3D Visualization of Muon g-2 data and simulation

Student: Marco Di Bella
Supervisor: Adam Lyon

Final presentation

09/22/2016

ROMA TRE
UNIVERSITÀ DEGLI STUDI

Overview

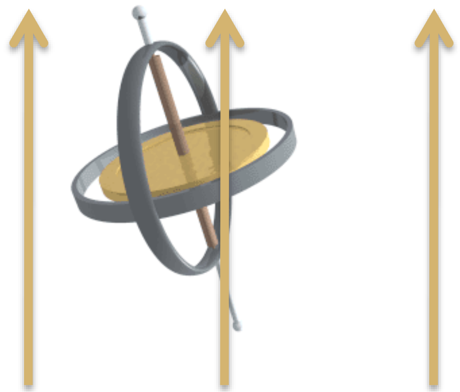
- The experiment Muon g-2
- Why Visualization
- Art framework
- gm2vtk
- Conclusion

Reason for Muon g-2

- For an elementary spin $1/2$ particle in Dirac's theory, g=2!

- Until precision measurement was done by Kusch and Foley in 1948 the electron magnetic moment was is good shape with Dirac's new theory, but then:

$$g_e = 2,00238(6)$$



- Thus the anomalous magnetic moment was discovered, fractionally g differs from 2 by (g-2)/2 = 0,1%

- This difference was explained supposing that virtual particles continually fluctuate in and out of the vacuum.

$$\omega_s = g\frac{eB}{2mc}$$

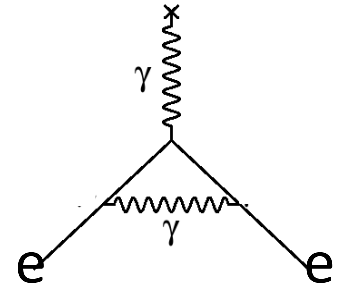
- The extent to which g differs fractionally from 2 is what we call the anomalous magnetic moment

$$a_\mu = \frac{g-2}{2}$$

Reason for Muon g-2

- **Schwinger takes one look at the anomaly in the g-factor and immediately knows what's up**

$$g_e \approx 2\left(1+\frac{\alpha}{2\pi}\right) \approx 2,00232$$



Schwinger term describing 1^{st} Order electron self-interaction

- **Calculation agrees well with experiment, and that is how we build confidence in new physics models!**

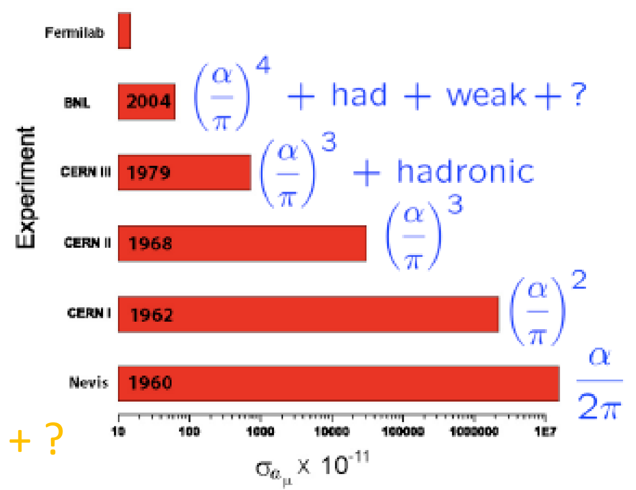
- **Adding precision to both the experimental and the theoretical value of g let us see the evidence of more and more partners emerging from the vacuum:**

$$g_{expt} = 2,00233184178$$

QED contribution

QCD contribution

Weak contribution + ?



Principles of Muon g-2 Expt

- **The Spin frequency relative to the Cyclotron frequency is the «anomalous precession frequency», ω_a :**

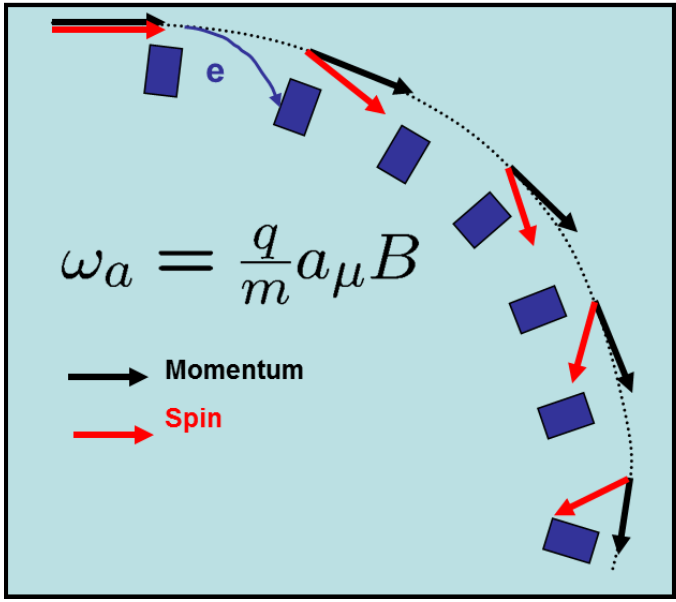
$$\omega_a = \omega_s - \omega_c = \left(\frac{g-2}{2}\right)\frac{eB}{mc}$$

- **Proportional to g-2 and B!**

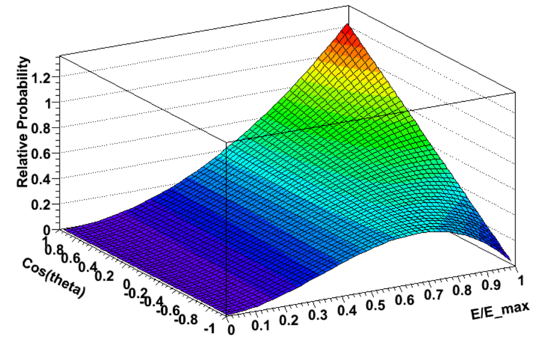
- **For vertical focusing reason it is needed also a E field which looks like a B field to a moving particle:**

$$\vec{\omega}_a = -\frac{e}{m}\left[a_\mu \vec{B} - \left(a_\mu - \frac{1}{\gamma^2 - 1}\right)\frac{\vec{\beta} \times \vec{E}}{c}\right]$$

- **Choosing $\gamma = 29,3$ ($p_\mu = 3,094\ GeV/c$)**

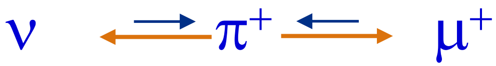


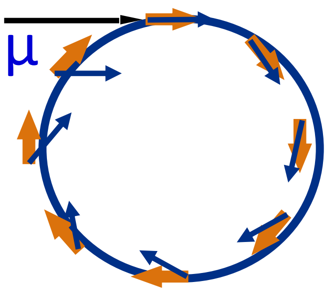
Principles of Muon g-2 Expt



- **Four key elements:**

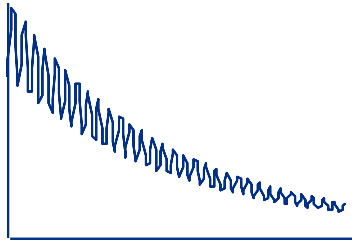
- **Polarized muons**
  - $\sim 97\%$ **polarized for forward decays**





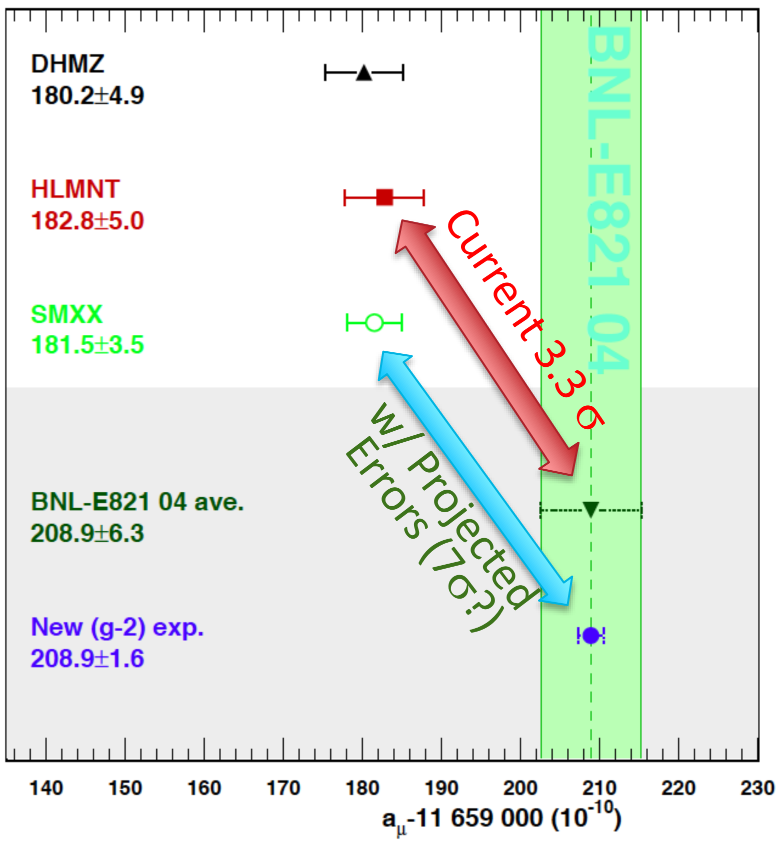
- **Precession proportional to (g-2)**

- P_μ **magic momentum = 3,094 GeV/c**



- **Parity violation in the decay gives average spin direction**

Muon g-2 experiment



$$a_\mu = \frac{\omega_a/\omega_p}{\mu_\mu/\mu_p - \omega_a/\omega_p}$$

In order to have a precise measurement we do not evaluate a_μ directly from the equation $\overrightarrow{\omega_a} = -\frac{e a_\mu \vec{B}}{m}$ but we use the ratio between the anomalous magnetic moment and the Larmor frequency of the proton.

The g-2 experiment has the goal to obtain a precision of 140 ppb which is much smaller than the previous one obtained by BNL. With such a precision and with the actual theoretical value we can reach a 5.6 or 7σ (a Discovery!)

Why Visualization?

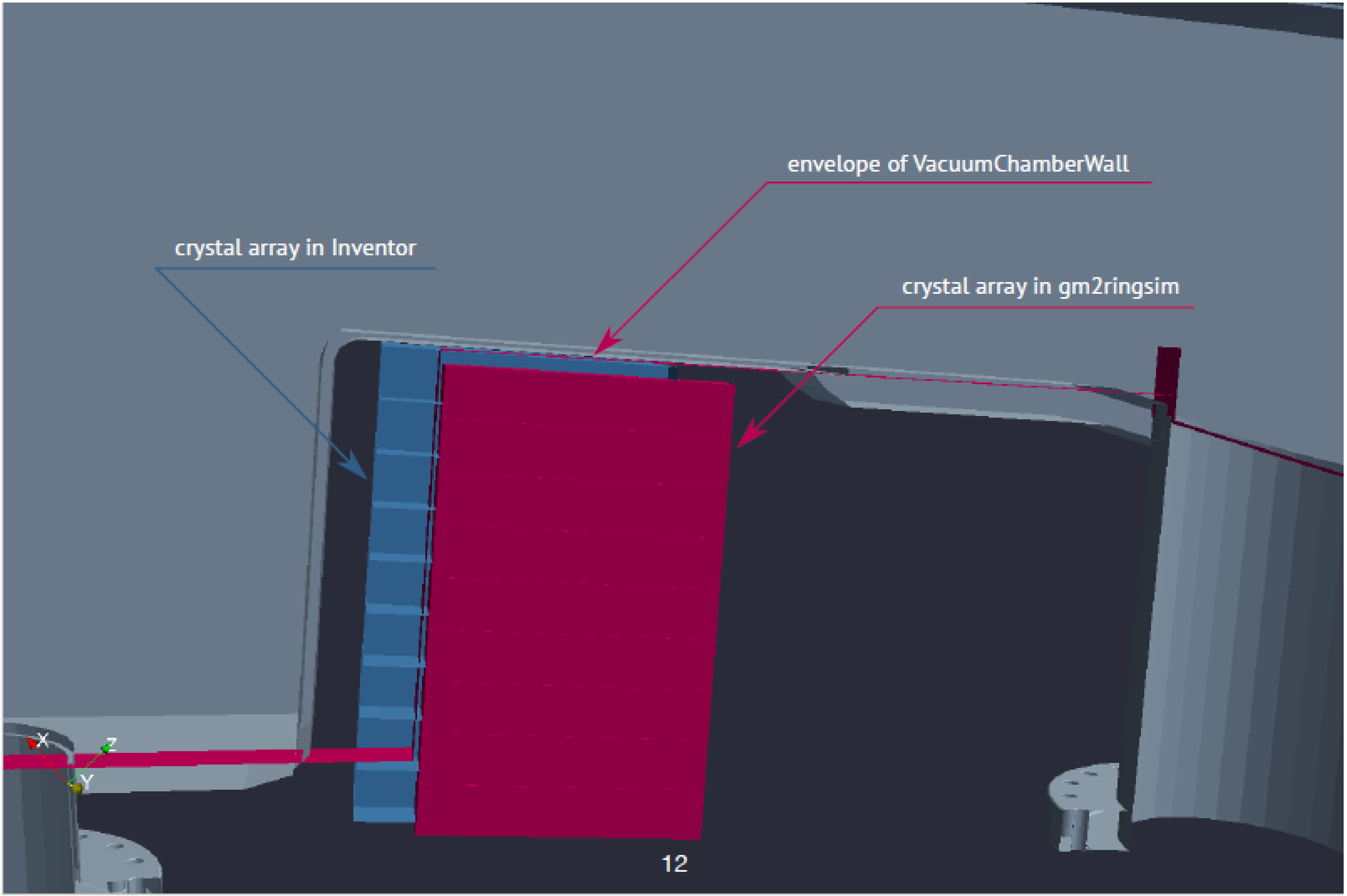
Muon g-2 was facing several problems where visualization could be very helpful

- **Validation of our Geant geometry and simulation**
  We had hints of incorrect positions in the geometry

- **Debugging of Magnetic fields in Geant**
  We have some complicated and time-varying fields (kicker magnets) – needed verifying

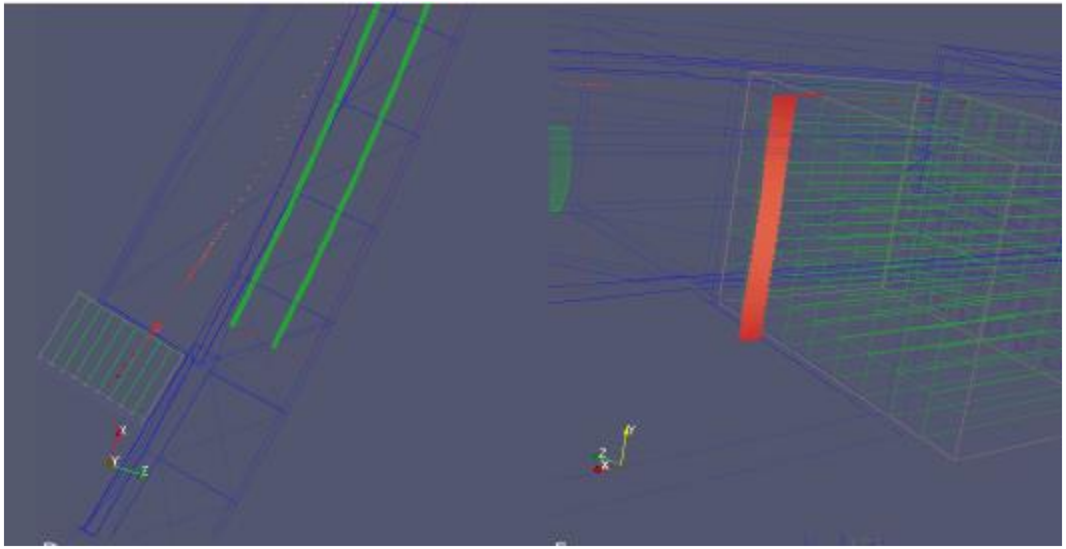
- **Debugging Tracking (the usual stuff)**
  Comparison of reconstructed hits & tracks to truth hits & trajectories – needs to run post-grant

Visualization



Visualization

- **A “stuck” Geant events (hard to debug since output never arrives)**



Visualization

- An important tool already used in public relations (pretty displays, movies, virtual reality), analysis (event scanning) is the visualization (simulation geometry verification).
- To display event in 3D allows us to very quickly understand what is going on in the data or simulation.

- What I have done is to write an Art VTK translation layer for popular art object.

- I have written codes to connect directly the Art modules with a visualization program called Paraview:

- It is a scientific visualization application with rich capabilities
- Free
- Uses advanced visualization solutions to maintain a good user experience
- Based on established VTK library

Art framework

A framework let you write physics code without worrying about the infrastructure.
It is useful because:

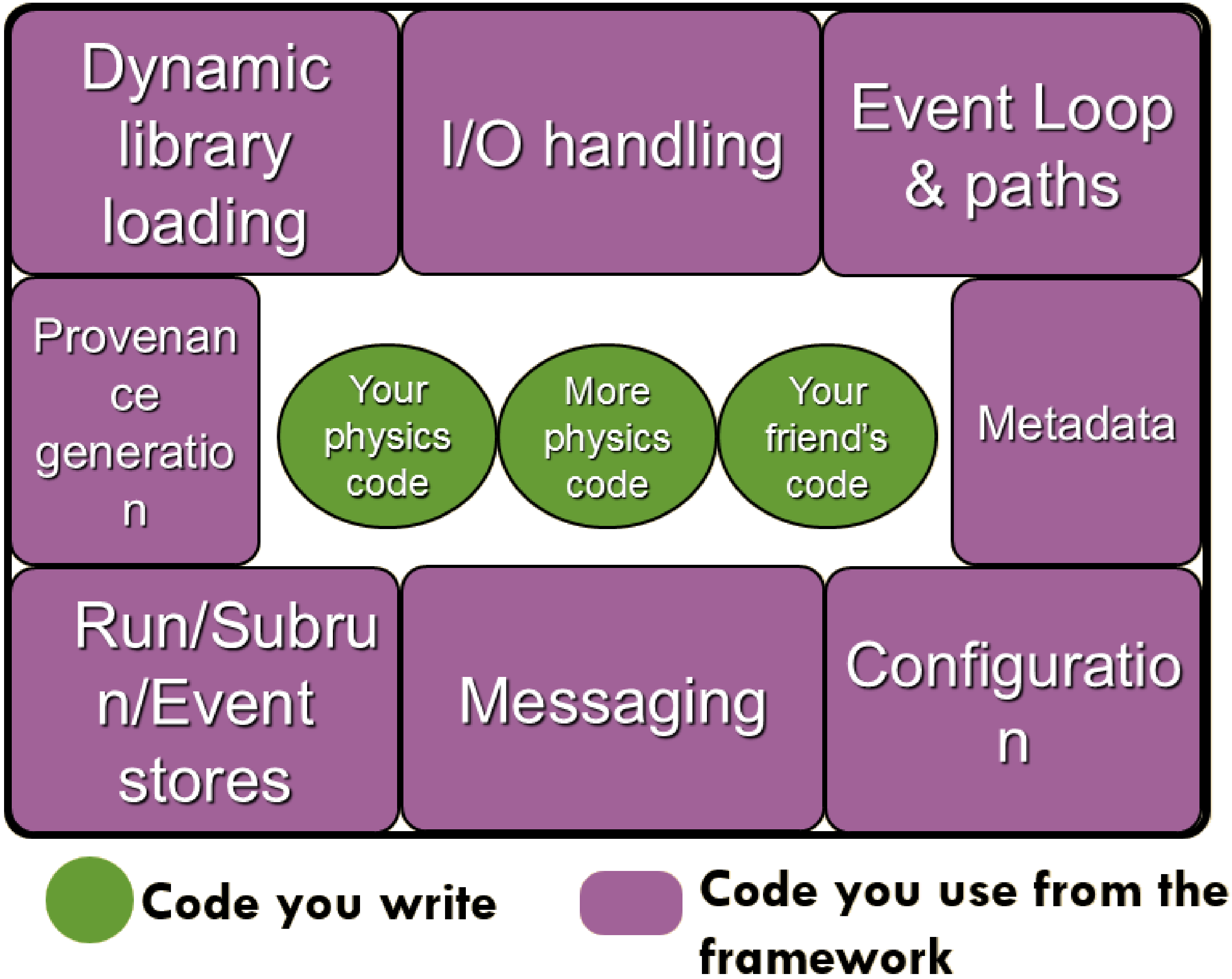
- It makes easier to work in group, to share ideas.
- You do not have to write super complicated C++ code
- You can have fun with plots and physics stuff without writing infrastructure code
- It is Modular (you write modules that piece together)

Why using Art?

- It is already used by most of the Fermilab Intensity Frontier Experiments (Nova, Mu2e, MicroBoone, DUNE)

- Built in Root i/o

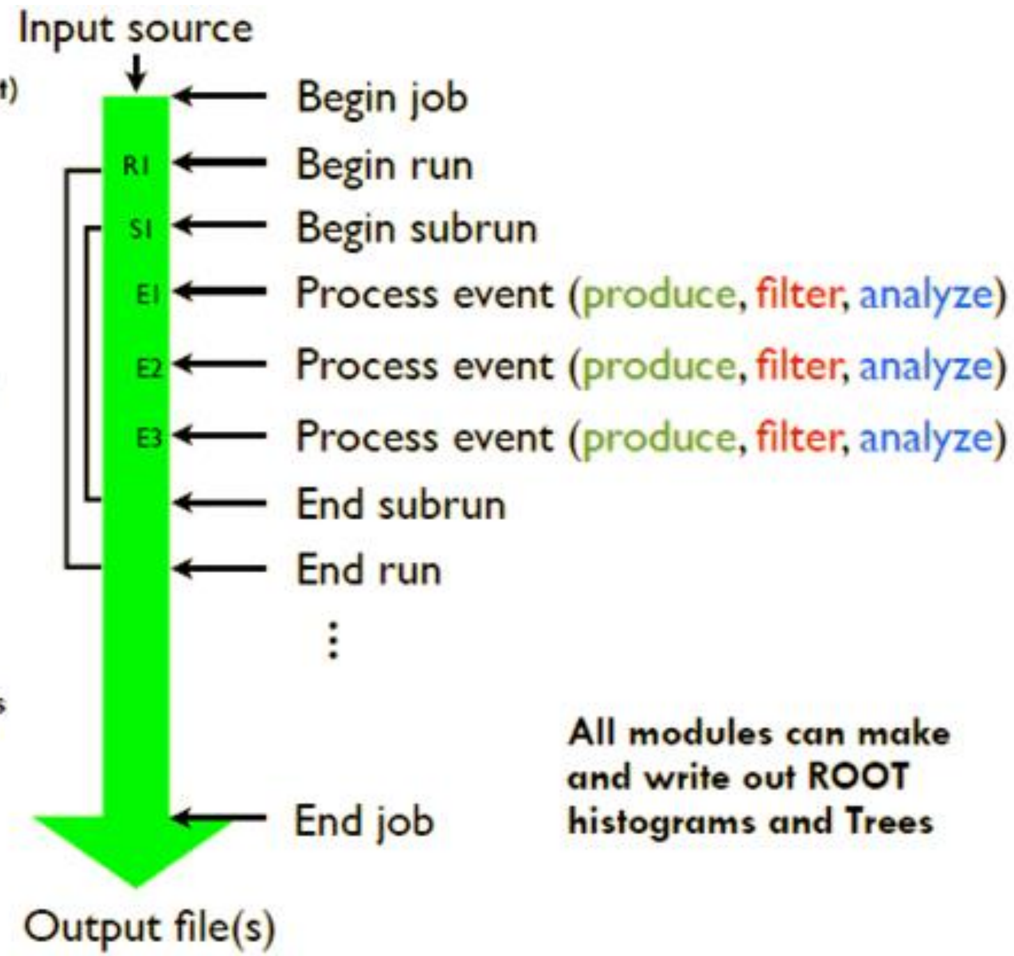
What does a framework do?



Art Modules

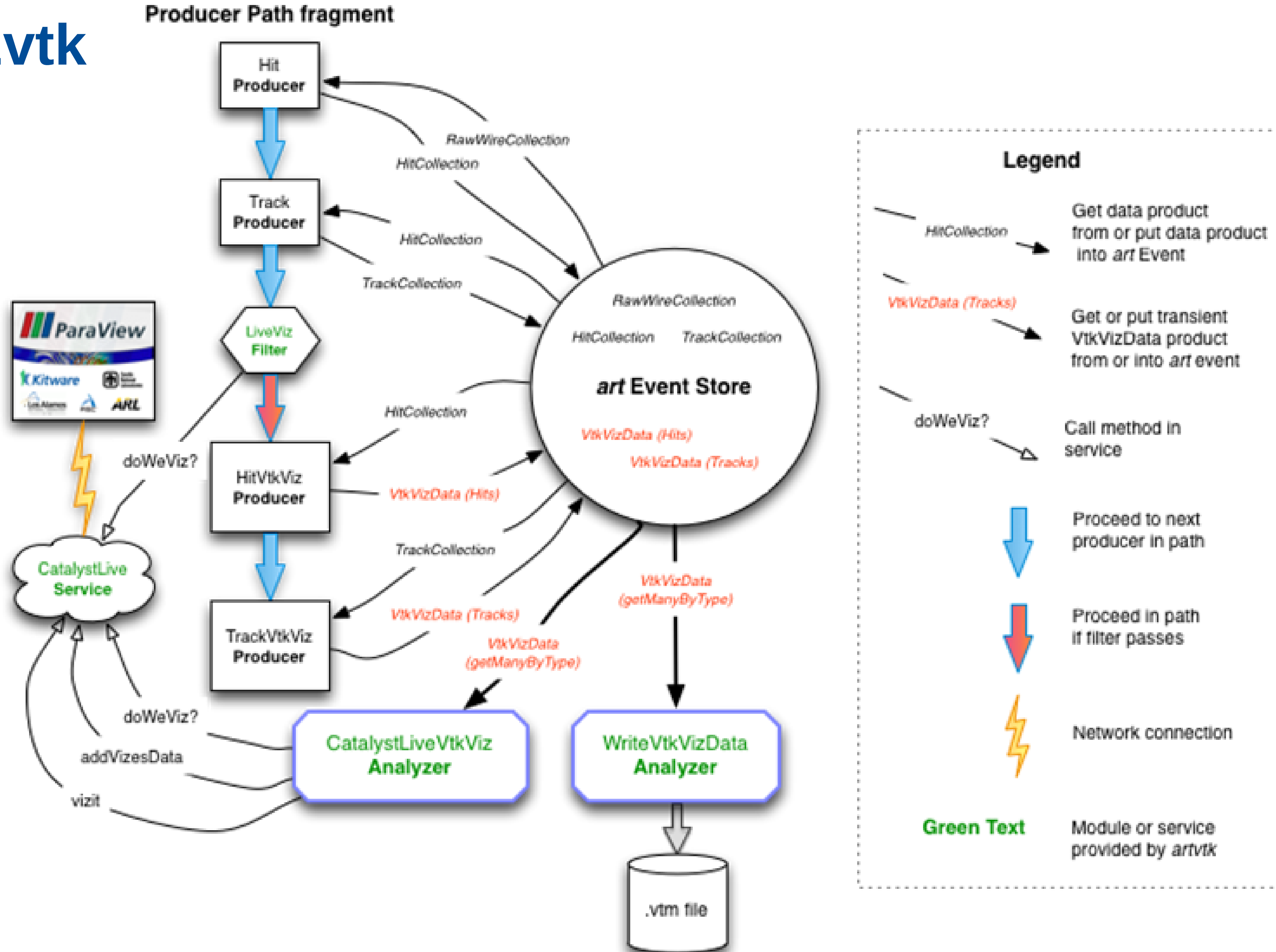
<u>Types of MODULES:</u>
(All modules can read data from the event)

- **Input source:**
  A source for data. E.g. a ROOT file or Empty for start of simulated data
- **Producers:**
  Create new event data from scratch or by running algorithms on existing data
- **Filters:**
  Like producers, but can stop running of downstream modules
- **Analyzers:**
  Cannot save to event. For, e.g. diagnostics plots
- **Output module:**
  Writes data to output file (ROOT). Can specify conditions and have many files

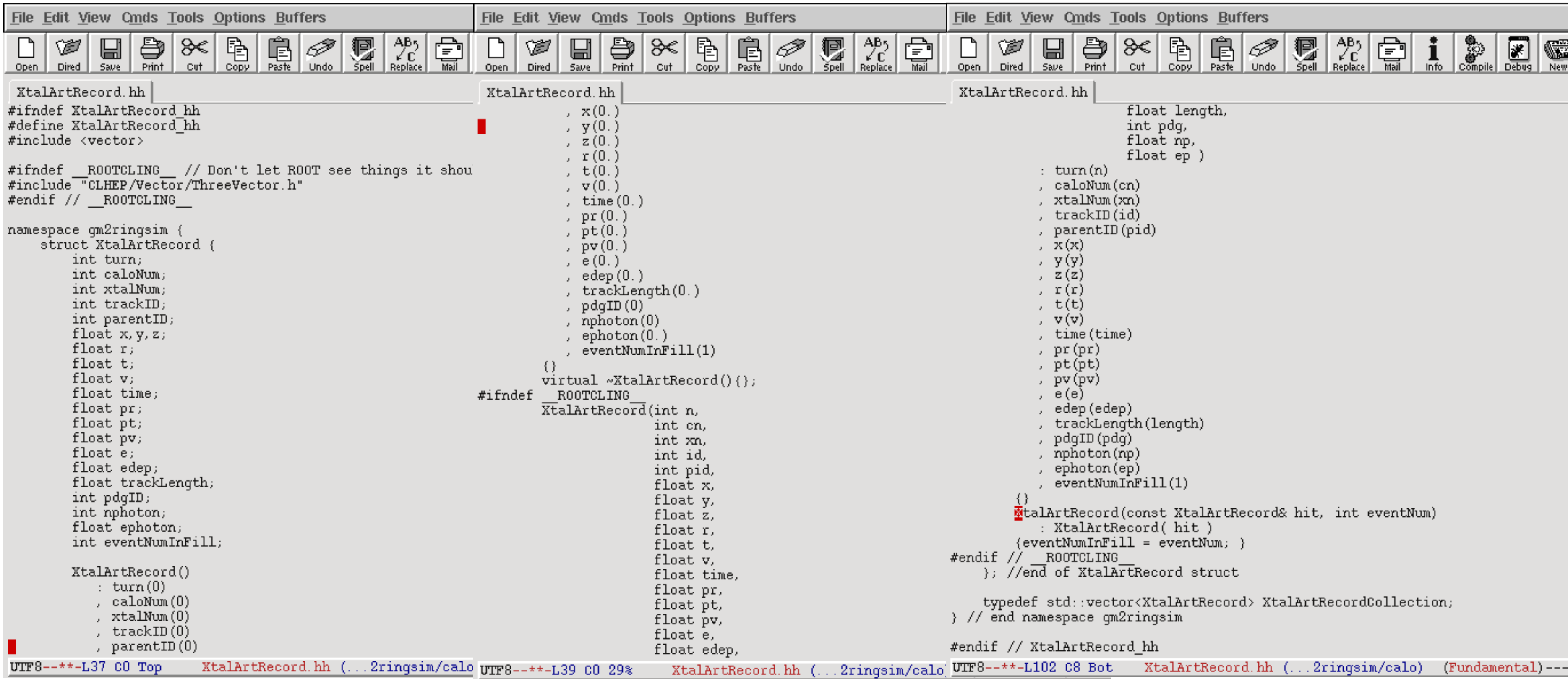


Fermilab

gm2vtk


Producer Path fragment
Hit Producer
Track Producer
LiveViz Filter
HitVtkViz Producer
TrackVtkViz Producer
RawWireCollection
HitCollection
HitCollection
TrackCollection
HitCollection
VtkVizData (Hits)
TrackCollection
VtkVizData (Tracks)
VtkVizData (getManyByType)
VtkVizData (getManyByType)
ParaView
Kitware
ARL
doWeViz?
CatalystLive Service
doWeViz?
addVizesData
vizit
CatalystLiveVtkViz Analyzer
WriteVtkVizData Analyzer
.vtm file
RawWireCollection
HitCollection
TrackCollection
art Event Store
VtkVizData (Hits)
VtkVizData (Tracks)
Legend
HitCollection
Get data product from or put data product into art Event
VtkVizData (Tracks)
Get or put transient VtkVizData product from or into art event
doWeViz?
Call method in service
Proceed to next producer in path
Proceed in path if filter passes
Network connection
Green Text
Module or service provided by artvtk

Paraview Language



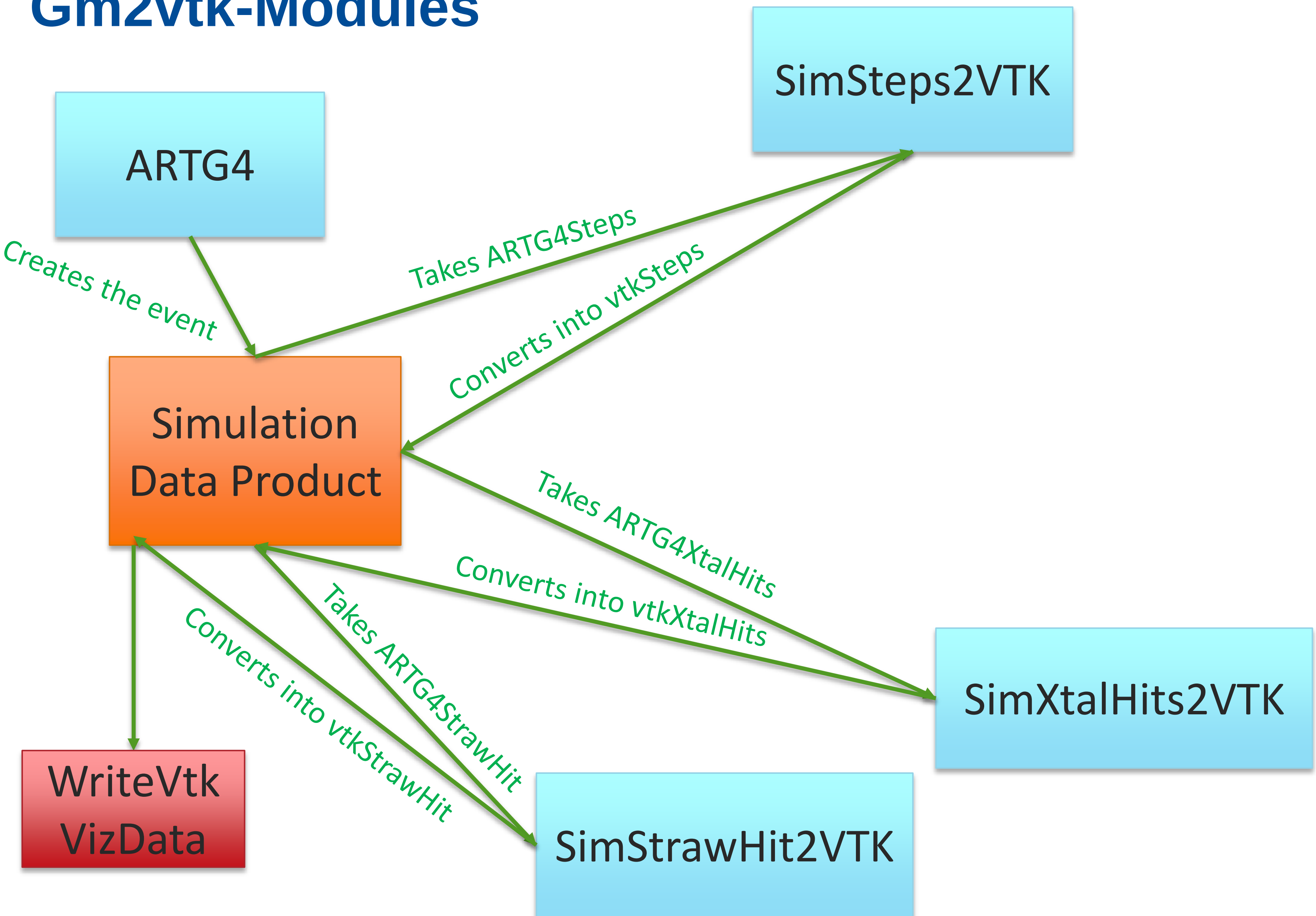
```
XtalArtRecord.hh
#ifndef XtalArtRecord_hh
#define XtalArtRecord_hh
#include <vector>

#ifndef __ROOTCLING__ // Don't let ROOT see things it shou
#include "CLHEP/Vector/ThreeVector.h"
#endif // __ROOTCLING__

namespace gm2ringsim {
    struct XtalArtRecord {
        int turn;
        int caloNum;
        int xtalNum;
        int trackID;
        int parentID;
        float x,y,z;
        float r;
        float t;
        float v;
        float time;
        float pr;
        float pt;
        float pv;
        float e;
        float edep;
        float trackLength;
        int pdgID;
        int nphoton;
        float ephoton;
        int eventNumInFill;

        XtalArtRecord()
            : turn(0)
            , caloNum(0)
            , xtalNum(0)
            , trackID(0)
            , parentID(0)
UTF8--**-L37 C0 Top    XtalArtRecord.hh (...2ringsim/calo
```

```
XtalArtRecord.hh
            , x(0.)
            , y(0.)
            , z(0.)
            , r(0.)
            , t(0.)
            , v(0.)
            , time(0.)
            , pr(0.)
            , pt(0.)
            , pv(0.)
            , e(0.)
            , edep(0.)
            , trackLength(0.)
            , pdgID(0)
            , nphoton(0)
            , ephoton(0.)
            , eventNumInFill(1)
        {}
        virtual ~XtalArtRecord(){};
#ifndef __ROOTCLING__
        XtalArtRecord(int n,
                      int cn,
                      int xn,
                      int id,
                      int pid,
                      float x,
                      float y,
                      float z,
                      float r,
                      float t,
                      float v,
                      float time,
                      float pr,
                      float pt,
                      float pv,
                      float e,
                      float edep,
UTF8--**-L39 C0 29%    XtalArtRecord.hh (...2ringsim/calo
```

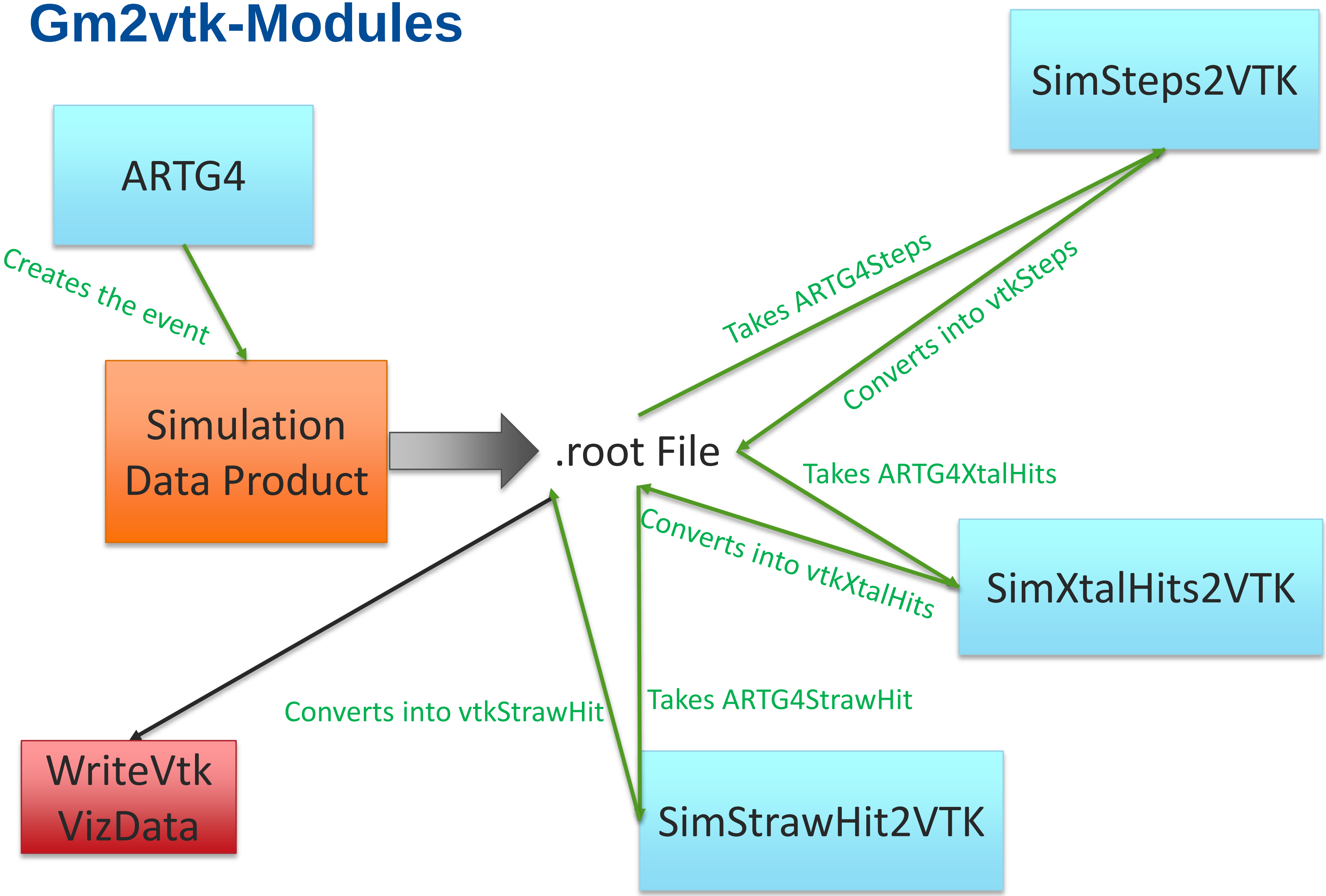
```
XtalArtRecord.hh
                      float length,
                      int pdg,
                      float np,
                      float ep )
            : turn(n)
            , caloNum(cn)
            , xtalNum(xn)
            , trackID(id)
            , parentID(pid)
            , x(x)
            , y(y)
            , z(z)
            , r(r)
            , t(t)
            , v(v)
            , time(time)
            , pr(pr)
            , pt(pt)
            , pv(pv)
            , e(e)
            , edep(edep)
            , trackLength(length)
            , pdgID(pdg)
            , nphoton(np)
            , ephoton(ep)
            , eventNumInFill(1)
        {}
        XtalArtRecord(const XtalArtRecord& hit, int eventNum)
            : XtalArtRecord( hit )
        {eventNumInFill = eventNum; }
#endif // __ROOTCLING__
    }; //end of XtalArtRecord struct

    typedef std::vector<XtalArtRecord> XtalArtRecordCollection;
} // end namespace gm2ringsim

#endif // XtalArtRecord_hh
UTF8--**-L102 C8 Bot    XtalArtRecord.hh (...2ringsim/calo)  (Fundamental)----
```

VTK MultiBlockDataSet

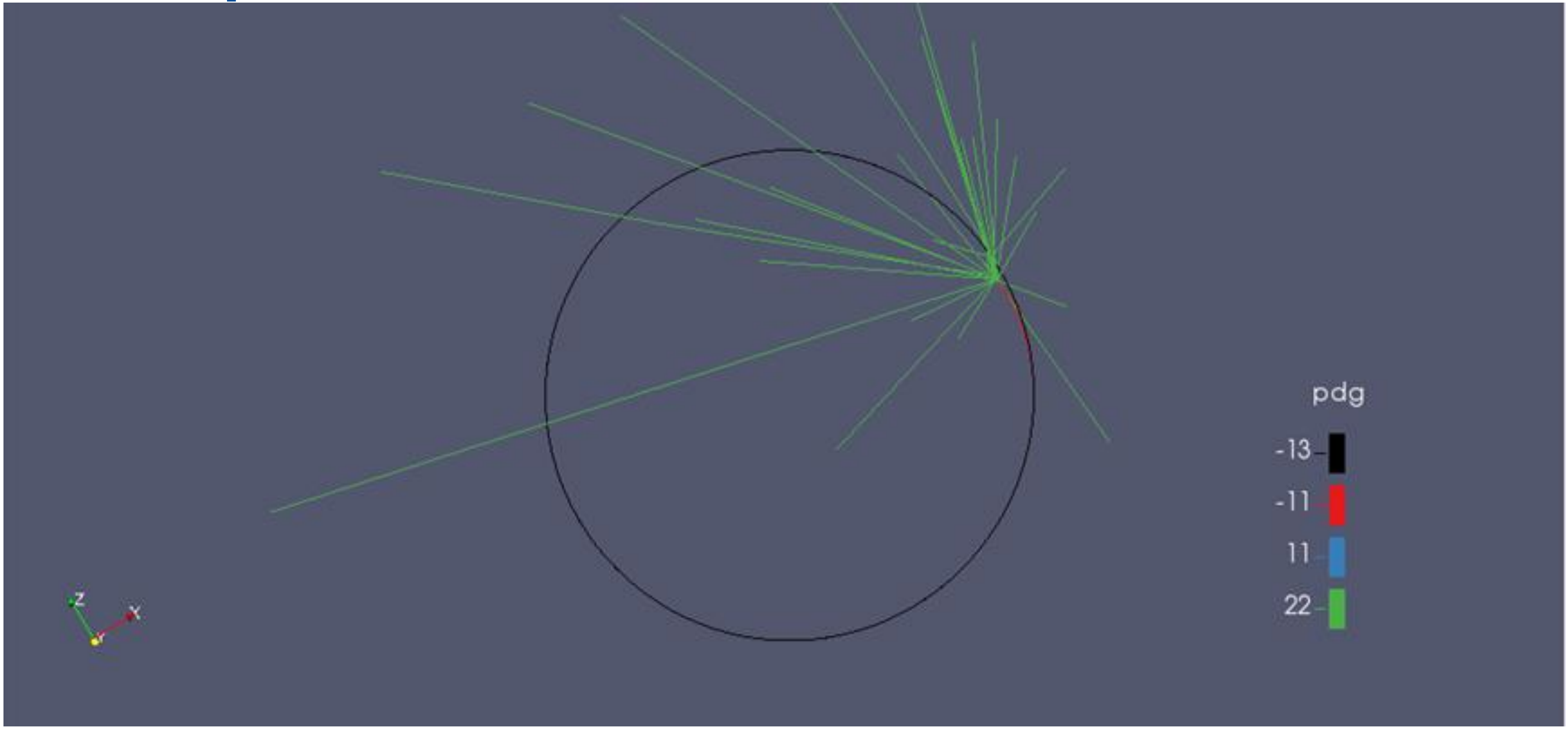
Gm2vtk-Modules



Gm2vtk-Modules



SimSteps2VTK



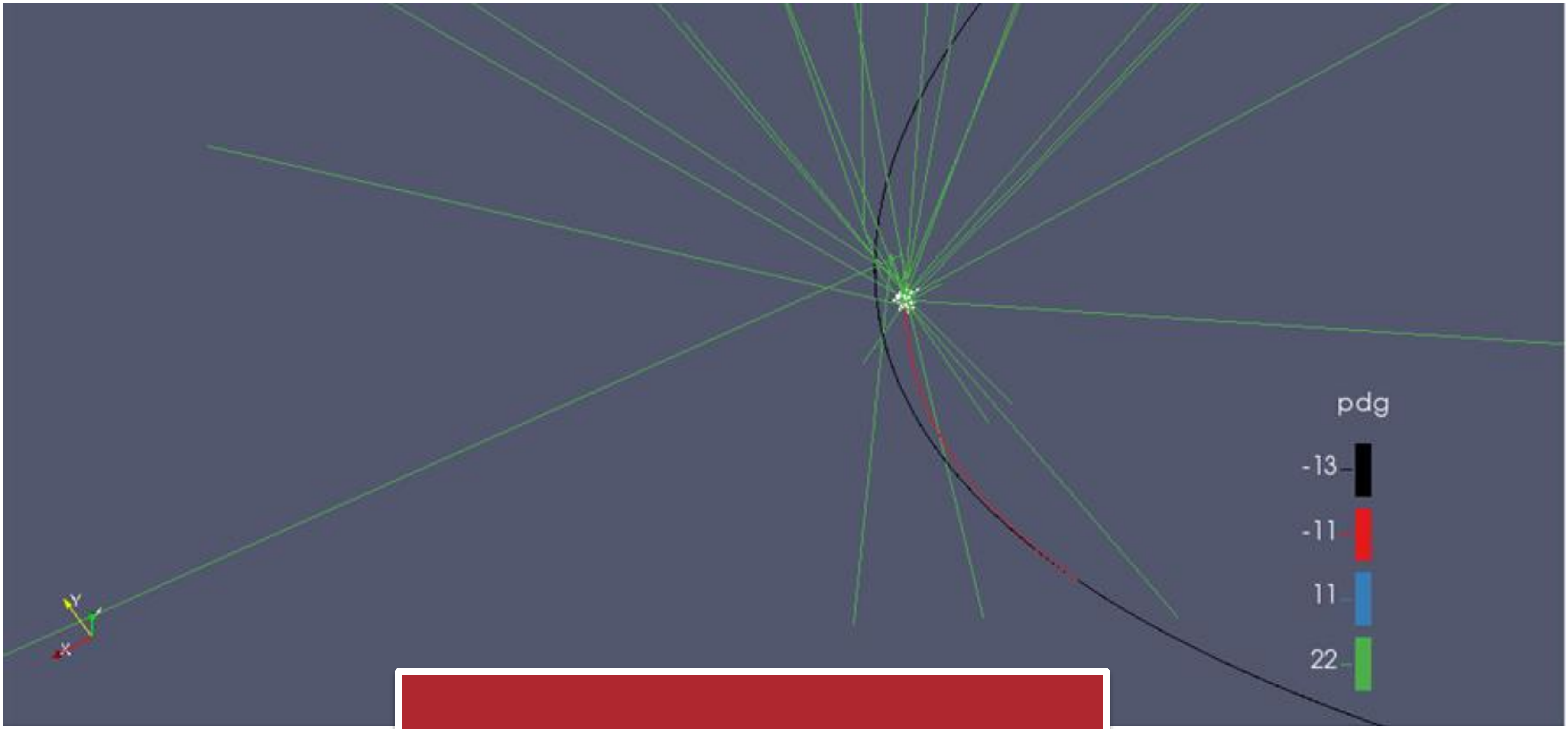
Properties of the points

- Global Step
- Global Time
- Local Time
- p
- S

Properties of the cells

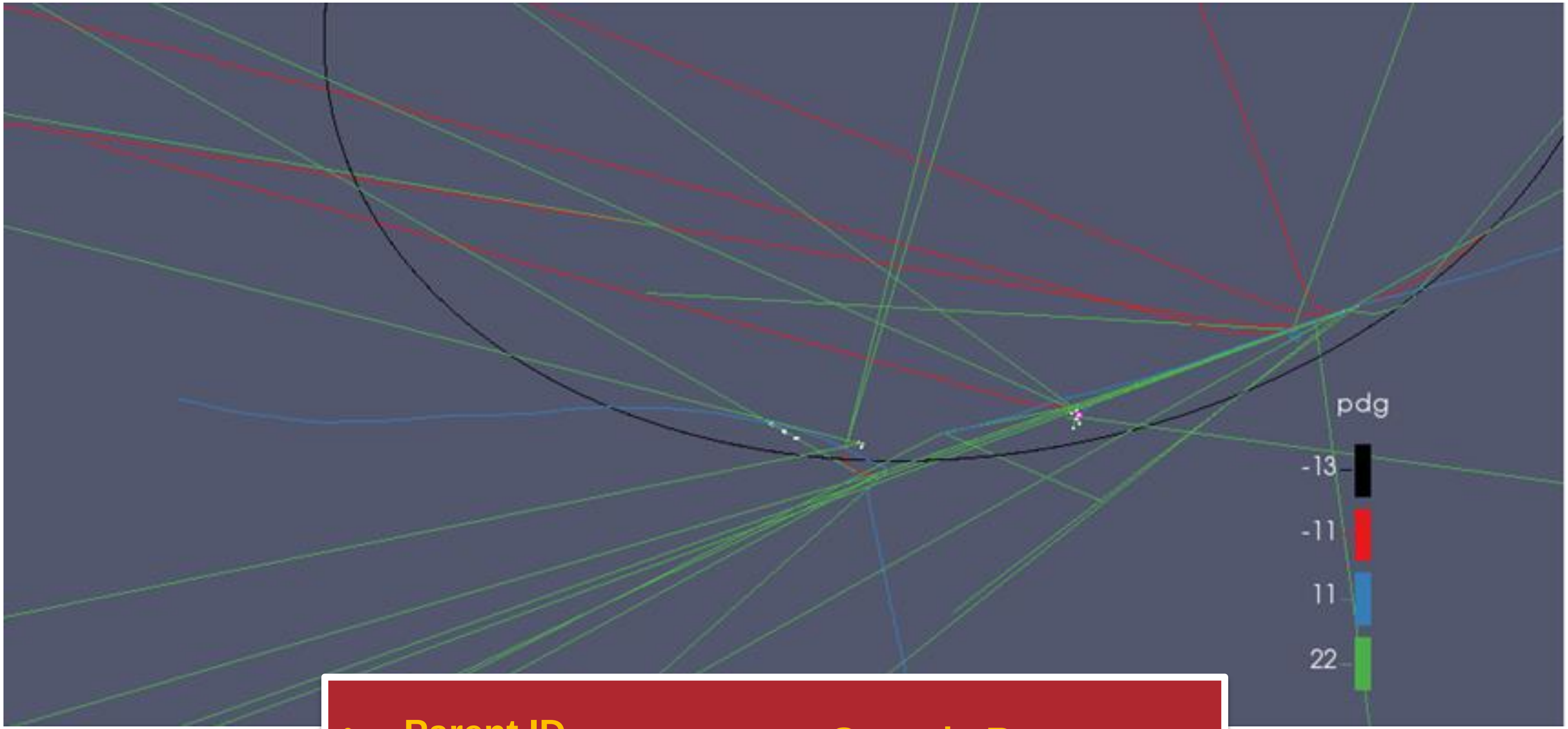
- Energy dep
- Step lenght
- Delta time
- Parent ID
- Track ID
- Pdg

SimXtalHits2VTK



- Parent ID
- Track ID
- Energy
- Hit Time
- Xtal Number
- Particle Name
- p

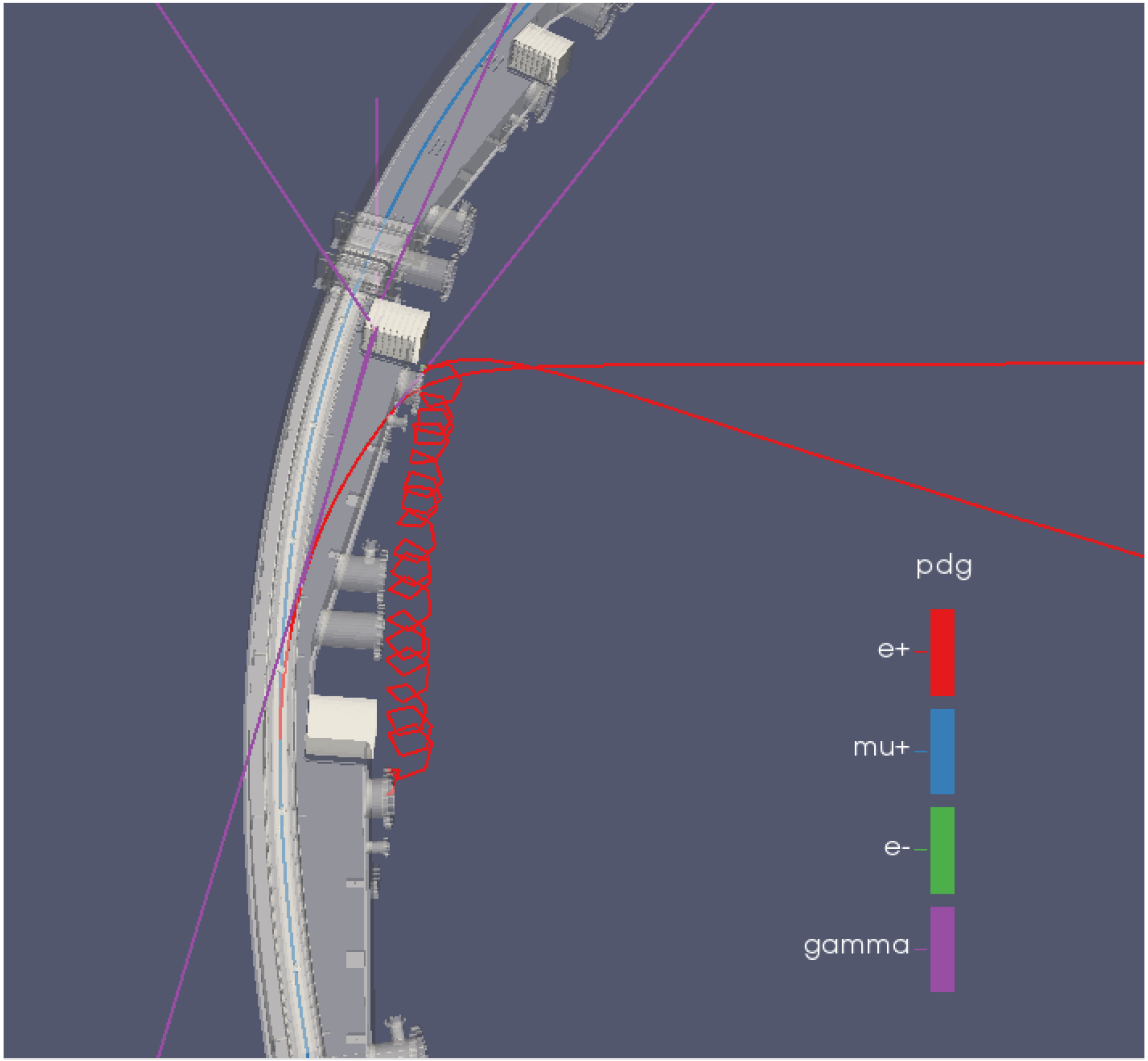
SimStrawHit2VTK



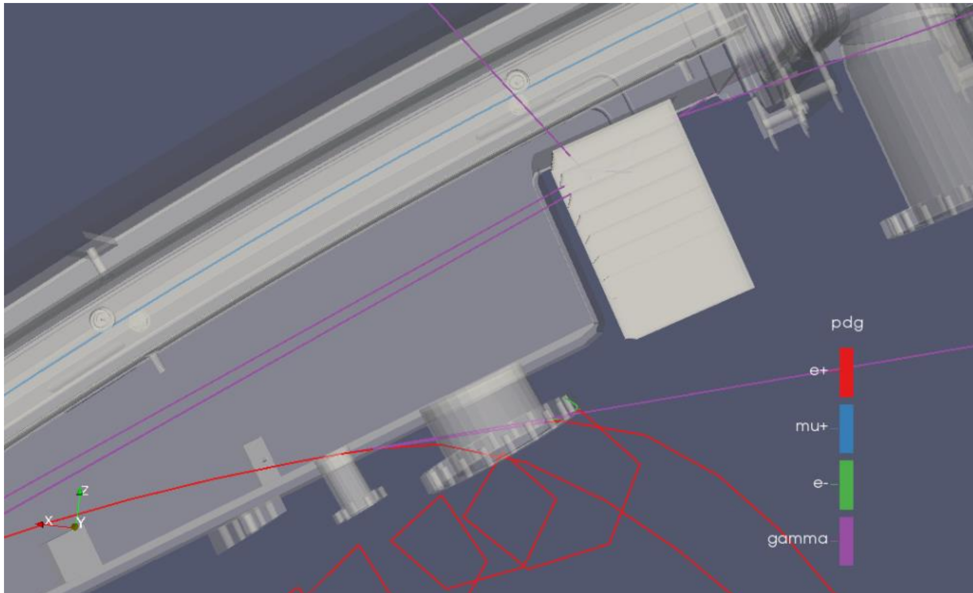
- Parent ID
- Track ID
- Hit Time
- VolumeUID
- Particle Name
- p
- Straw in Row
- Layer Number
- View Number
- Module Number
- Station Number

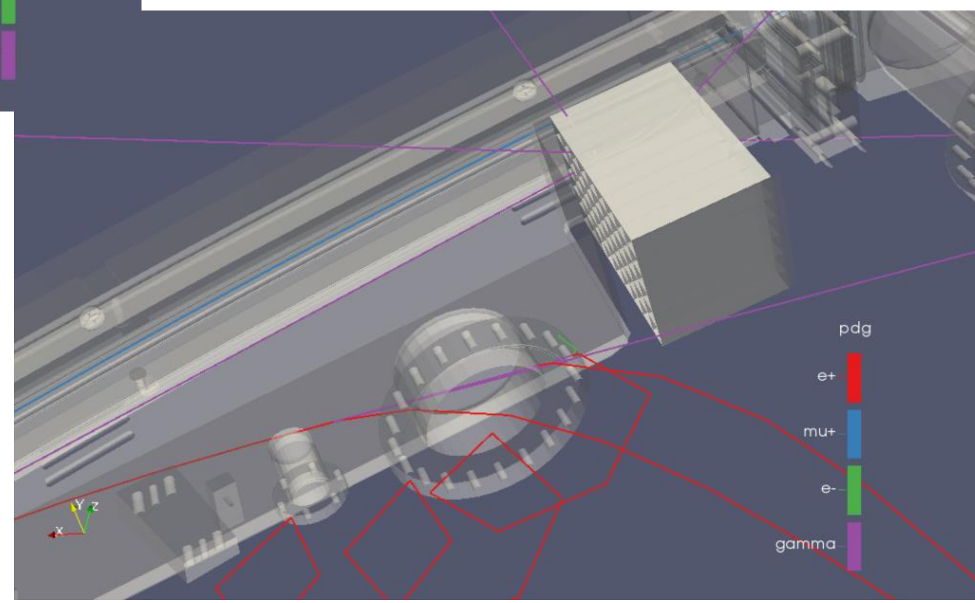
Fermilab

Event 1

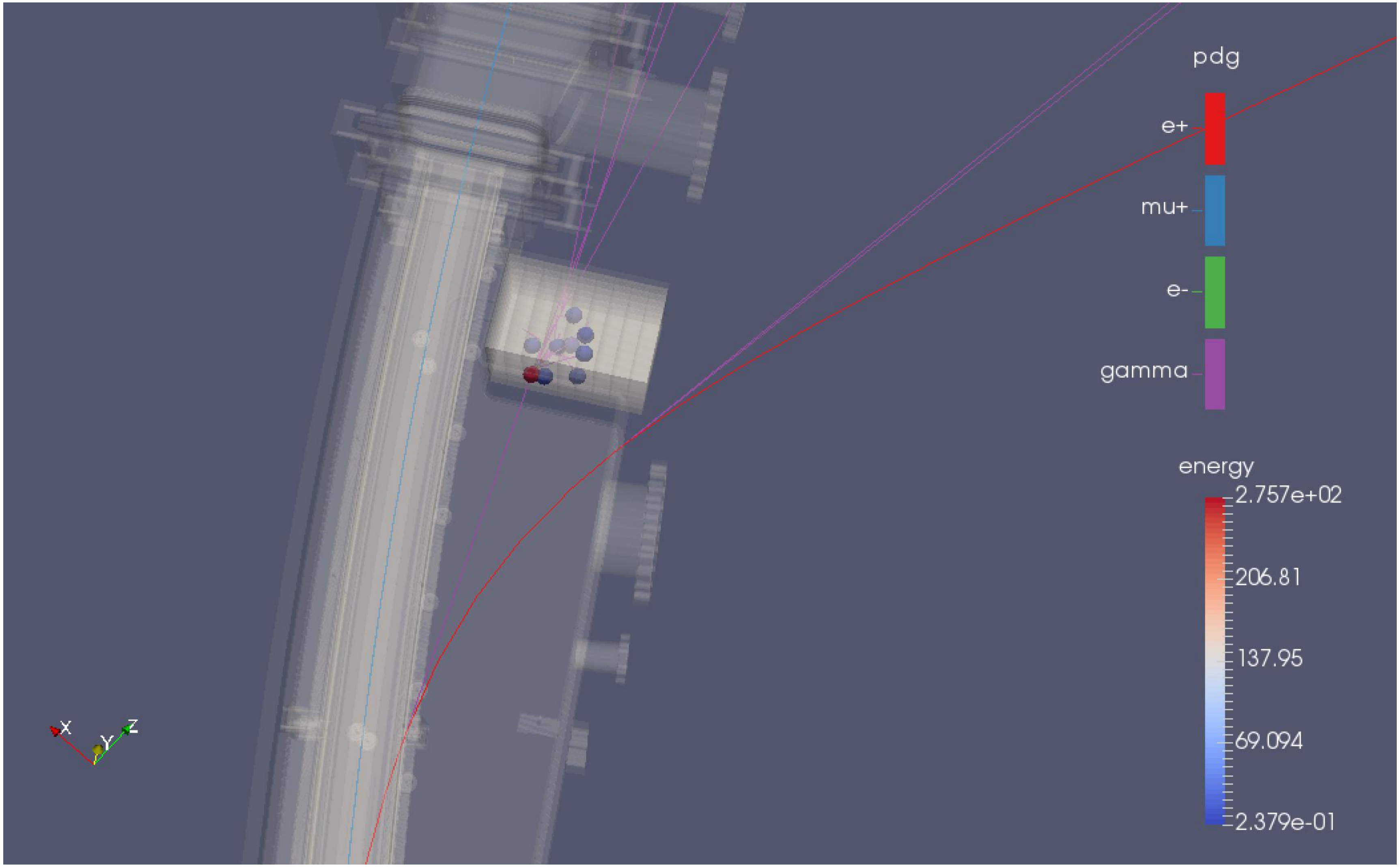


Event 1



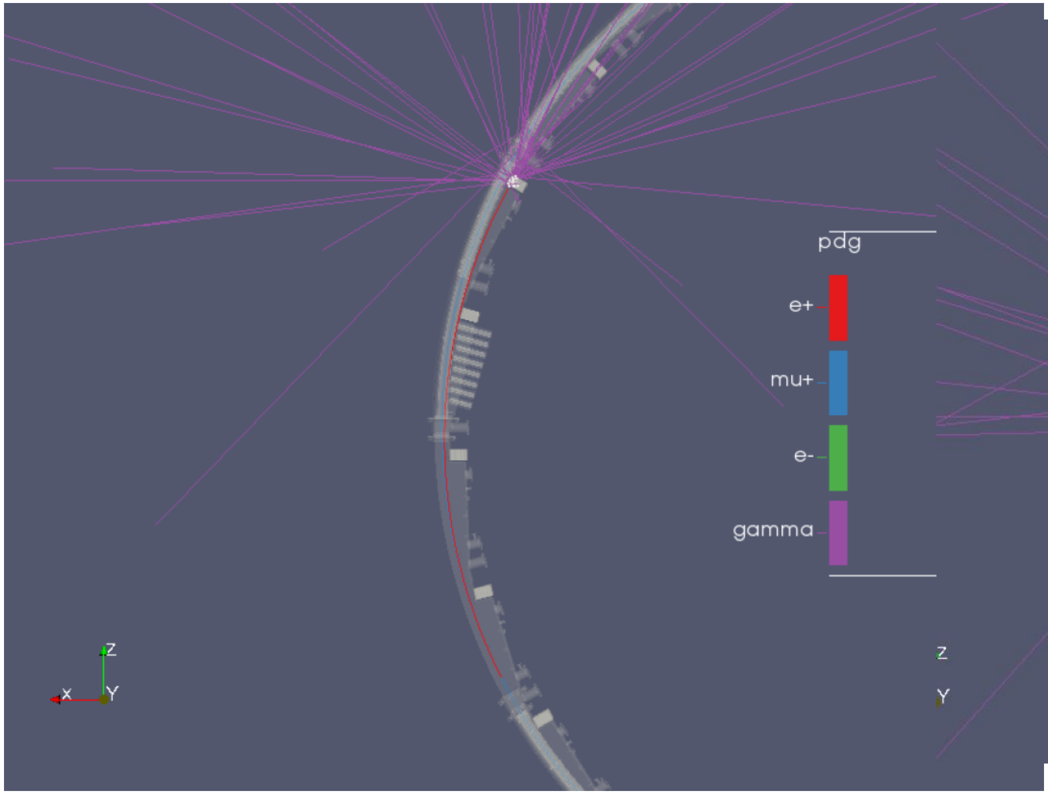


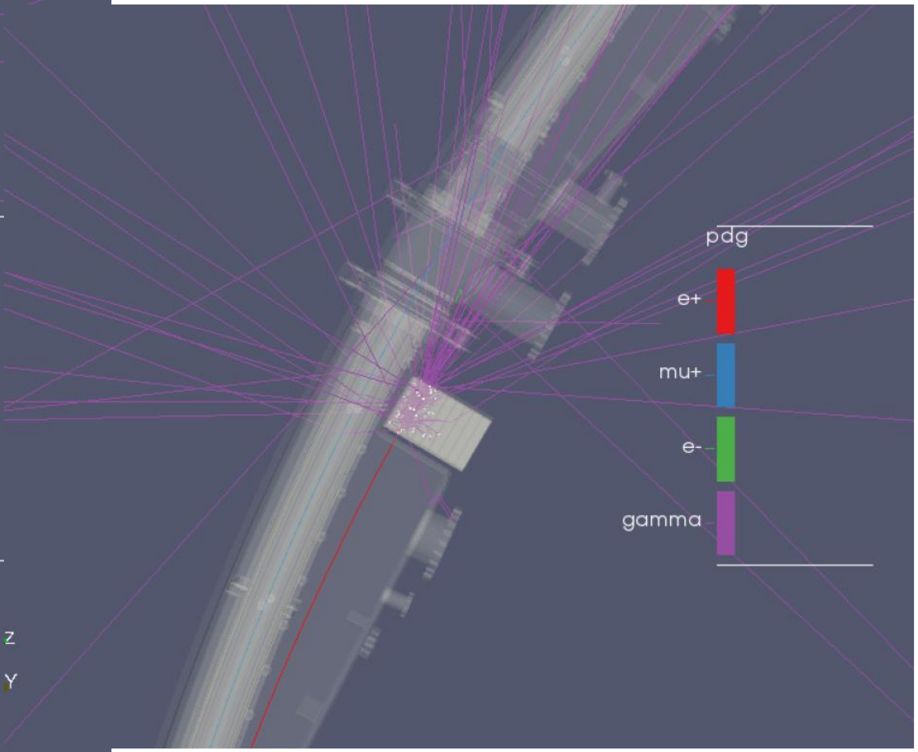
Event 2



pdg
e+
mu+
e-
gamma
energy
2.757e+02
206.81
137.95
69.094
2.379e-01
X
Y
Z

Event 2





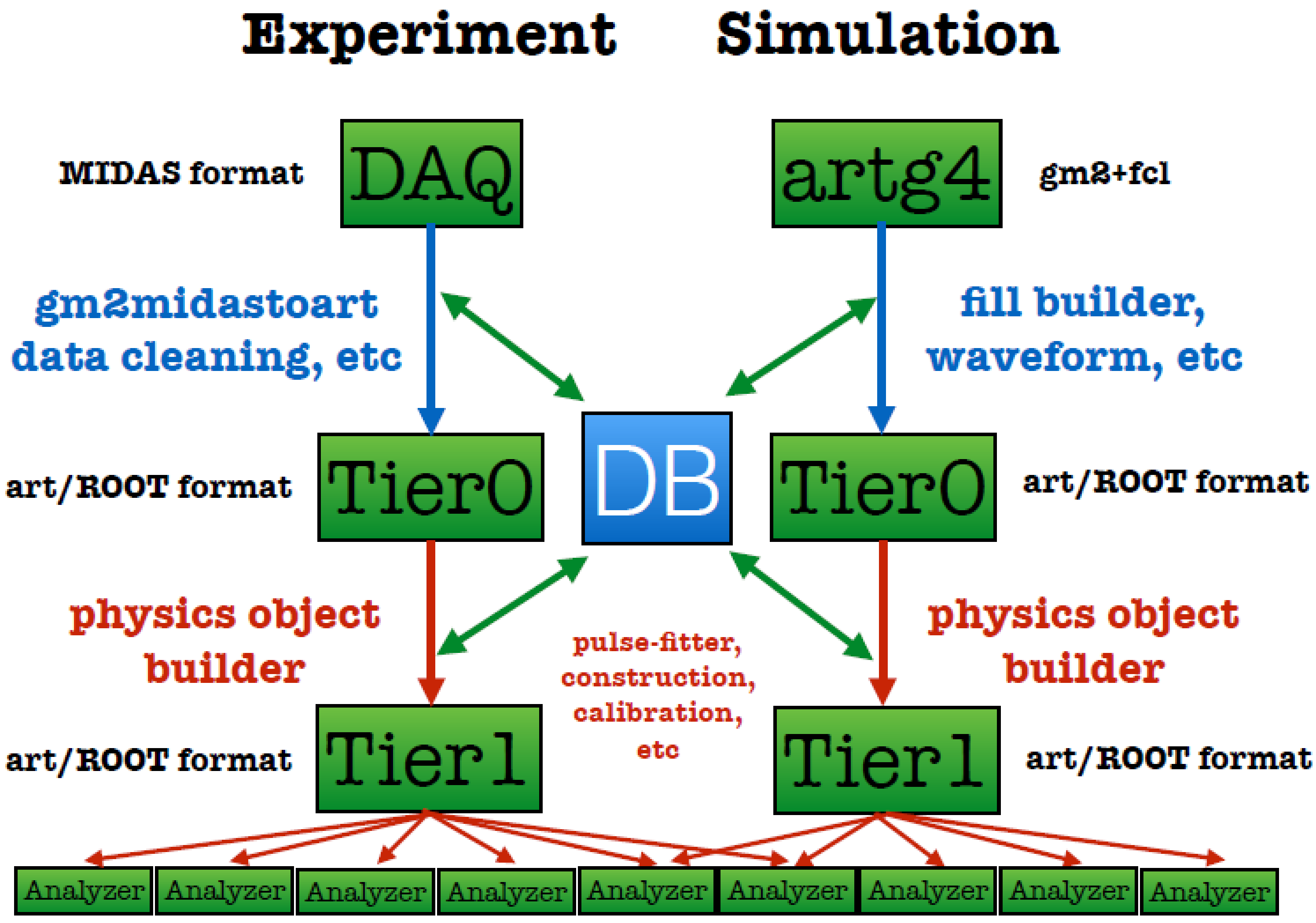
Conclusions

What I have learned:

- Muon g-2 experiment
- Playing with Art framework
- Using Paraview
- Simulation for Muon g-2 ring and detectors
- To work into a clean room

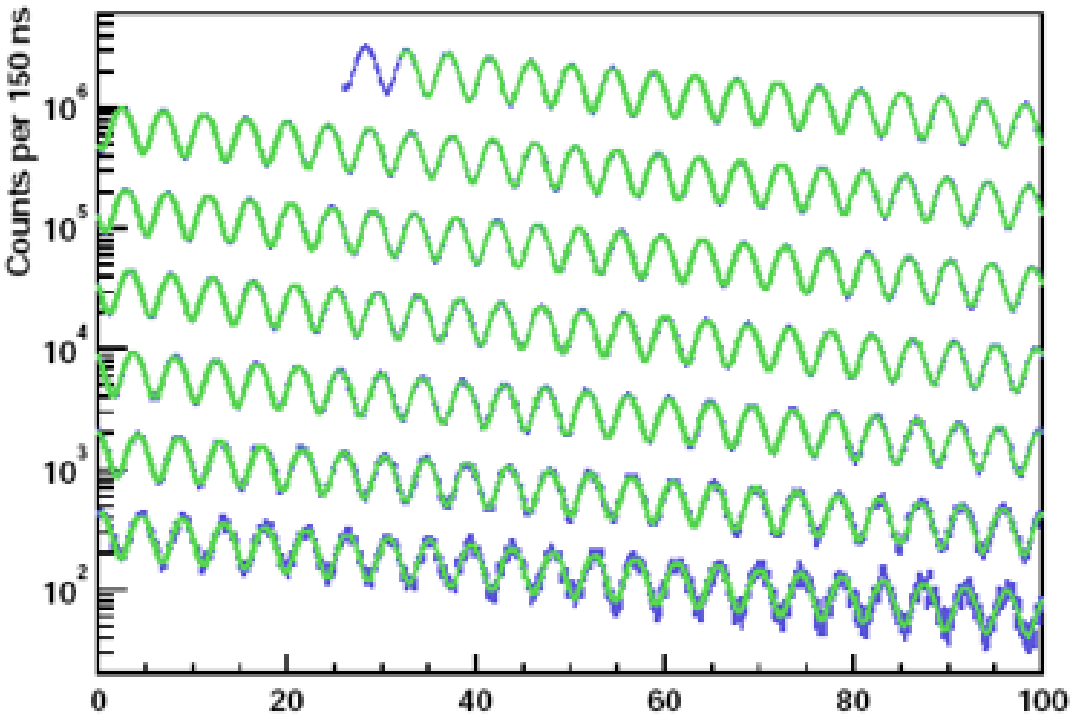
Thank you for the attention

Extra Slides

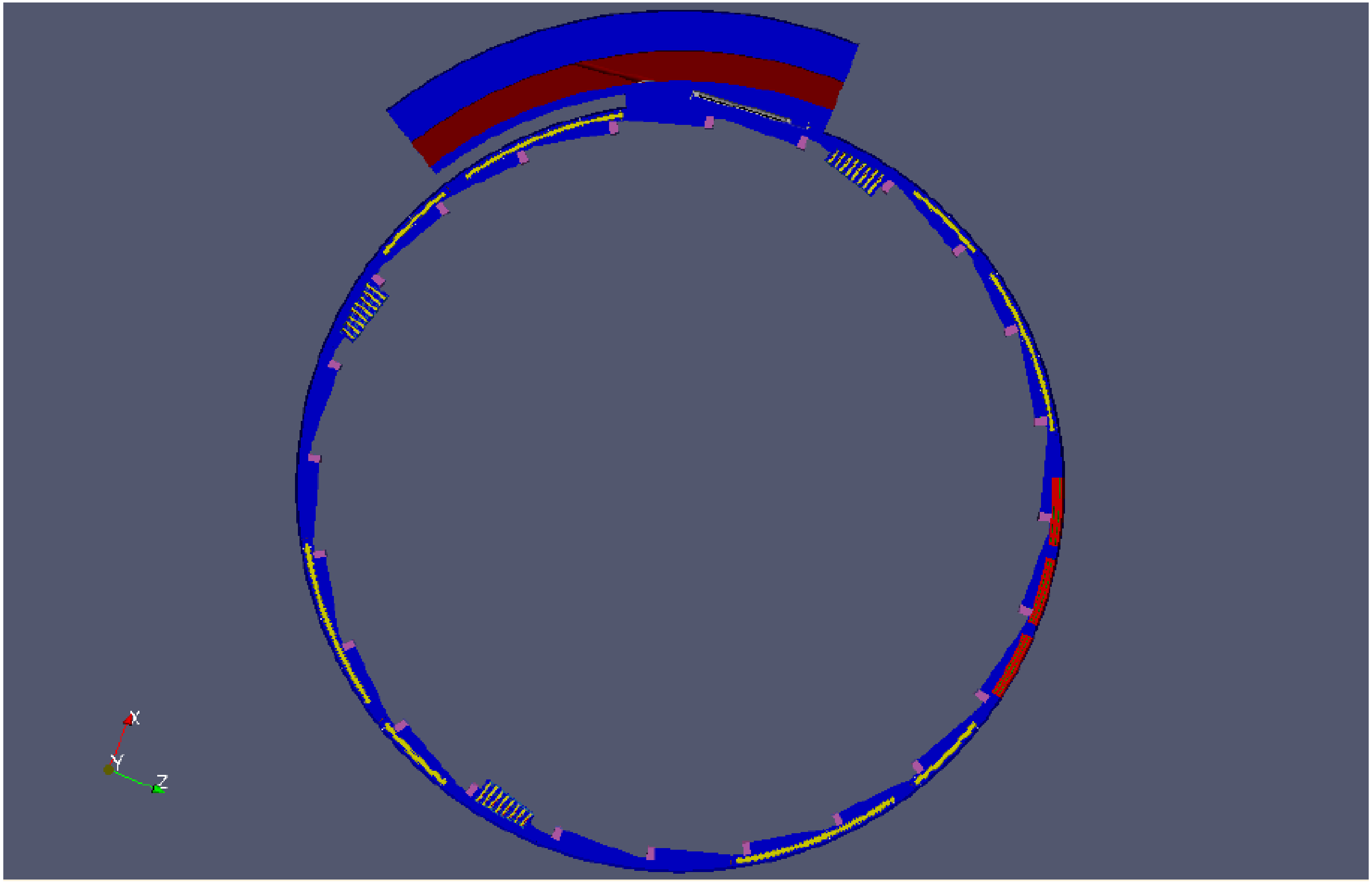


Fermilab

Extra Slides



Extra Slides



Extra Slides