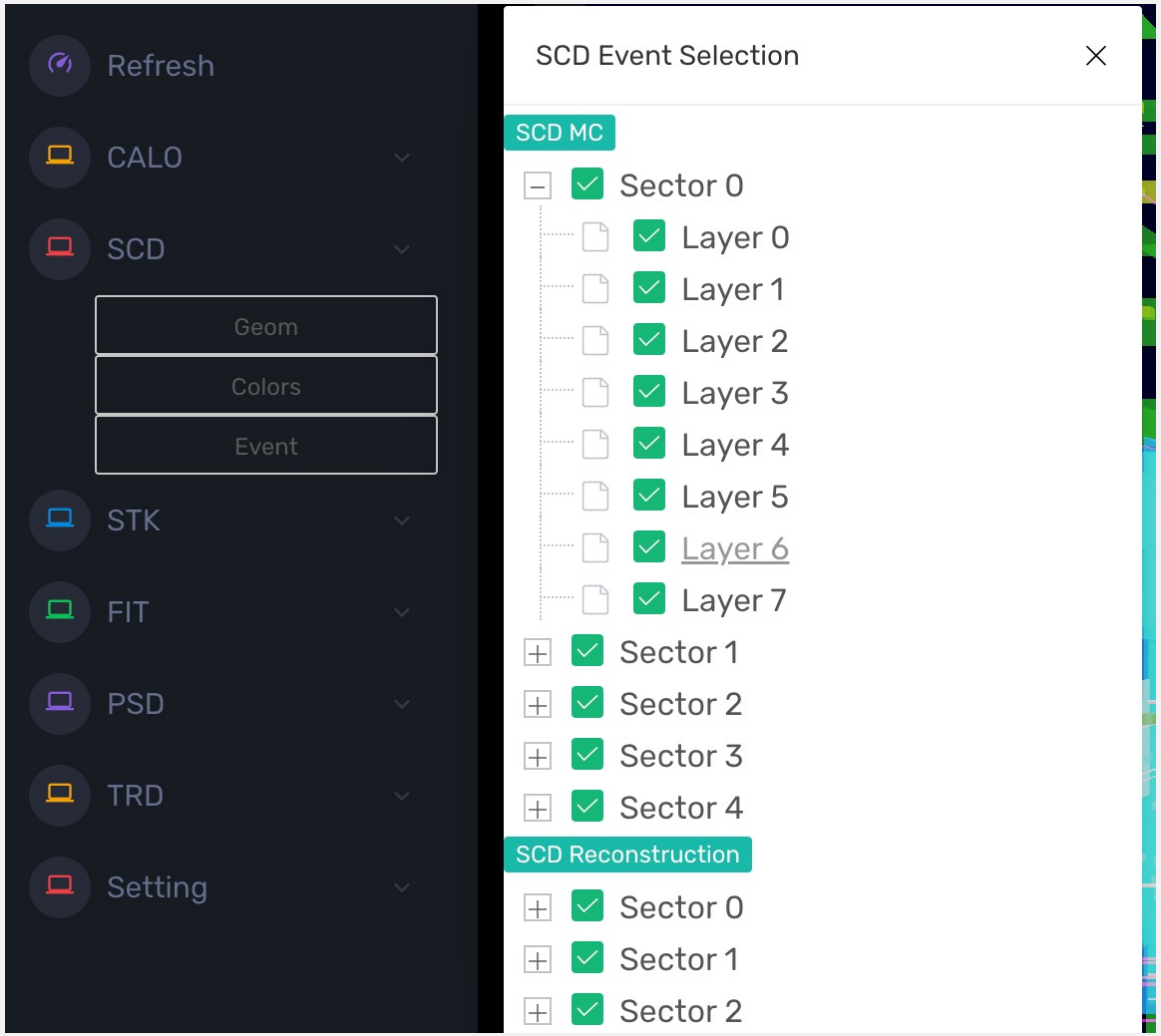




- Show/Hide specific x-layer on CALO
- 23(x)×23(y)×21(z) layers

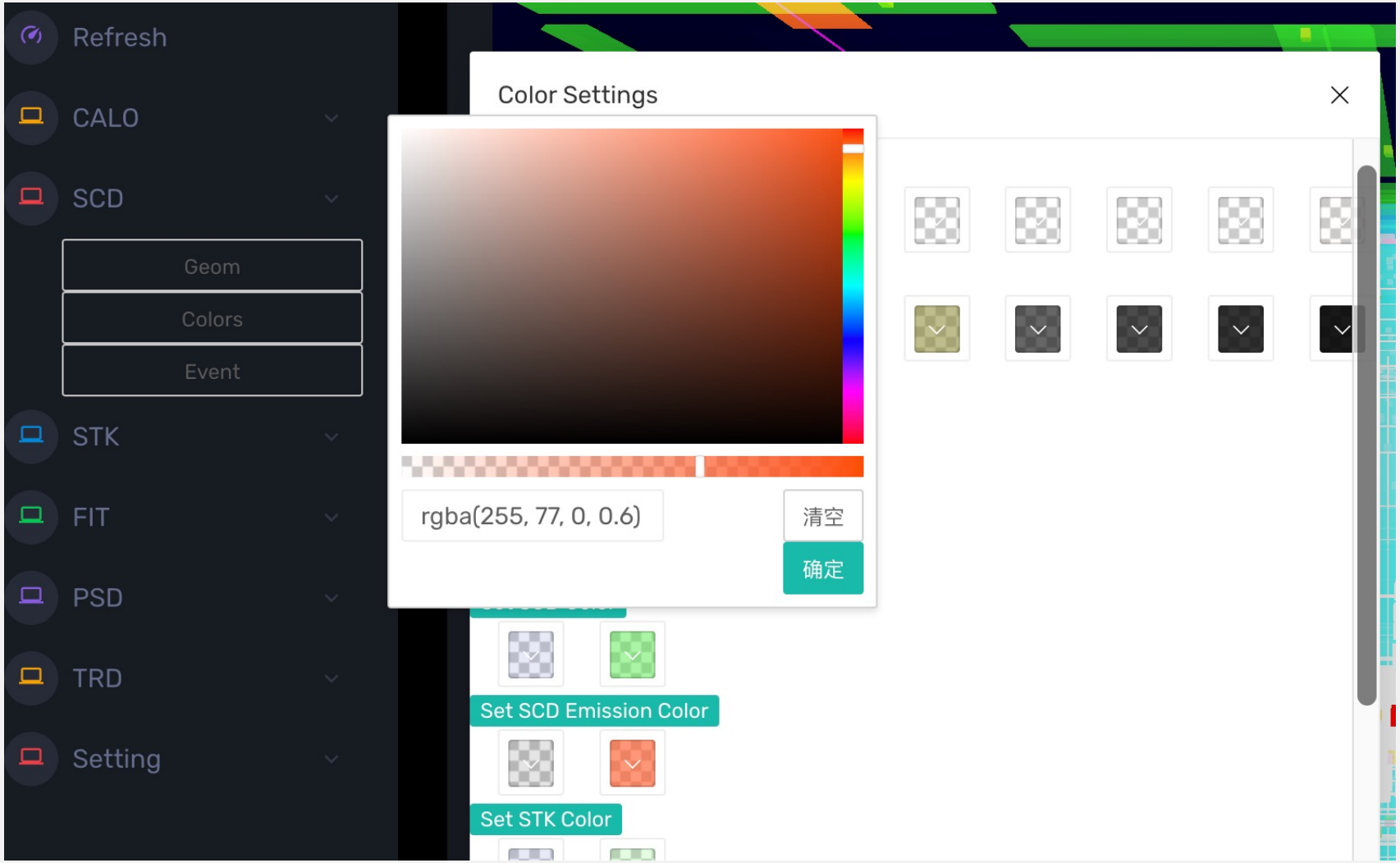
- Show/Hide specific sector/layer on SCD

WebGL Example – Event Data Selection



- Show/Hide MC hits on specific sector/layer of SCD
- Show/Hide Cluster on specific sector/layer of SCD

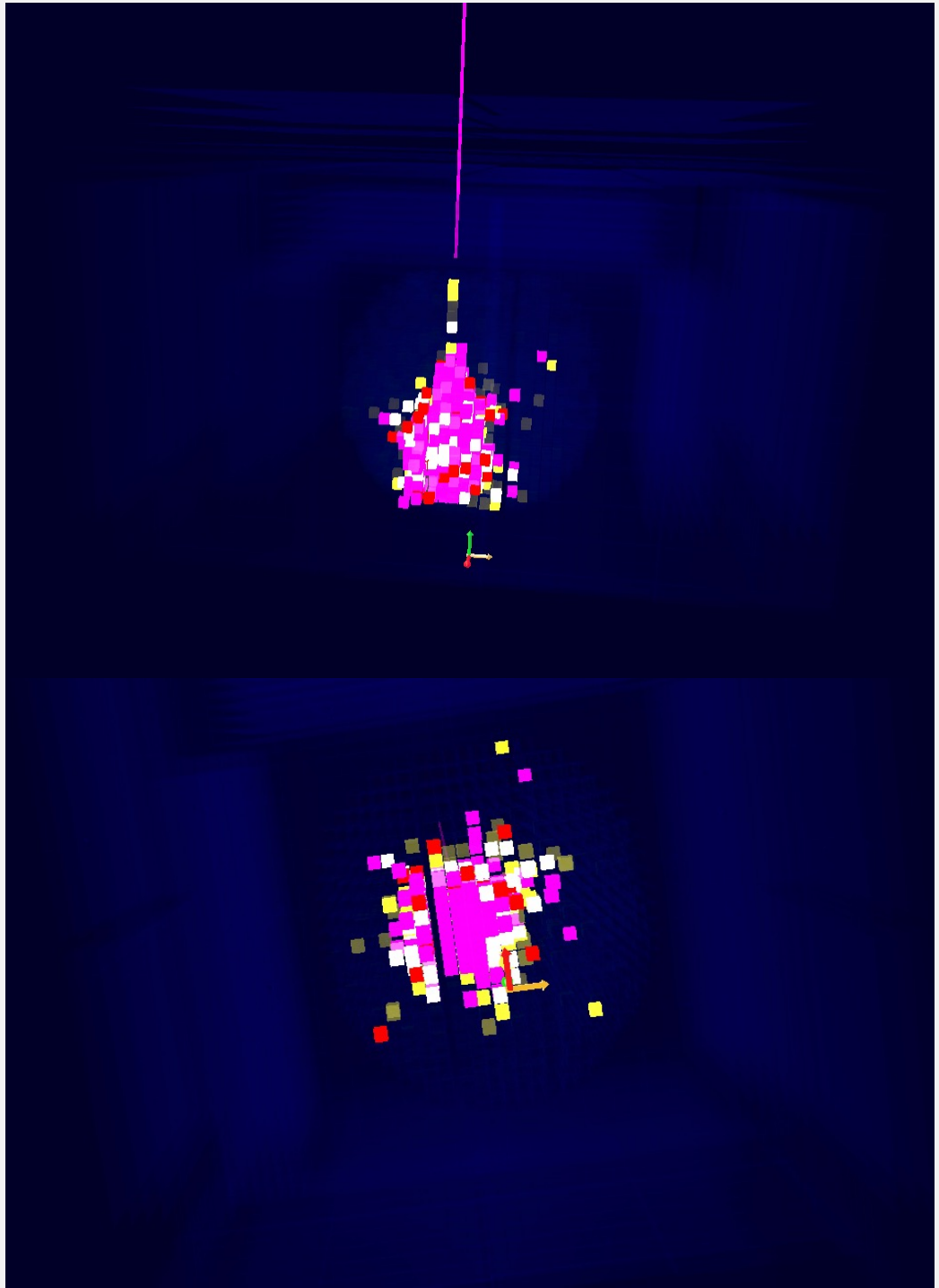
WebGL Example – Set Color



- Set Color for SCD/STK/FIT
- Set Color/Transparency for CALO on different energy bin
- Set Color for Background

WebGL Example – Move/Rotate/Zoom

- Move
  - Left button drag
- Rotate
  - Right button drag
- Zoom
  - Middle button



Front View

Bottom View

Compitiblity & Future development

- Compitiblity
  - ZDisplay generate 2 files in home directory
    - ~/.herd_geom.json
    - ~/.herd_event.json
  - Any other way of loca display using ROOT/blender/VPython can use files directly.
- Future development
  - Combine RTI information
    - CSS **real time location**
    - CSS **solar panel shading**
  - Particle galaxy coordinate
  - Reconstruction objects
    - PCA direction, shower 3D fit, machine learning direction
    - Other high level reconstruction objects.

How to Run ZDisplay

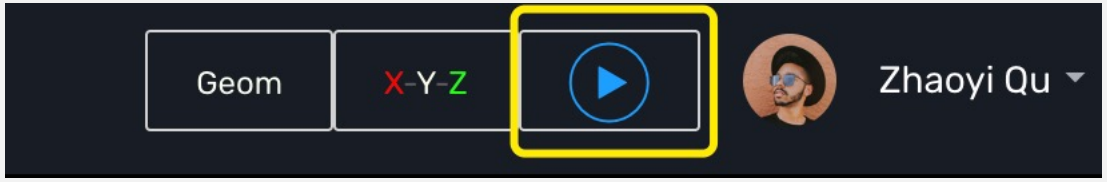
Running Mode1. Event by Event

- Display event by event

```
zdisplay --geometry $HEDOS_INSTALL/compact/v2022a/v2022a-test.xml
         --input sample.root   --entry 5
```

zdisplay: binary file in $PATH
--geometry: geometry xml file
-- input: root file to be display
-- entry: entryID to start display

Display next event



RUNNING MODE 2. Monitor

- Run event, automatically update next event by n seconds

```
zdisplay --geometry $HEDOS_INSTALL/compact/v2022a/v2022a-test.xml
         --input sample.root   --entry 5 --seq 3
```

zdisplay:	binary file in $PATH
--geometry:	geometry xml file
-- input:	root file to be display
-- entry:	entryID to start display
--seq:	show next event by n second

Loop whole sampel.root file, from entry 5 to end, with interval of 3 seconds.

RUNNING MODE 3. Special Event in Production/Analysis

- Some events are more attractive
  - High energy event (> TeV)
  - MIP particle
  - other selection cuts
- Assert c++ code in your Production/Analysis Loop:
  - Show this event

```
ZDisplayAlg::showThisEvent(true);
```

  - Do not show this event

```
ZDisplayAlg::showThisEvent(false);
```

- Assert python code in your scripts

```
import ZDisplay
disalg = task.createAlg("ZDisplayAlg")
```

Recent Updates

Unity ->ThreeJS

New version of Unity is banned in China

According to machine translation, the company said in a statement "**Unity 6 and subsequent versions will no longer be provided to Chinese users**" in an "adjustment aimed at ensuring that developers can obtain game engine services that are more in line with the needs of the Chinese market. 17 Apr 2025

- Migration is finished and deployed in IHEP and SDIAT.
- Performance is better.
- Multi-platform is achieved by Electron framework.

All modules are open-sourced

Conclusion

- General understanding of HERD detector
- Data revision package development in HERDOS framework
- Read event and geometry data to json
- Render event and geometry data
- Migration from Unity to ThreeJS
- Deployed in M.P.O, IHEP, SDIAT