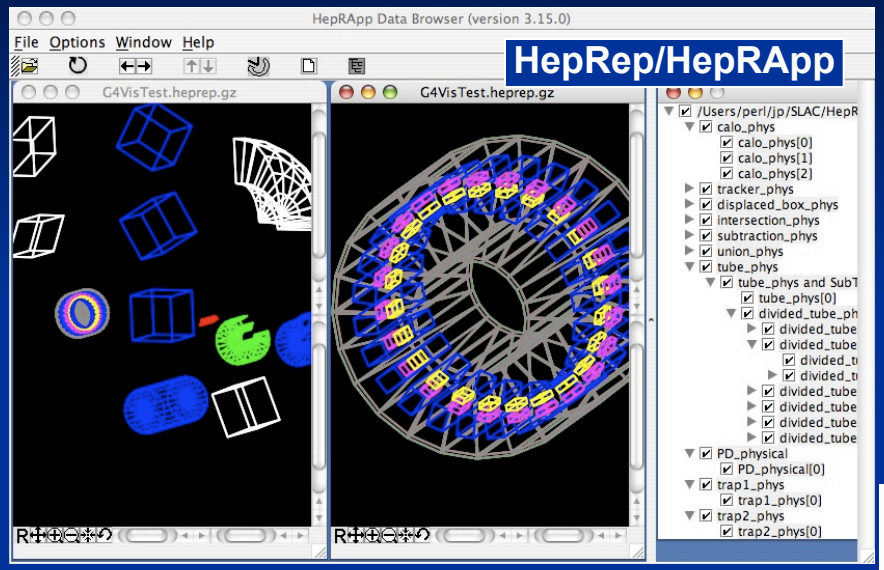
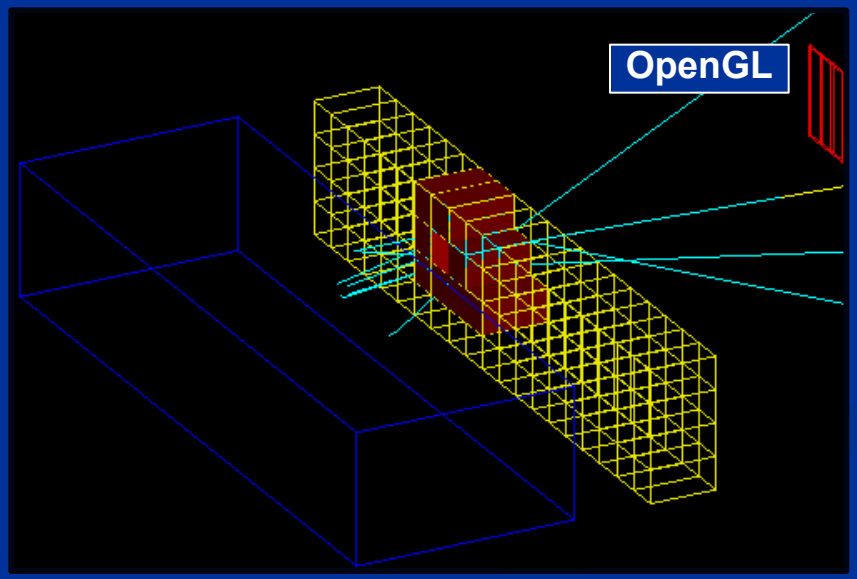
Geant4 Visualization Drivers



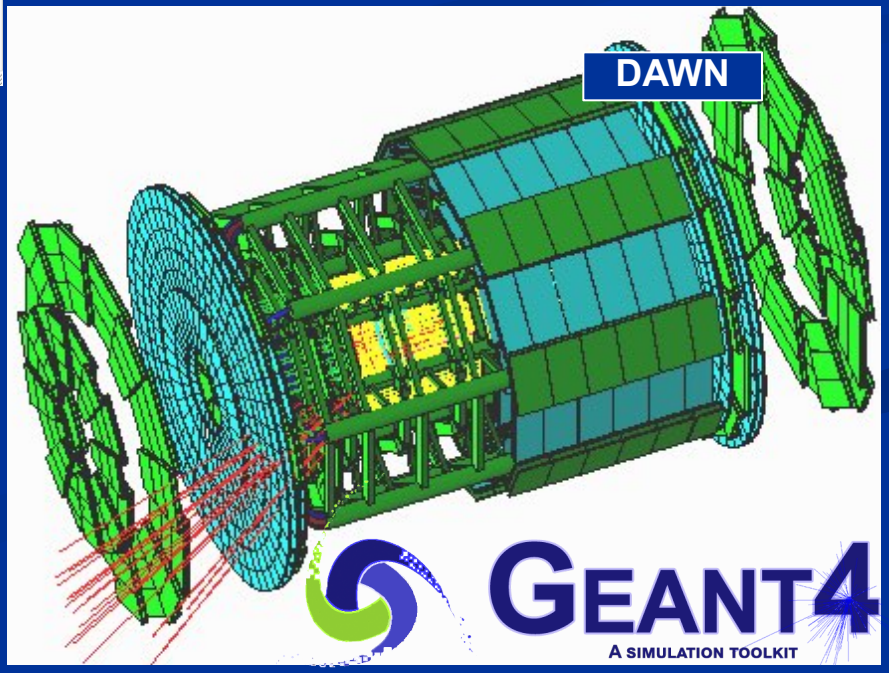


Igor Semeniouk

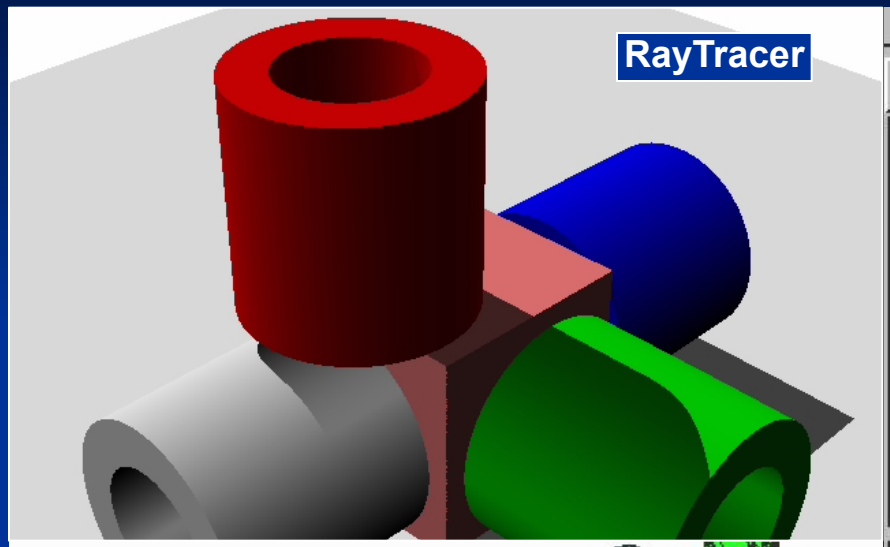
LLR, CNRS - Ecole Polytechnique

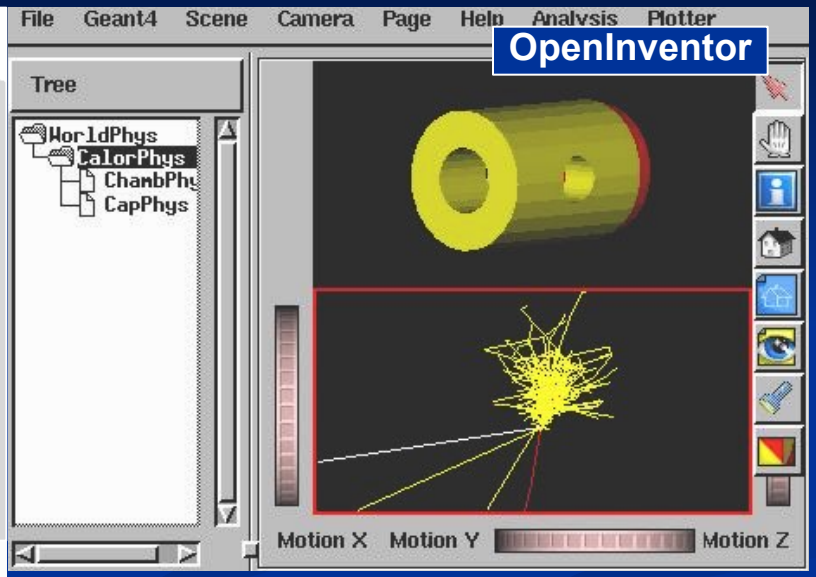
Slides from Laurent GARNIER (IRISA)

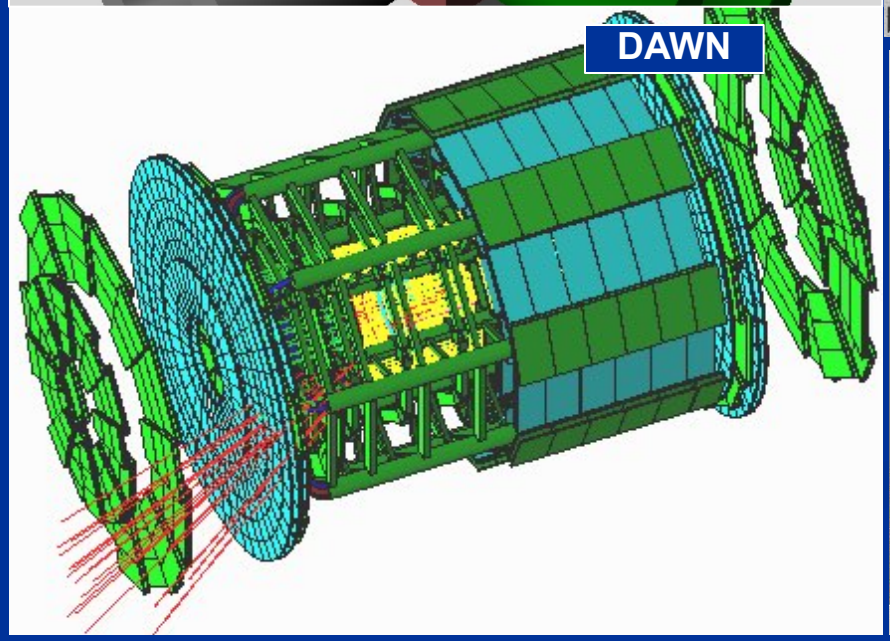
Based on Joseph Perl (SLAC) slides

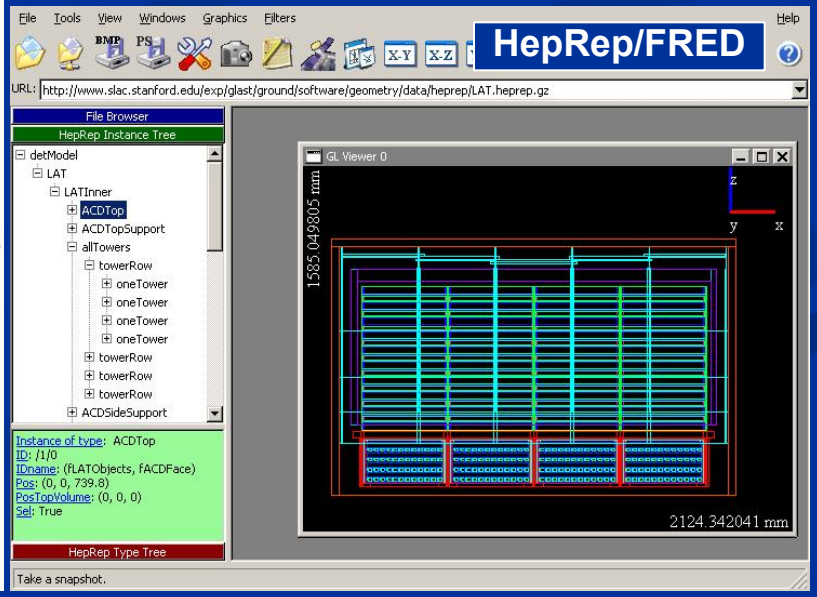


Geant4 Visualization Drivers

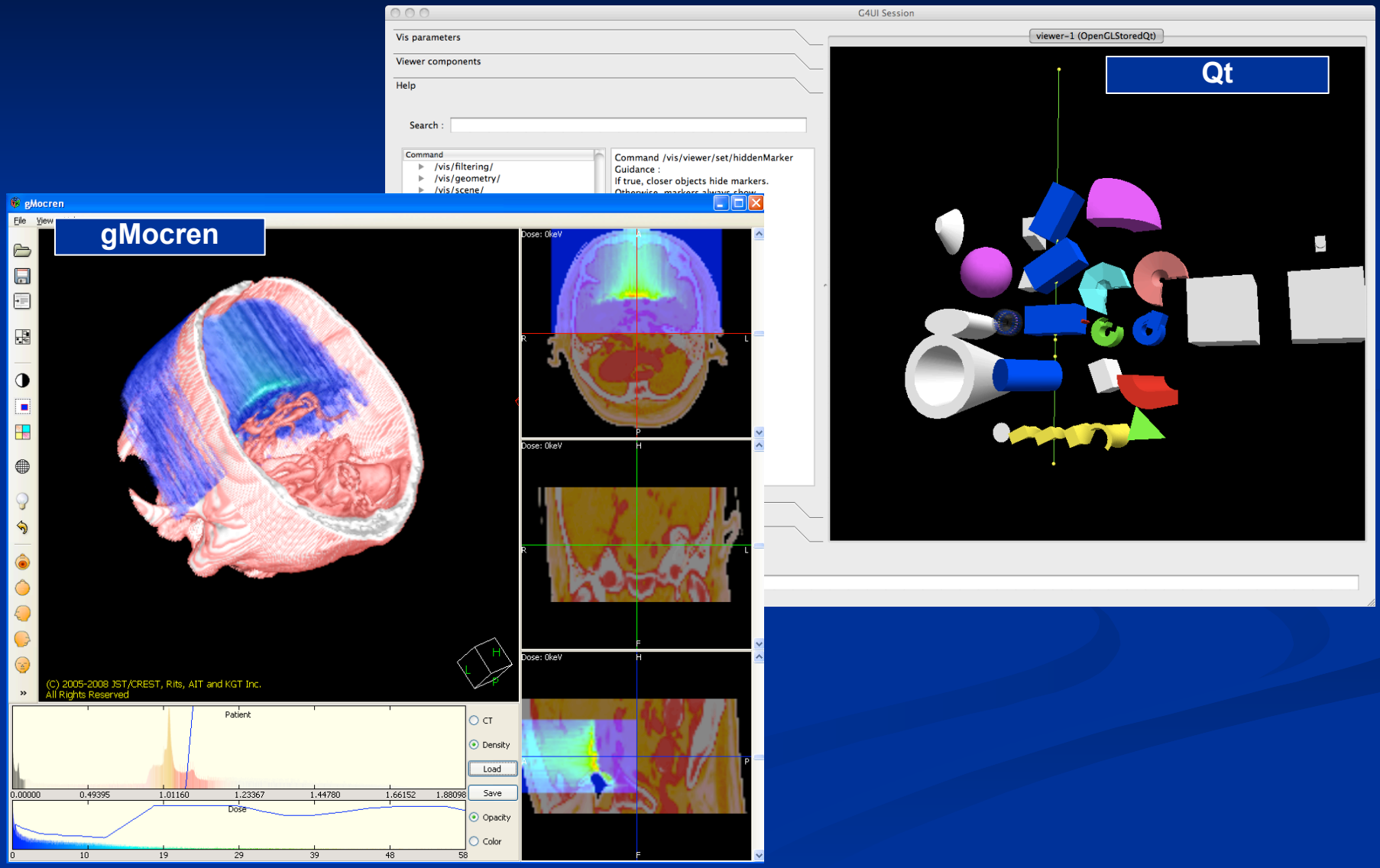




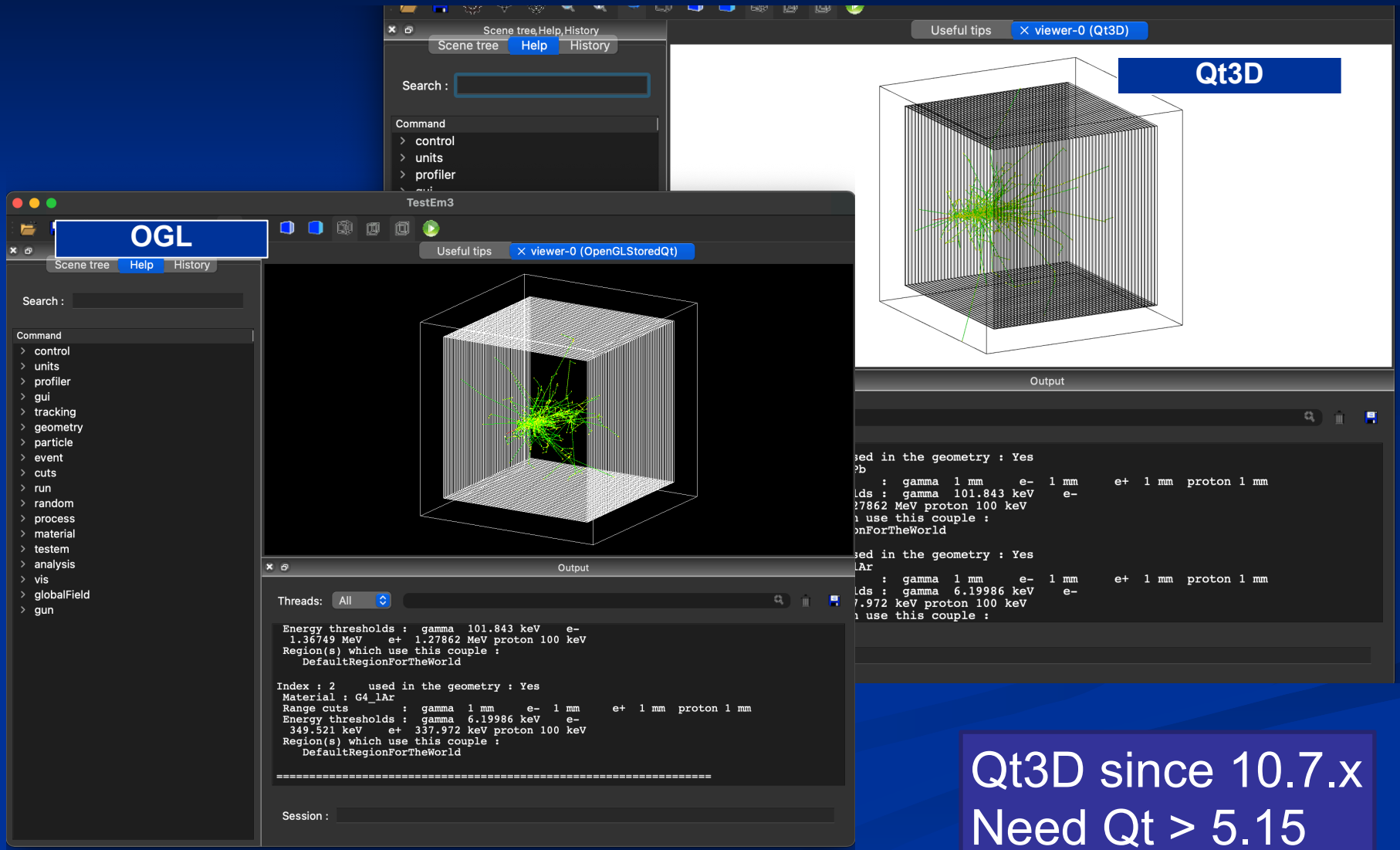




Geant4 Visualization Drivers



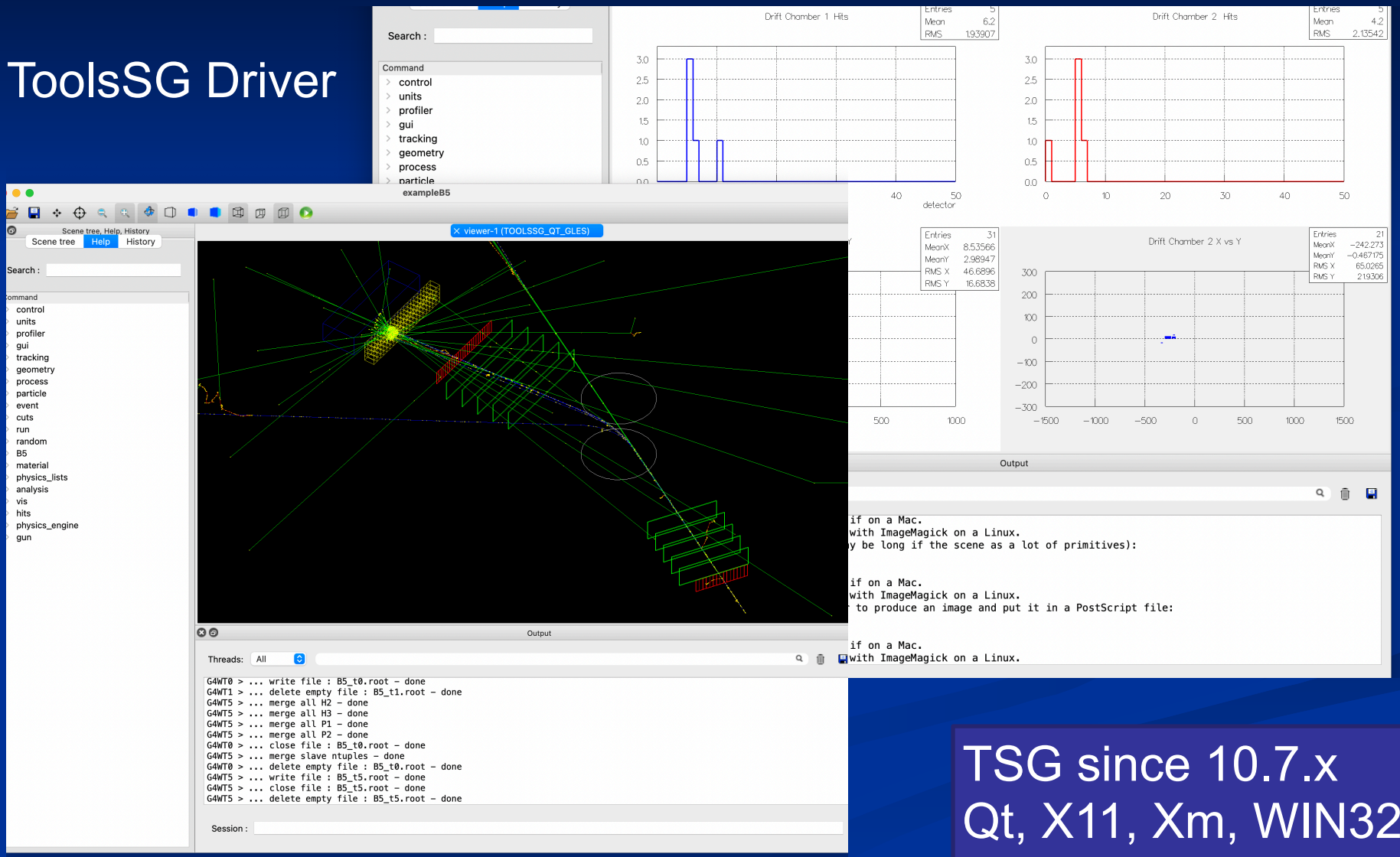
Geant4 Visualization Drivers



Qt3D since 10.7.x
Need Qt > 5.15

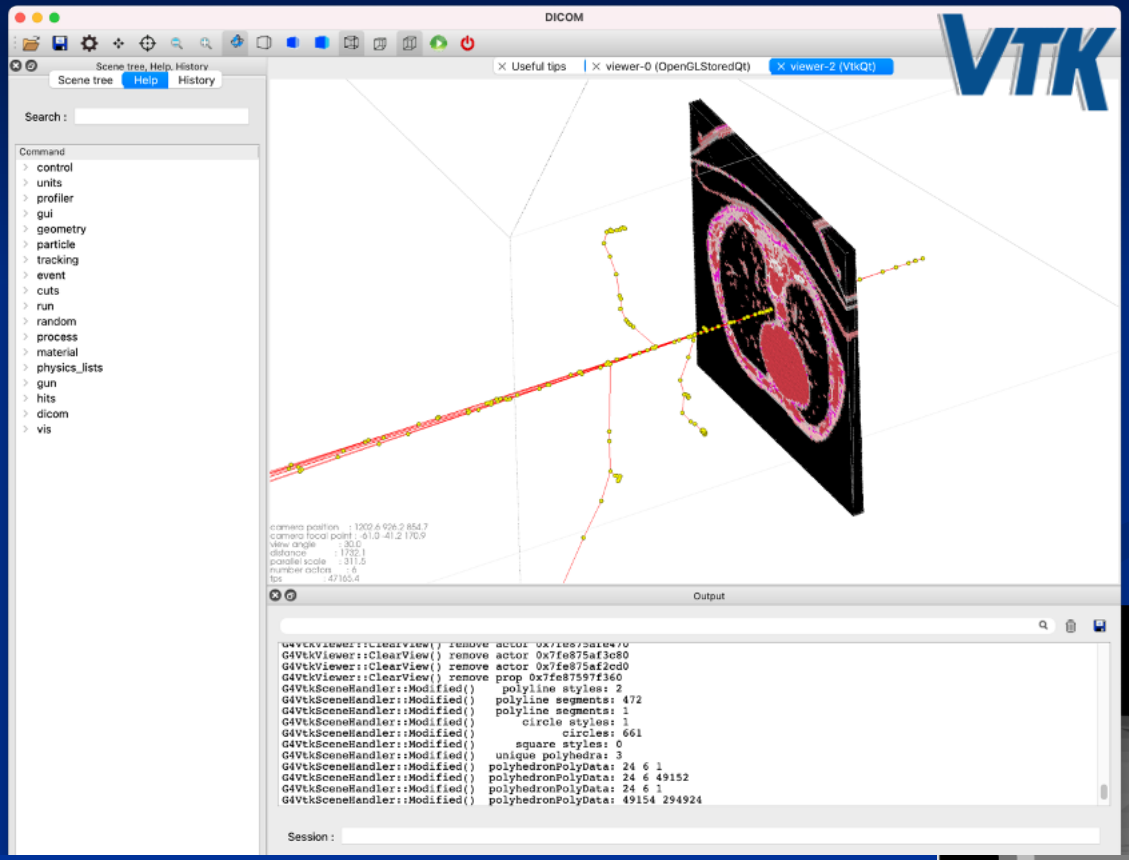
Geant4 Visualization Drivers

ToolsSG Driver



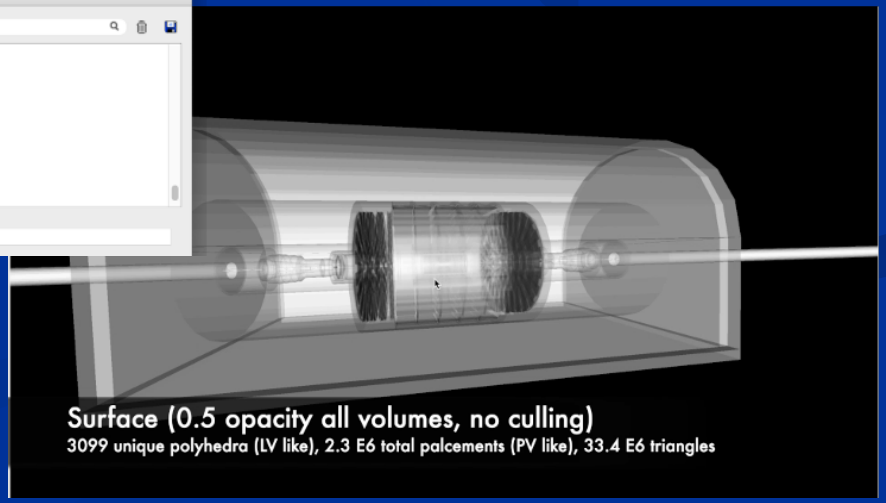
TSG since 10.7.x
Qt, X11, Xm, WIN32
Interactive Histos

Geant4 Visualization Drivers



VTK visualisation Driver
- Medical
- Large GDML Files
- OBJ/GLTF output

Since 10.7



What Can be Visualized

- Simulation data can be visualized:
  - Geometrical components
  - Particle trajectories and tracking steps
  - Hits of particles in the geometry
  - Scored energy, dose, etc.
  - Histograms Plotting

- Other user defined objects can be visualized:
  - Polylines
    - such as coordinate axes
  - 3D Markers
    - such as eye guides
  - Text
    - descriptive character strings
    - comments or titles

Visualization Driver Choices 1

- First I'll explain why there are so many visualization driver choices
- If you want more details about each visualization driver, see "references" at the end of this presentation

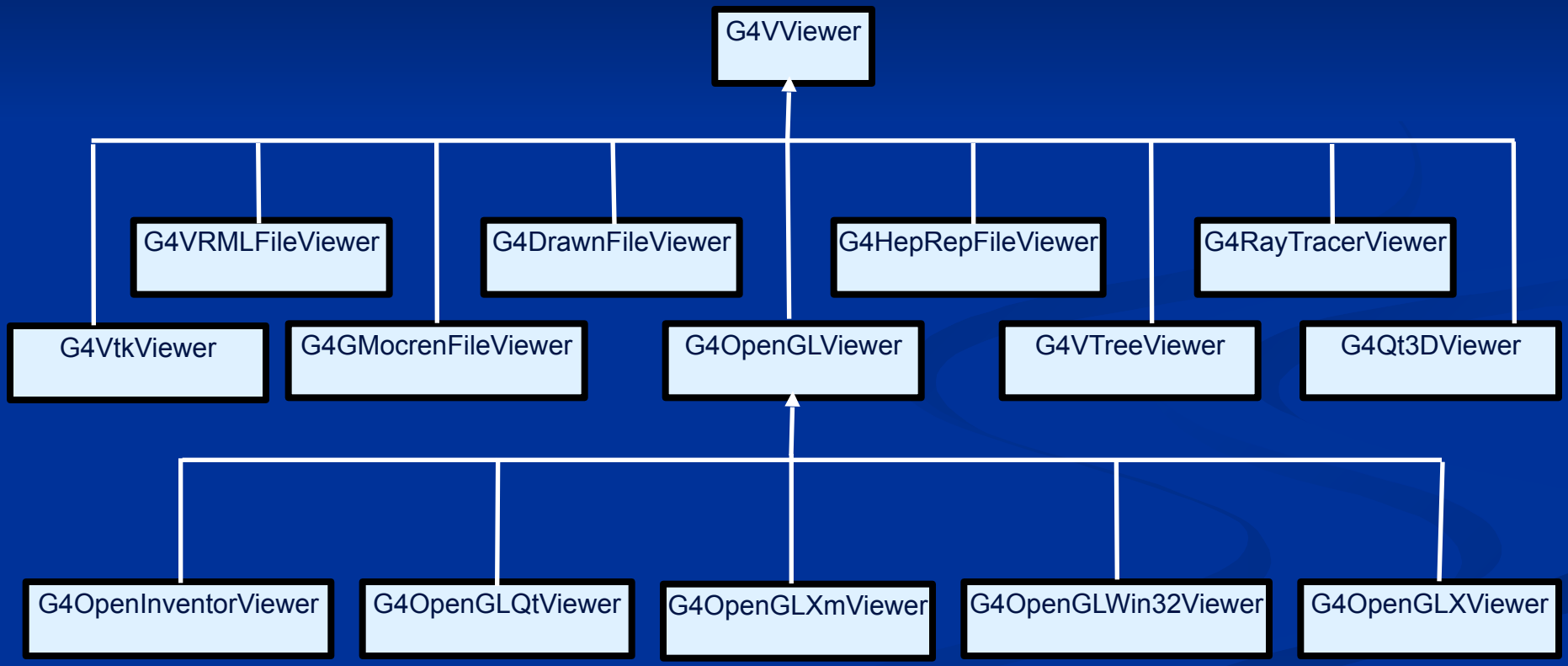
Driver	Variant	Hight quality print	Interactive	browse geometry hierarchies	Direct access to G4 kernel	Make movies	Web/3D
OpenGL	X						
	Xm						
	Qt						
	Win32						
OpenInventor	Xt						
	Win32						
	Qt						
Qt3D	Qt						
ToolsSG	X						
	Xm						
	Qt						
	Win32						

Visualization Driver Choices 2

- First I'll explain why there are so many visualization driver choices
- If you want more details about each visualization driver, see "references" at the end of this presentation

Driver	Variant	Hight quality print	Interactive	browse geometry hierarchies	Direct access to G4 kernel	Make movies	Web/3D
VTK	Qt						
	Native						
VRML							
DAWN							
gMocren							
RayTracer							
ACSII File							

Class hierarchy diagram



Many Visualization Drivers

- No Single Visualization Solution Can Meet all of Our Demands
- Quick response with flexible camera control
- High-quality Output for Publications
- Interactive Picking to Get More Information
- Complex Boolean Solids and Transparent or Reflective Surfaces
- 3D Format Suitable for Web Distribution
- Visualize Volume Data
- Understand Geometry Hierarchies

- By exploiting the same interface design that we need anyway to support visualization systems of existing frameworks
- We are able to take advantage of the best features of several different visualization drivers
- With a common set of user commands
- And minimal maintenance for many of the drivers

- We take advantage of the best features of many pre-existing visualization systems without having to reinvent those systems.

Controlling Visualization

- Your Geant4 code stays basically the same no matter which driver you use
- Visualization is performed either with commands or from C++ code
  - For the present tutorial, we confine ourselves to command-driven visualization.
- Some visualization drivers work directly from Geant4
  - OpenGL
  - OpenInventor
  - Qt3D
  - VTK
  - ToolSG
  - RayTracer
  - ASCIITree
- For other visualization drivers, you first have Geant4 produce a file,
  and then you have that file rendered by another application (which may have GUI control)
  - HepRepFile
  - DAWNFILE
  - VRML2FILE
  - gMocrenFile
- You can open more than one driver at a time.
  - For example, do a quick check in OpenGL,
    then save the same event for a beautiful DAWN plot

Controlling Which Drivers are Available

- Six of the visualization drivers are always included by default (since they require no external libraries):
  - HepRepFile
  - DAWNFILE
  - VRMLFILE
  - RayTracer
  - gMocrenFile
  - ASCIITree
- Other visualization drivers are included only if appropriate flags are set in CMake
- You can also add your own visualization driver.
  - Geant4's visualization system is modular. By creating just three new classes, you can direct Geant4 information to your own visualization system.

Simplest command Example

- Visualize your geometry in OpenGL:
- /vis/open OGL
- /vis/drawVolume

- Most examples come with a visualization macro more complete (including our exercise), which will be explained in more details later

To Open Visualization

- To Open a Driver
  - /vis/open <driver name>
- for example
  - /vis/open OGL
  - /vis/open DAWNFILE
  - /vis/open HepRepFile
  - /vis/open VRML2FILE
- The set of available drivers is listed when you first start Geant4, but you can also get this list with the command:
  - help /vis/open

OpenGL Additional Modes

- For all OpenGL drivers, 2 modes available :
  - Immediate mode
    - draws only to screen, no “memory”; detector can be redrawn after view changes but event data is lost.

    => Slow if you want to rotate/move the scene
  - Stored mode
    - Stored mode: creates graphical database (display lists); nothing is lost on simple operations like change of viewing angle

    ⇒ Slower at first draw, but faster after if you want to rotate/move the scene

exampleB2b run	Immediate mode	Stored mode
command	/vis/open OGLI	/vis/open OGLS
First draw time (sec)	1	15
Frame per sec	1	12

References

Users Guide For Application Developer - Visualization

http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/Visualization/visdrivers.html

DAWN download

https://geant4.kek.jp/~tanaka/