Integrating Visualization Applications, such as ParaView, into HEP Software Frameworks for In-situ Event Displays

Chris Jones for Adam Lyon and Jim Kowalkowski (Fermilab)

Computing in High Energy Physics 2016

October 2016

Why are there so many 3D visualization applications?

e.g. MicroBooNE has 7 different 3D visualization applications

Use cases:

o Public relations (movies, virtual reality, inspire public)

o Analysis (event scanning)

o Algorithm development / Interactive reconstruction

o Monitoring online data

o Understanding and validating simulated events

Difficult for one application to satisfy all of these use cases

And they are fun to write (and seemingly easy to write with WebGL)

US Secretary of Energy Moniz tries the MicroBooNE VENu VR platform

[Photo from MicroBooNE]

Common elements

1. Display event data (hits, tracks) with annotations (color, text)
   - Annotations: particle ID, energy, momentum, charge, time, …

2. Display detector in background (cartoon or detailed)

3. Interact with display – rotate/zoom/pan, hide elements, select, view annotations, … Needs to be very responsive for non-frustrating experience

- Extra credit — Data processing framework reacts to display interaction and refreshes (e.g. go back 3 events; remove these hits and redetermine track)

- Most Fermilab Neutrino/Muon experiments use the *art* framework

Common Problems

- No visualization tool understands directly understands the *art* HEP data objects

- Various translation strategies:
  - Convert to Root visualization objects in an art module
  - Convert to JSON or other format in an art module (for WebGL displays) and write out
    - These files must be stored, cataloged, etc
    - Makes interaction back with the framework next to impossible (framework isn't running while display is being viewed)

- Display interaction rates can be slow for complicated scenes
  - Poor, frustrating user experience
  - Developer must figure out how to mitigate (e.g. somehow show subset of data for speed)

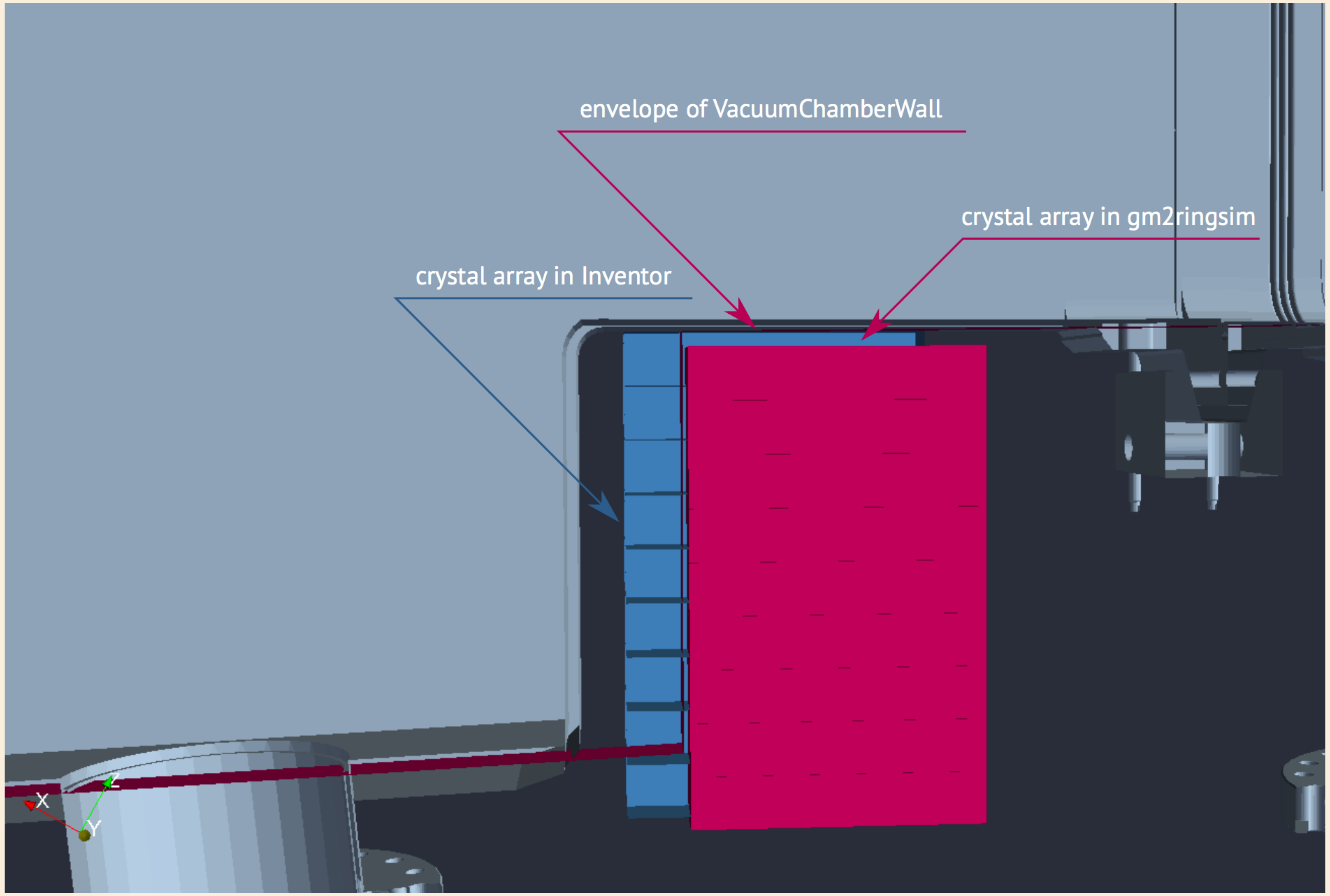
- May want to overlay visualizations (multiple events; CAD 3D models)

Overlay example (Muon g-2)

Overlay of Geant geometry (red) and 3D CAD model (blue)

Visualization makes misplacement of calorimeter (crystal array) in simulation obvious

Very difficult to discover otherwise


envelope of VacuumChamberWall
crystal array in gm2ringsim
crystal array in Inventor
X
Y
Z

[Jarek Kaspar]

Providing a visualization framework

Concentrate on use cases regarding analysis, development and validation

There is no standard HEP visualization solution – so we must make choices regarding technology and tools

We desire to focus on physics, leaving the graphics and interactions up to the visualization application (hope it's fast)

The solution should enable communication between the visualization and data processing frameworks

ParaView is a possible solution

- A scientific visualization application with rich capabilities
- DOE supported with a strong HPC community (e.g. Argonne, NERSC)
- Based on established VTK library
- ParaView uses advanced visualization techniques to maintain a good user experience (e.g. decimating image during manipulation)
- Very active developer/user base (e.g. v5.1 released with new Intel ray-tracer) and very responsive support from Kitware
- ParaView Client Scriptable in python with NumPy capabilities
- Catalyst client-server system
  - data processing framework is the server pushing VTK objects to online ParaView client

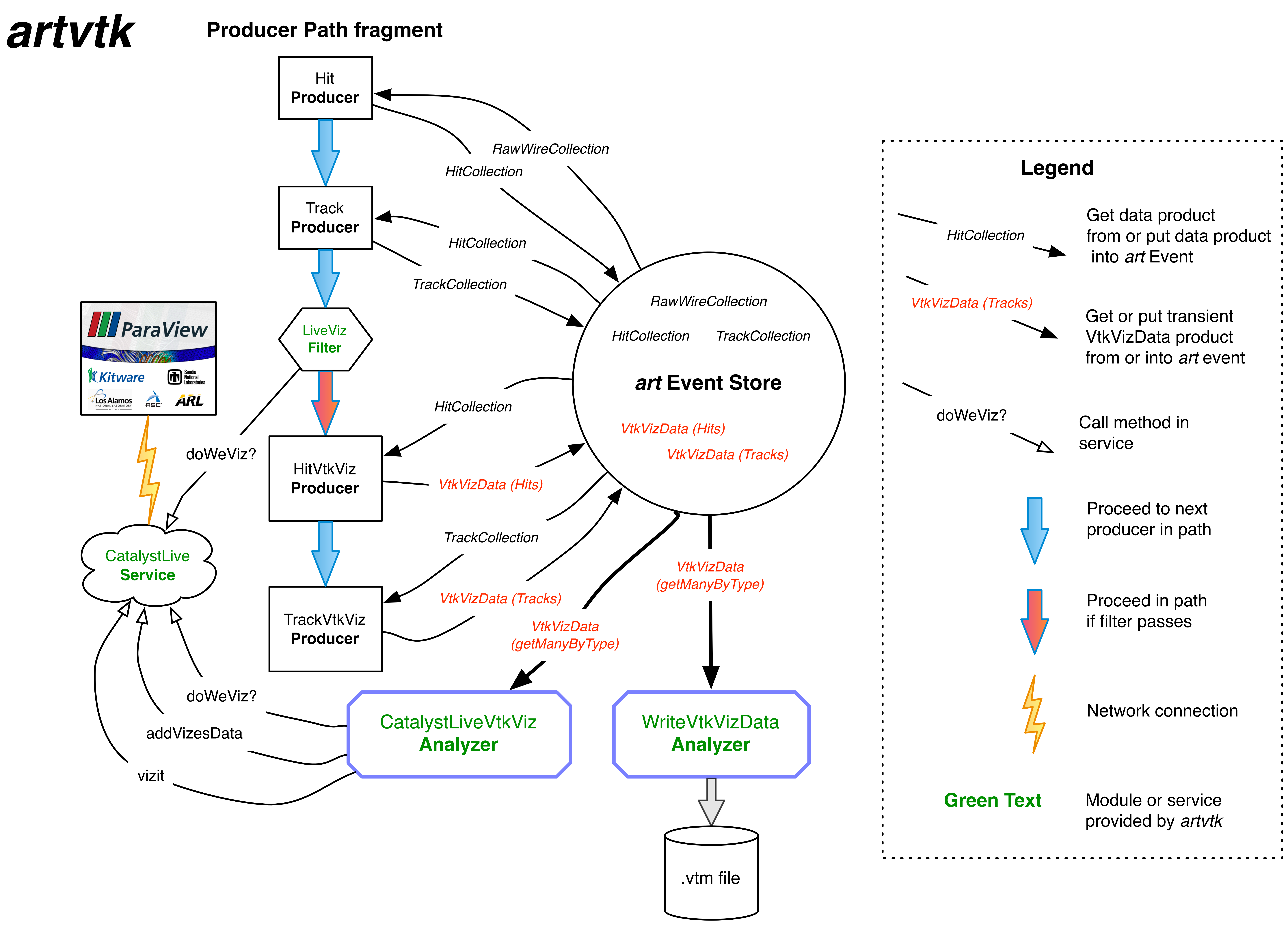
ParaView
Kitware
Sandia National Laboratories
Los Alamos NATIONAL LABORATORY EST. 1943
ASC
ARL

www.kitware.com Version: 5.1.0 64-bit www.paraview.org

Interface Design

- **Infrastructure for translating physics data products to VTK objects for ParaView - artvtk**
- **Modules dynamically put VTK objects as data products into event**
- **Modules pull VTK objects out of event and write out or pass directly to ParaView via Catalyst**

  (translation will be skipped if no ParaView client is connected)
- **e.g. gm2vtk, larvtk for experiment specific data translation producers**



artvtk

Producer Path fragment

Hit **Producer**

Track **Producer**

LiveViz **Filter**

HitVtkViz **Producer**

TrackVtkViz **Producer**

CatalystLive **Service**

CatalystLiveVtkViz **Analyzer**

WriteVtkVizData **Analyzer**

.vtm file

***art* Event Store**

RawWireCollection

HitCollection

TrackCollection

VtkVizData (Hits)

VtkVizData (Tracks)

VtkVizData (getManyByType)

doWeViz?

addVizesData

vizit

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

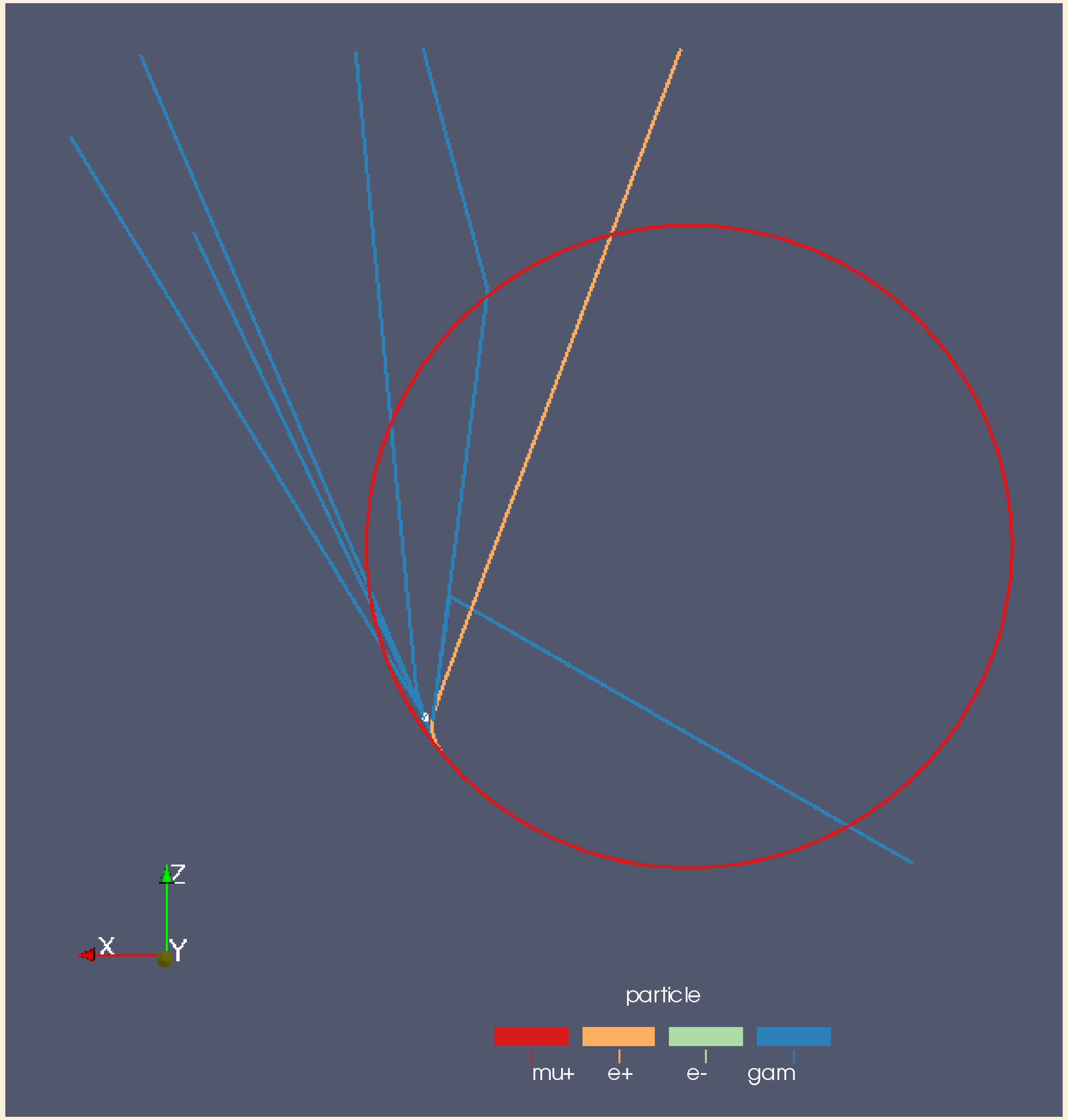
Proceed to next producer in path

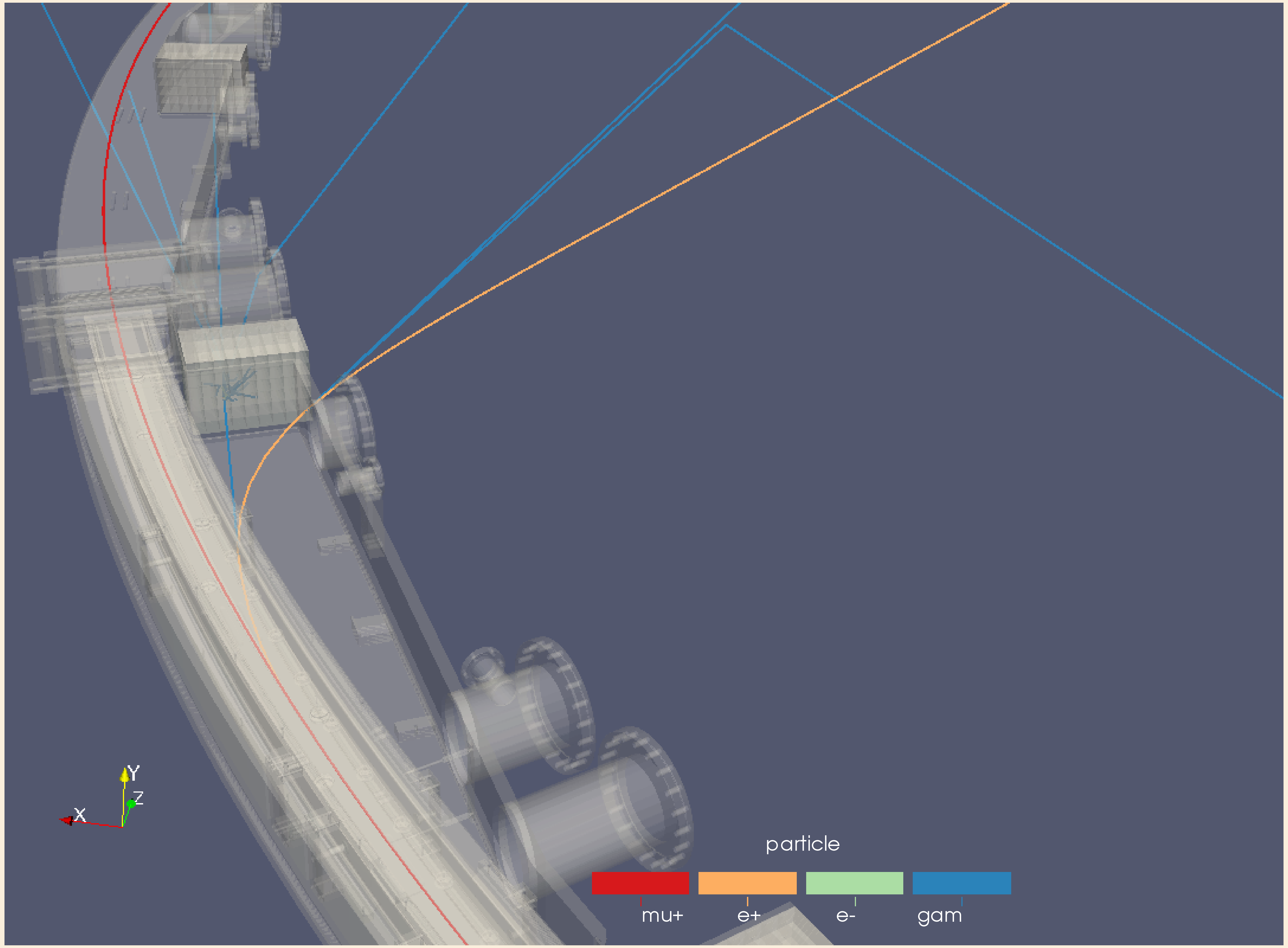
Proceed in path if filter passes

Network connection

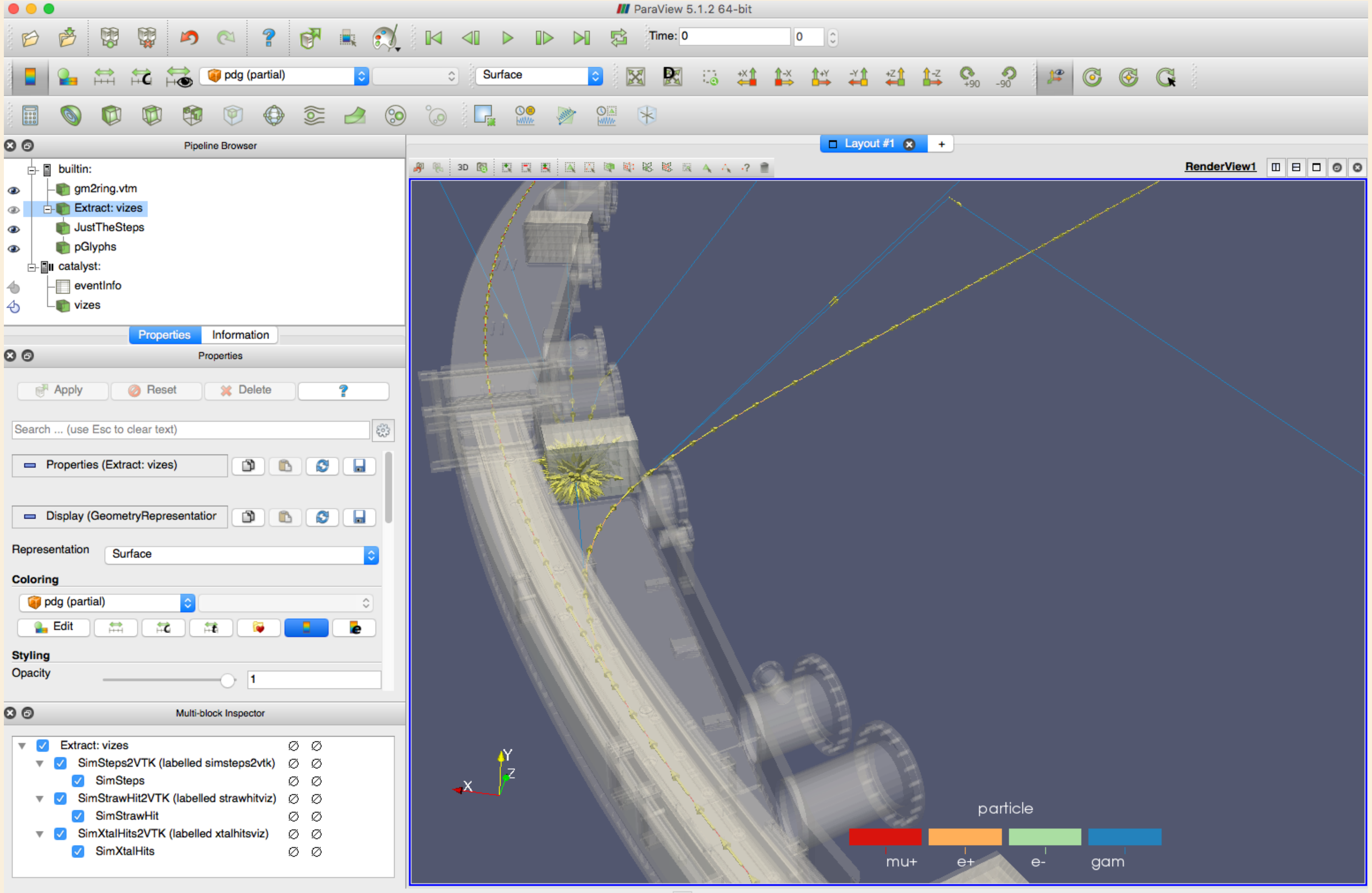
Green Text — Module or service provided by *artvtk*

Example, Muon g-2 Geant4 Steps





Example ParaView session connected to Catalyst



ParaView 5.1.2 64-bit

Time: 0

Pipeline Browser

```
builtin:
  gm2ring.vtm
  Extract: vizes
    JustTheSteps
    pGlyphs
catalyst:
  eventInfo
  vizes
```

Properties

Properties (Extract: vizes)

Display (GeometryRepresentatior

Representation: Surface

Coloring

pdg (partial)

Styling

Opacity: 1

Multi-block Inspector

```
☑ Extract: vizes
  ☑ SimSteps2VTK (labelled simsteps2vtk)
    ☑ SimSteps
  ☑ SimStrawHit2VTK (labelled strawhitviz)
    ☑ SimStrawHit
  ☑ SimXtalHits2VTK (labelled xtalhitsviz)
    ☑ SimXtalHits
```

Fermilab

Summary

- 3D visualizations are an essential part of the HEP toolbox
- We are concentrating on the ParaView application and interfacing to the *art* framework
- artvtk provides infrastructure for translating data objects to VTK and pushing to ParaView in real time
- … also direct Geant4 VTK driver is in the works (for the detector image)