

Why VTK?

- Best in class scientific rendering
- KitWare world leading opensource scientific computing company (cf. Cmake)
- Trusted in medical applications, high end engineering, non-HEP scientific visualisation
- Large user community
- Stable and **documented** API in C++
- 1000s of C++ examples to base visualization code on
- Started in December 1993 (pre-dates Geant4)
- **Works exceptionally well for Geant4!**

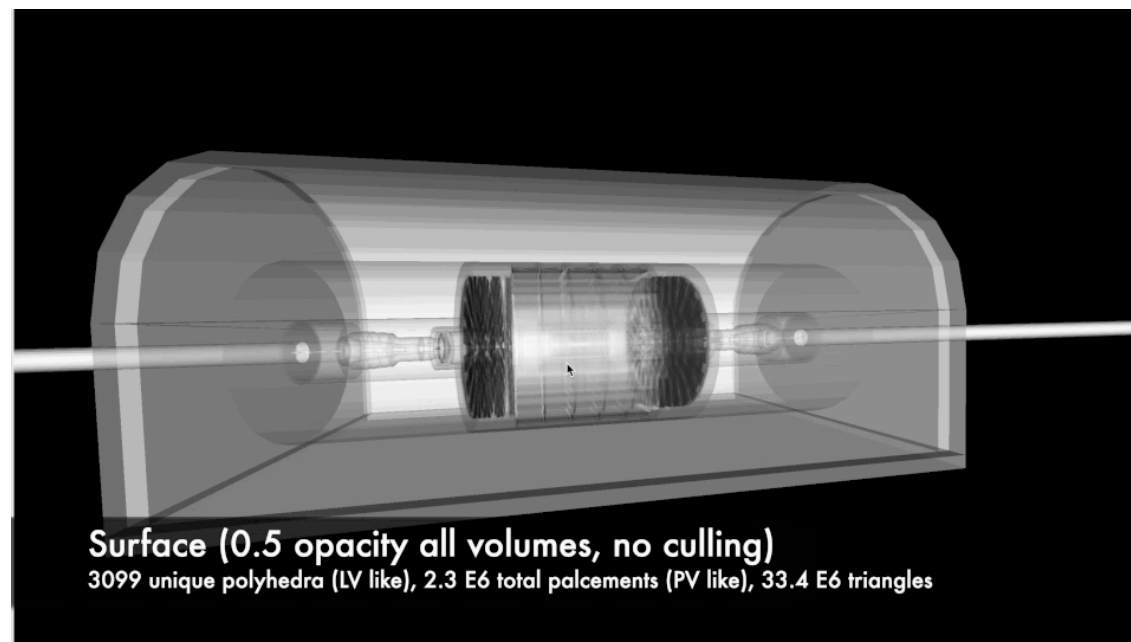
Development supported by



Vtk raw performance

- **Outstanding performance relatively easy to achieve**

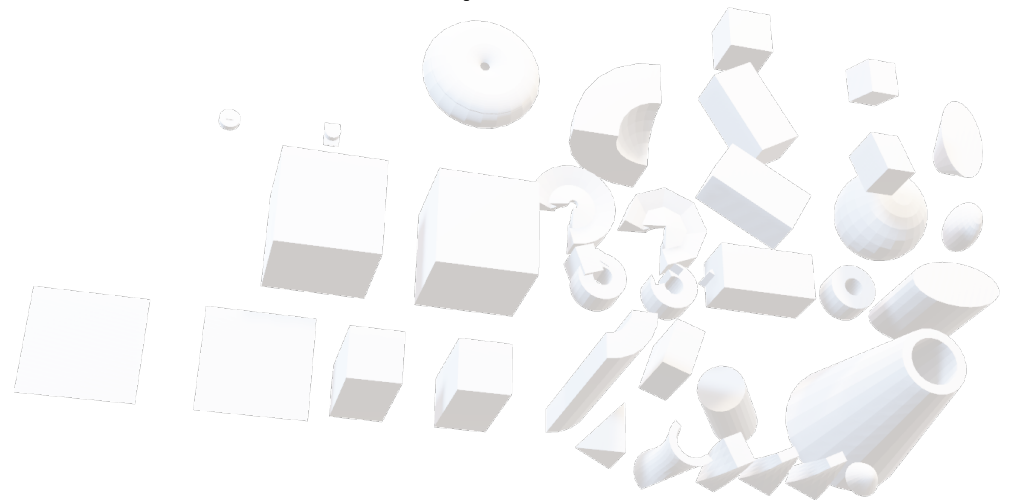
- CMS GDML from JA
- Completely pipelined, shader based rendering engine
- Conformant to modern OpenGL
- Allows visualization algorithms to run down stream (see cutting and clipping)



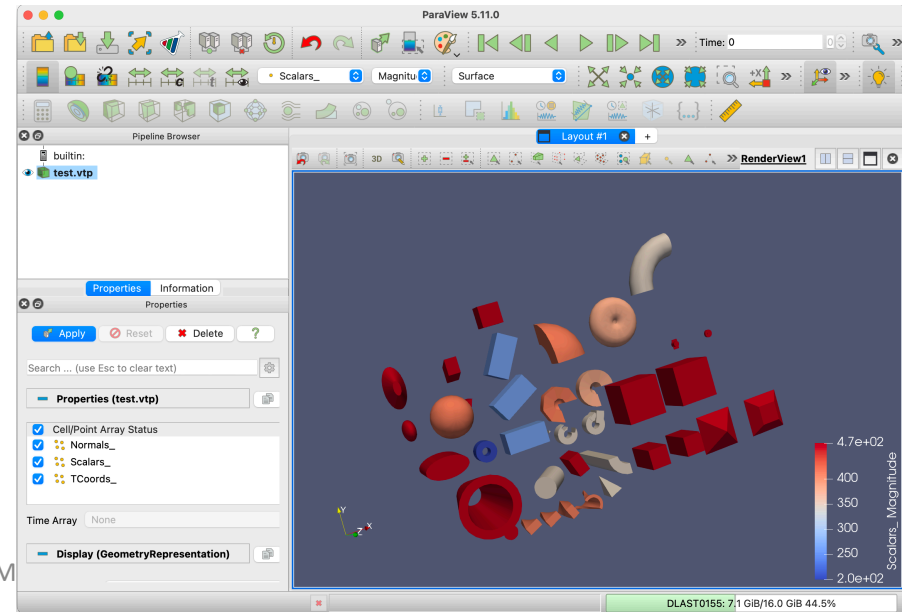
Vtk exports

- Export to almost all possible 3D file formats
 - VTU/VTP (vtk.js web model)
 - OBJ (convert to GLTF/USD web mode)
 - VRML
 - PLY
- Export screen grabs
 - PS
 - JPEG
 - TIFF
 - PNG
 - BMP
- Offscreen rendering and 3D conversion etc

Vtk → OBJ/GLTF → **Powerpoint**

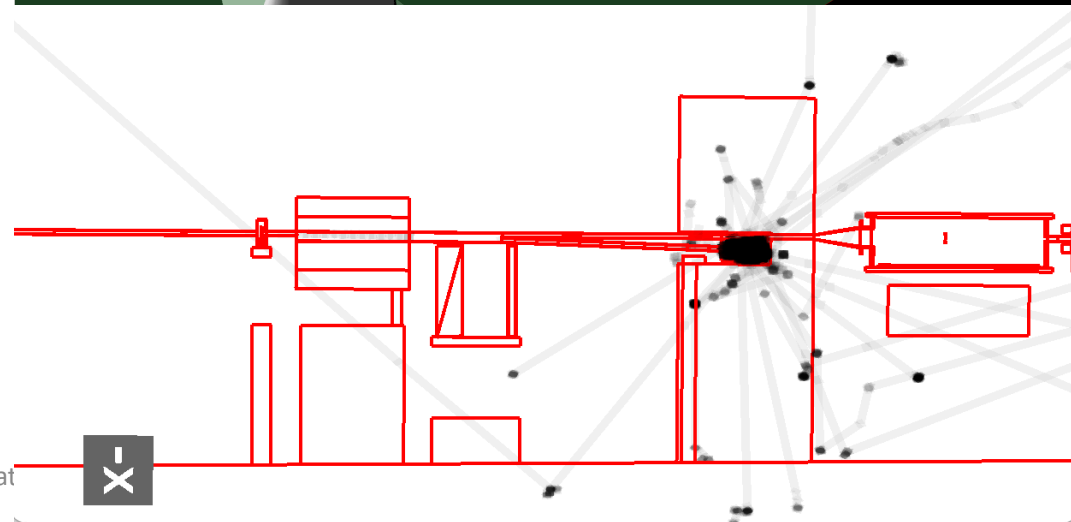
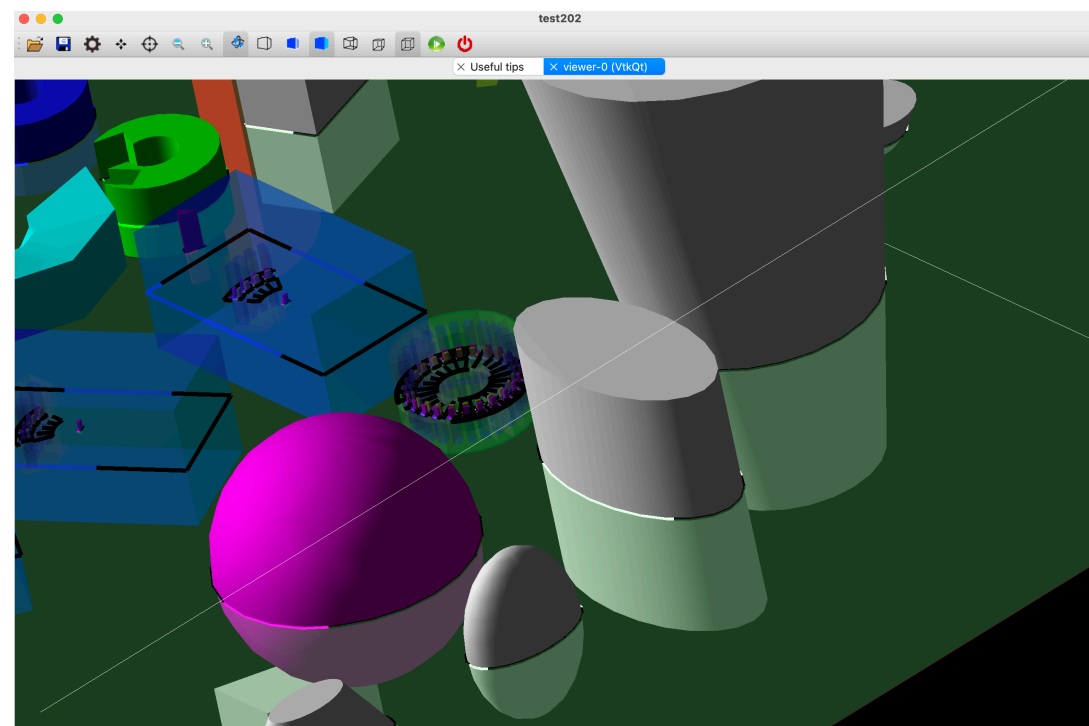


Vtk → VTP/VTP → **Paraview**



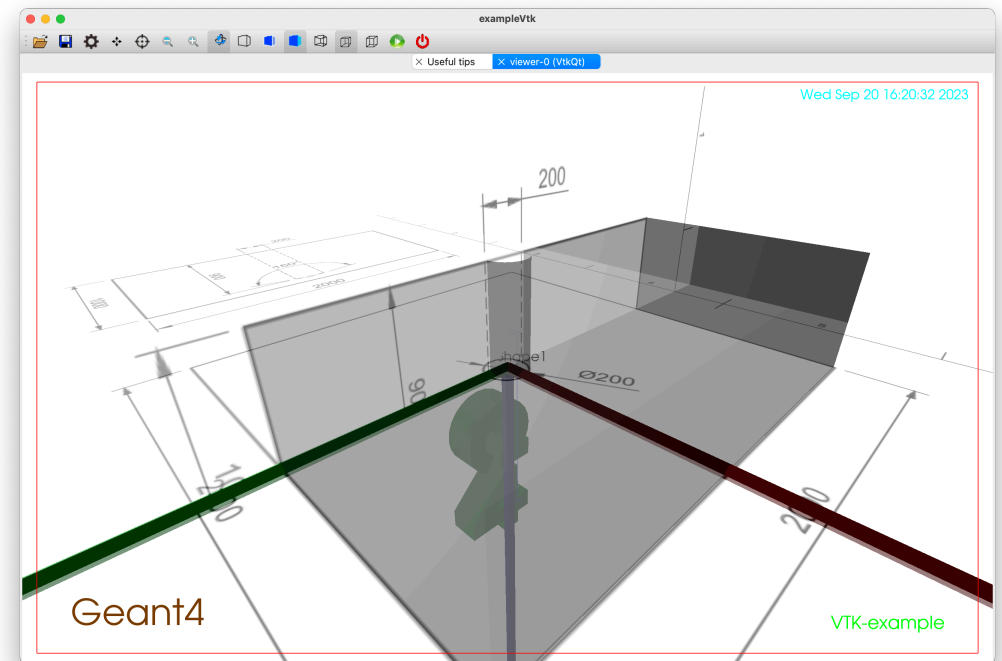
Vtk interactive cutters and clippers

- Clipper
 - Remove geometry and close with appropriate polygon
- Cutter
 - Make a slice through geometry
 - Export data to file
 - Outstanding for subsequent analysis and scoring plotting
- Cut and clip geometry and transient objects (tracks, scoring, hits etc)
- Both controlled inside the viewer with a widget
- **Realtime performance**



Vtk “side-loaders”

- Load data to augment modelling process
 - 2D image or 3D data
 - DICOM
 - Multiphysics data
- Example is CAD 2D images
 - exported as PDF->PNG
 - Side loaded in to VTK
- **Need to put into generic vis not just VTK**



Vtk possibilities

Some examples if VTK is available within G4

- Surface mesh to tetrahedron mesh
 - No need for external tetgen in medical examples (e.g. ICRP145)
- Marching cubes or tetrahedra
 - Surface extraction from DICOM data
- Mesh subdivision and refinement

Vtk possibilities (HEP outreach)

- Complex lighting
- Shadows
- Physically based rendering
- Ray tracing Ospray
- OpenXR simple to enable (VR/AR for free)
- Vtk for iOS and Android and Javascript so mobile version possible.
- **Each of these (non-critical improvements) each will take ~40 hours work to implement**
- Boogert happy to support keen post-doc or PhD student wanted to develop an HEP outreach project