

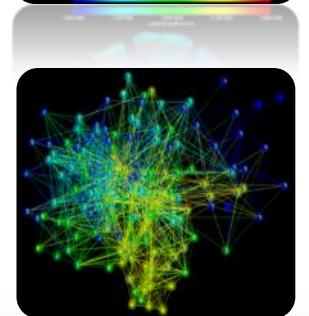
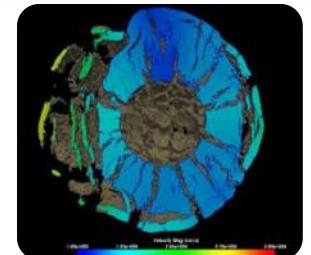


# Scientific Visualization with Open Source Tools

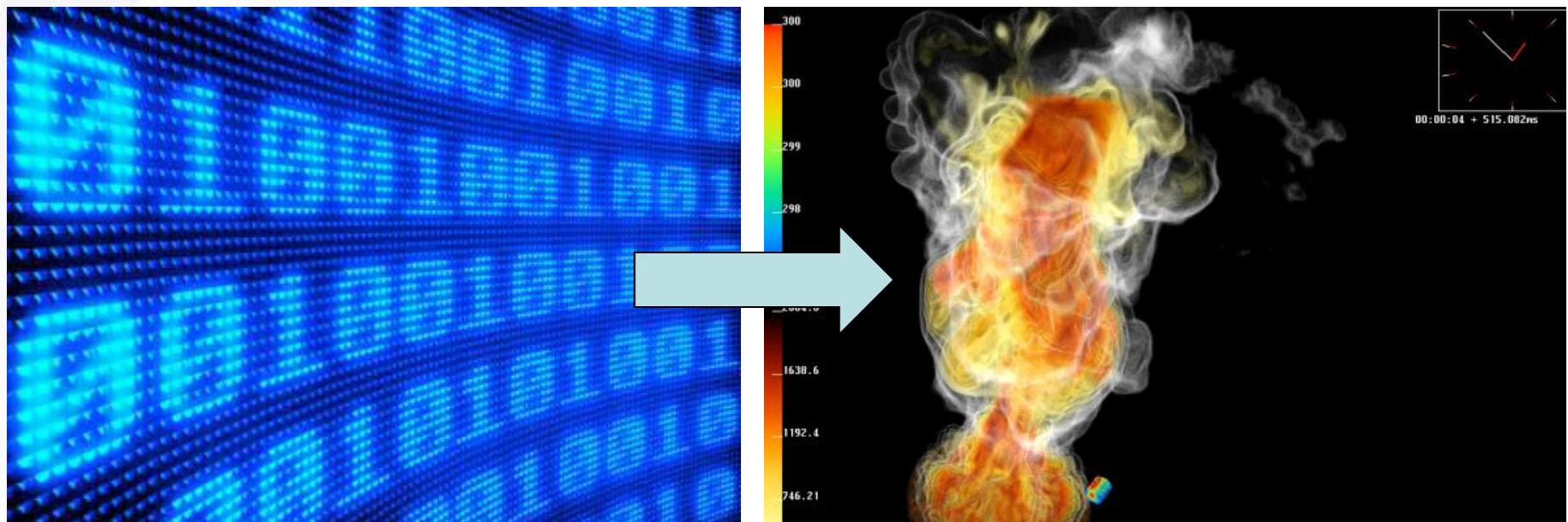
NGCM Summer School 2015

Jérôme Velut

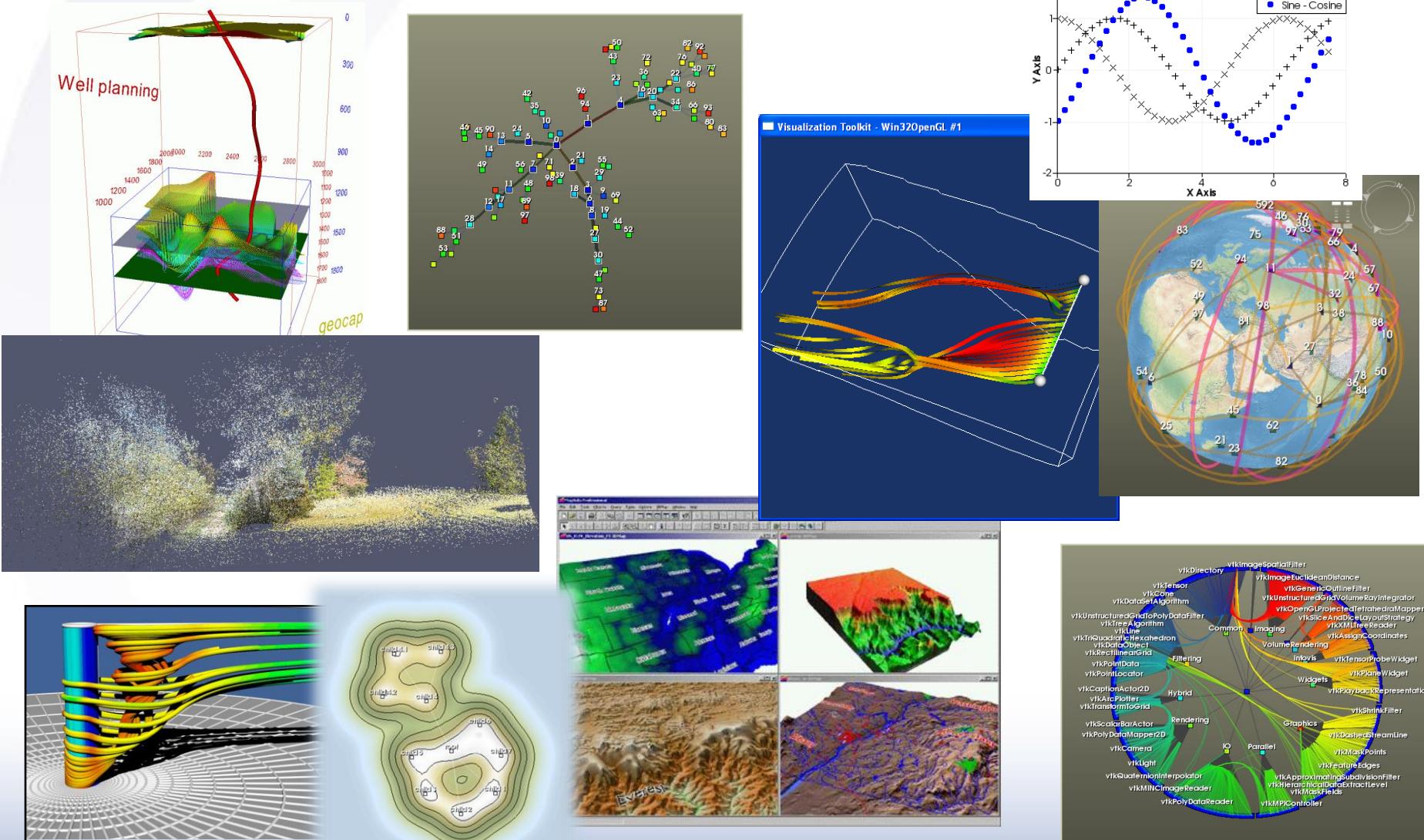
[jerome.velut@kitware.com](mailto:jerome.velut@kitware.com)



# Visualization is Communication



# Challenges of Visualization



# Challenges of Visualization

- Heterogeneous data
- Large/many/big data
- Distributed data
- Computing resources
- Domain-specific
- User-specific
- Heterogeneous devices
- Uncertainties

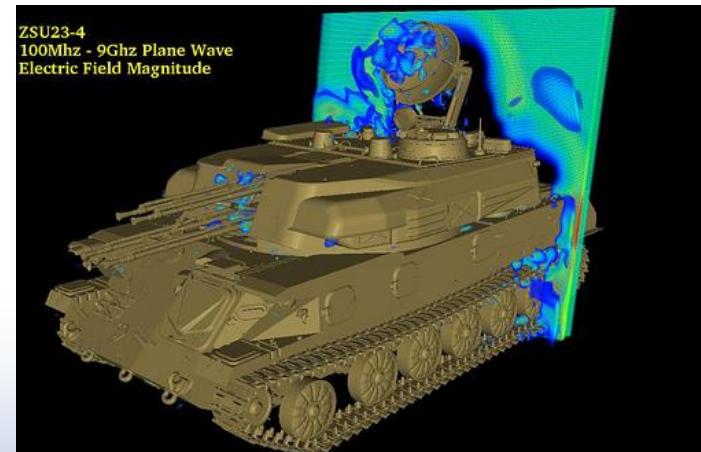
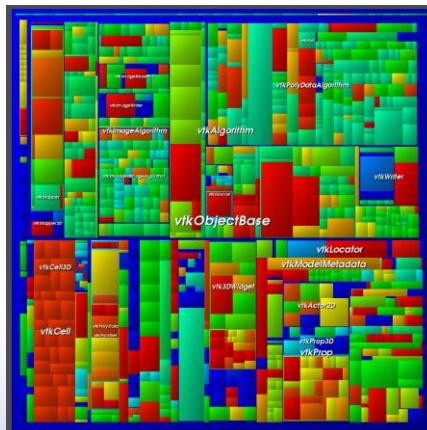
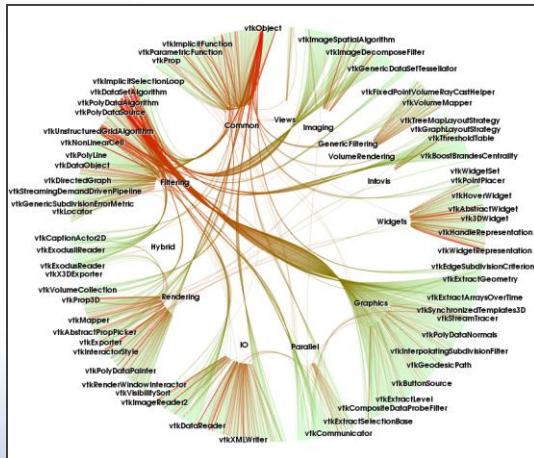


# **HOW TO OVERCOME THESE CHALLENGES**



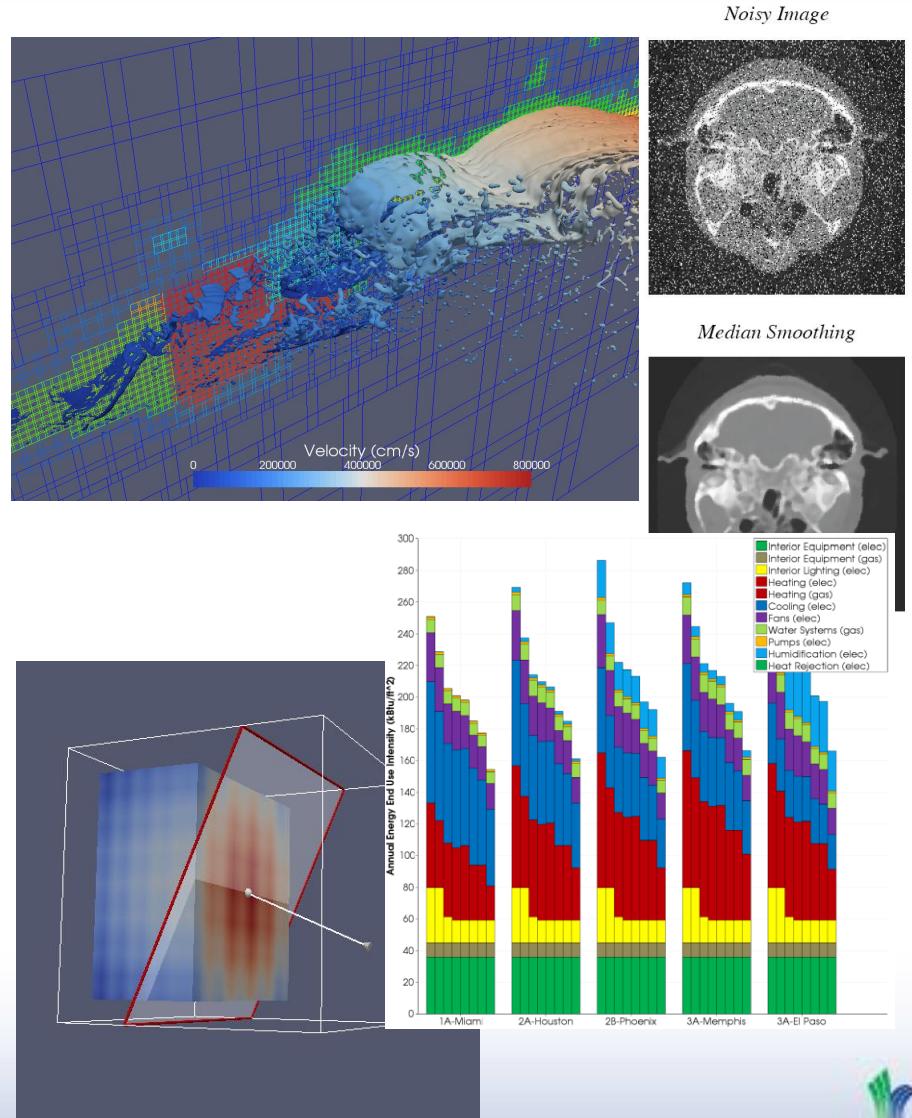
# The Visualization Toolkit (VTK)

- [www.vtk.org](http://www.vtk.org)
- Started in 1993 at GE
- Visualization Library
  - Written in C++ (+5.5 million LOC) – BSD License
  - Automatic binding for Java, TCL, Python
  - Portable by design: Linux, Windows, Mac OSX, Solaris...
- Very active community: 4000+ users on the mailing list



# What can VTK do for me?

- Sci Vis: 2 to 4D data processing and (volume) rendering
- Image processing
- Info Vis (0D)
- Charting/plotting
- Application support (GUI support, Widgets)



# Design Philosophy

- Underlying theme is to **process data**
  - Find the salient features
  - Produce imagery that conveys meaning
- An open-ended architecture used to **construct programs**
  - These programs usually give interactive controls to the user
  - Let the end user do the searching, **visually**
- Modular architecture
  - Modules implemented in Object-Oriented Classes
  - Pipeline: Data flows through modules in a pipeline
  - Lazy evaluation: Only process what is changed (for big data)



# VTK Main Components

- Data structures
  - How VTK stores/provides access to arbitrary data
- Algorithms/filtering pipeline
  - Manipulate data
  - Readers, sources, filters, writers, mappers
- Rendering classes
  - Display that data on the screen
  - Mappers, actors, lights, cameras, renderers, RenderWindows
- Interaction classes
  - Events, interactors, widgets
- Application support
  - Views, representations, Qt and MFC interfaces, wrapping

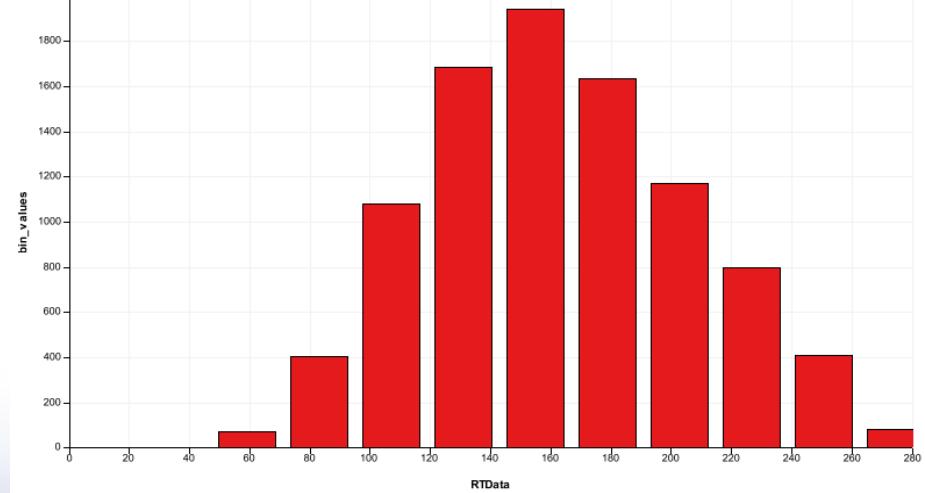
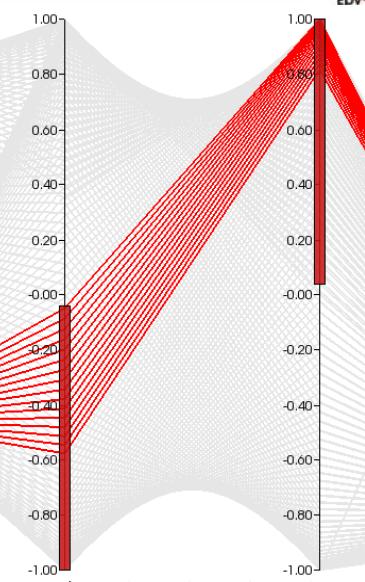
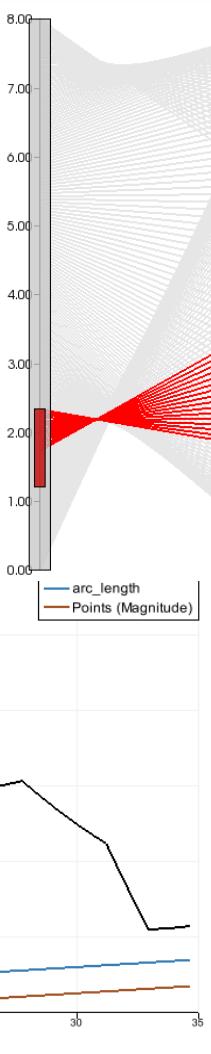
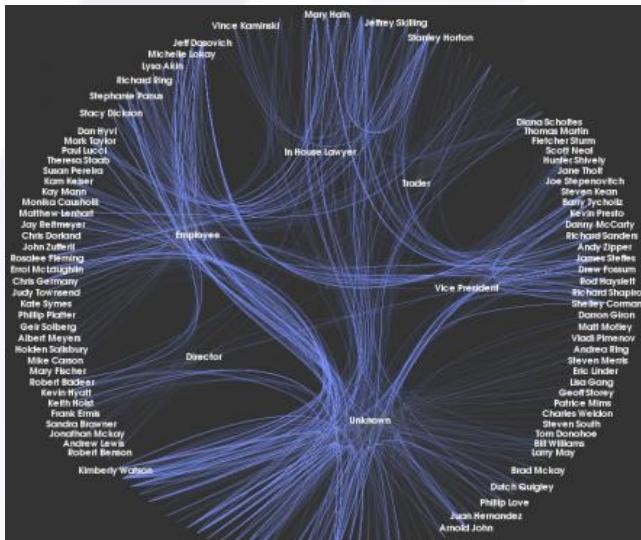


# VTK Main Components

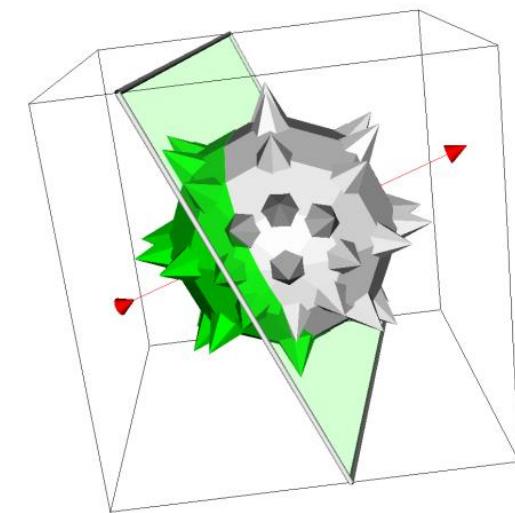
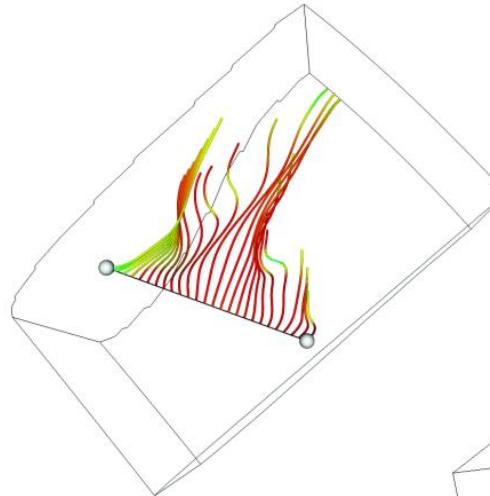
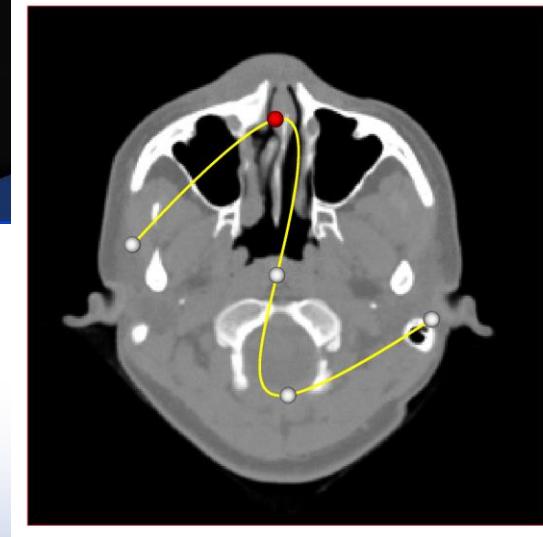
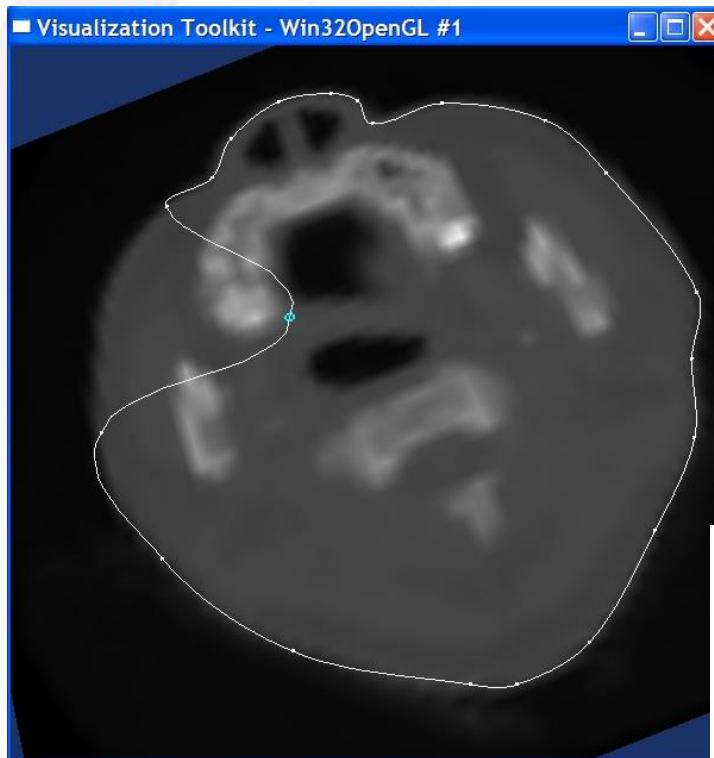
- **140 Readers**
  - STL, EnSight, TecPlot, BMP, JPEG...
- **150 Filters**
  - Contour, Subdivision, Delaunay, Elevation...
  - Statistics, Parallel, Geometry, FlowPaths, Extraction...
- **40 Widgets**
  - Distance, Angle, Plane, Seed, Checkerboard
- **Application domains**
  - Geo visualization
  - Chemistry
  - Imaging



# 2D: Graphs and Charts



# Interactive Widgets



# Example Code

```
#!/usr/bin/env python
```

```
import vtk
```

```
# Load an STL File
```

```
reader = vtk.vtkSTLReader()
```

```
reader.SetFileName("myfile.stl")
```

```
# Visualization Pipeline
```

```
mapper = vtk.vtkPolyDataMapper()
```

```
mapper.SetInputConnection(reader.Get  
OutputPort())
```

```
actor = vtk.vtkActor()
```

```
actor.SetMapper(mapper)
```

```
# Create a rendering window and renderer
```

```
ren = vtk.vtkRenderer()
```

```
renWin = vtk.vtkRenderWindow()
```

```
renWin.AddRenderer(ren)
```

```
# Create a renderwindowinteractor
```

```
iren = vtk.vtkRenderWindowInteractor()
```

```
iren.SetRenderWindow(renWin)
```

```
# Assign actor to the renderer
```

```
ren.AddActor(actor)
```

```
# Enable user interface interactor
```

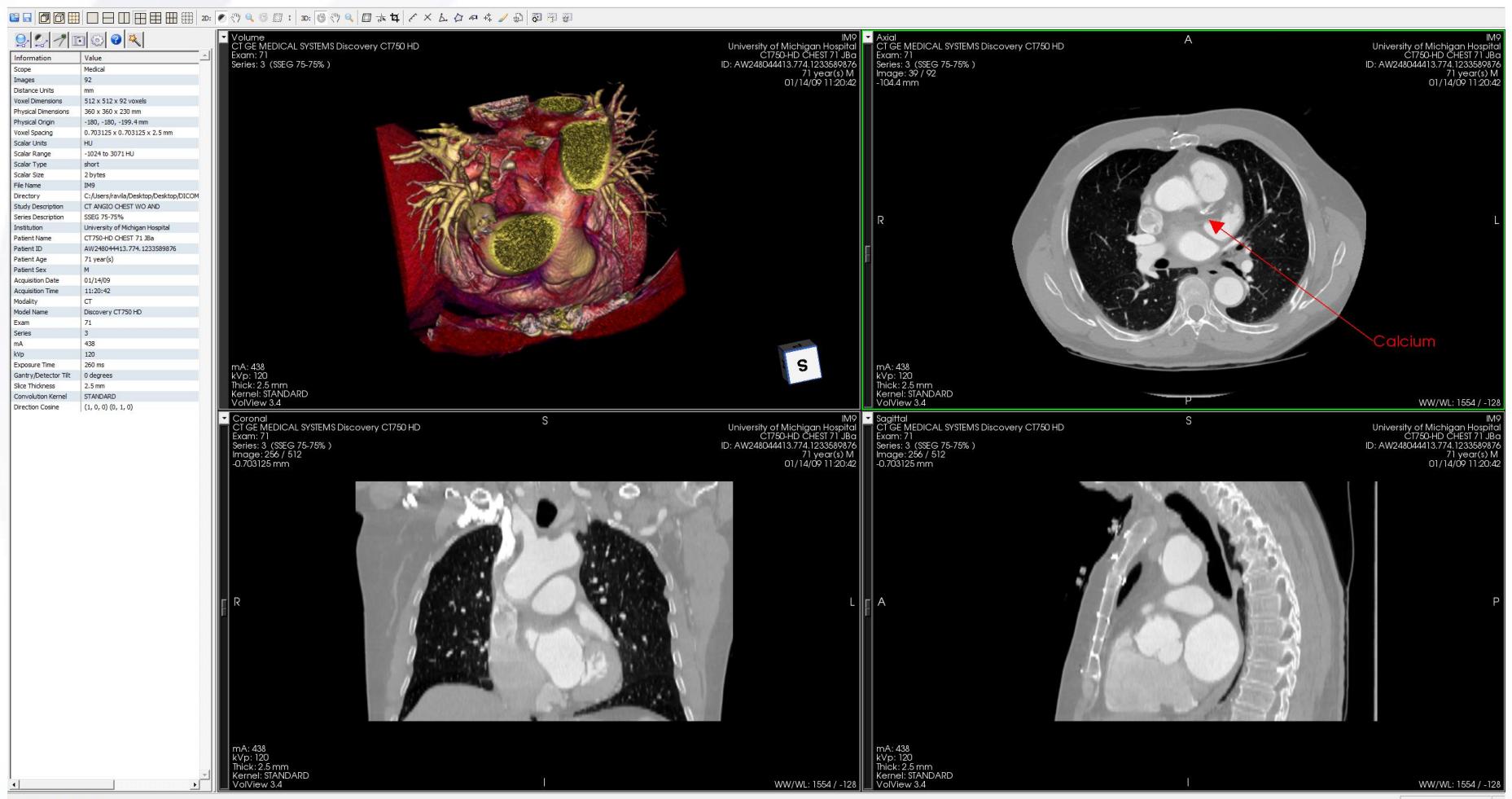
```
iren.Initialize()
```

```
renWin.Render()
```

```
iren.Start()
```



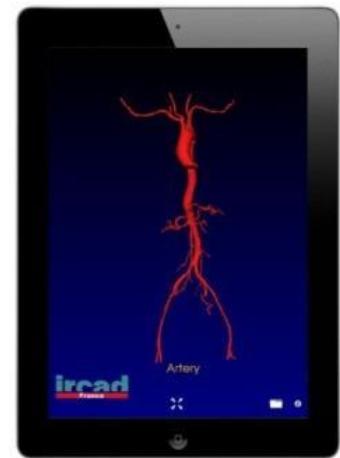
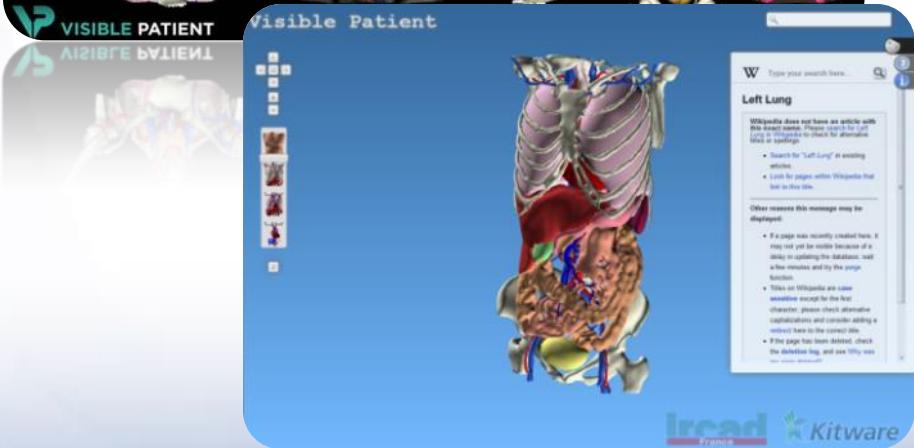
# Volume Rendering



# VTK in Use: Visible Patient



Visible Patient extracts the 3D models of patient



 Kitware  ircad

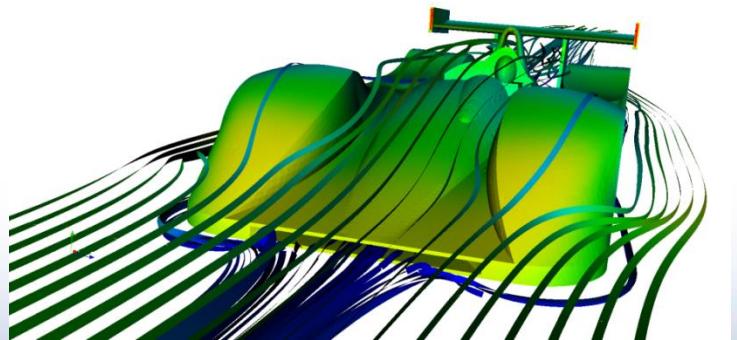
KITW916

# LARGE DATA VISUALIZATION



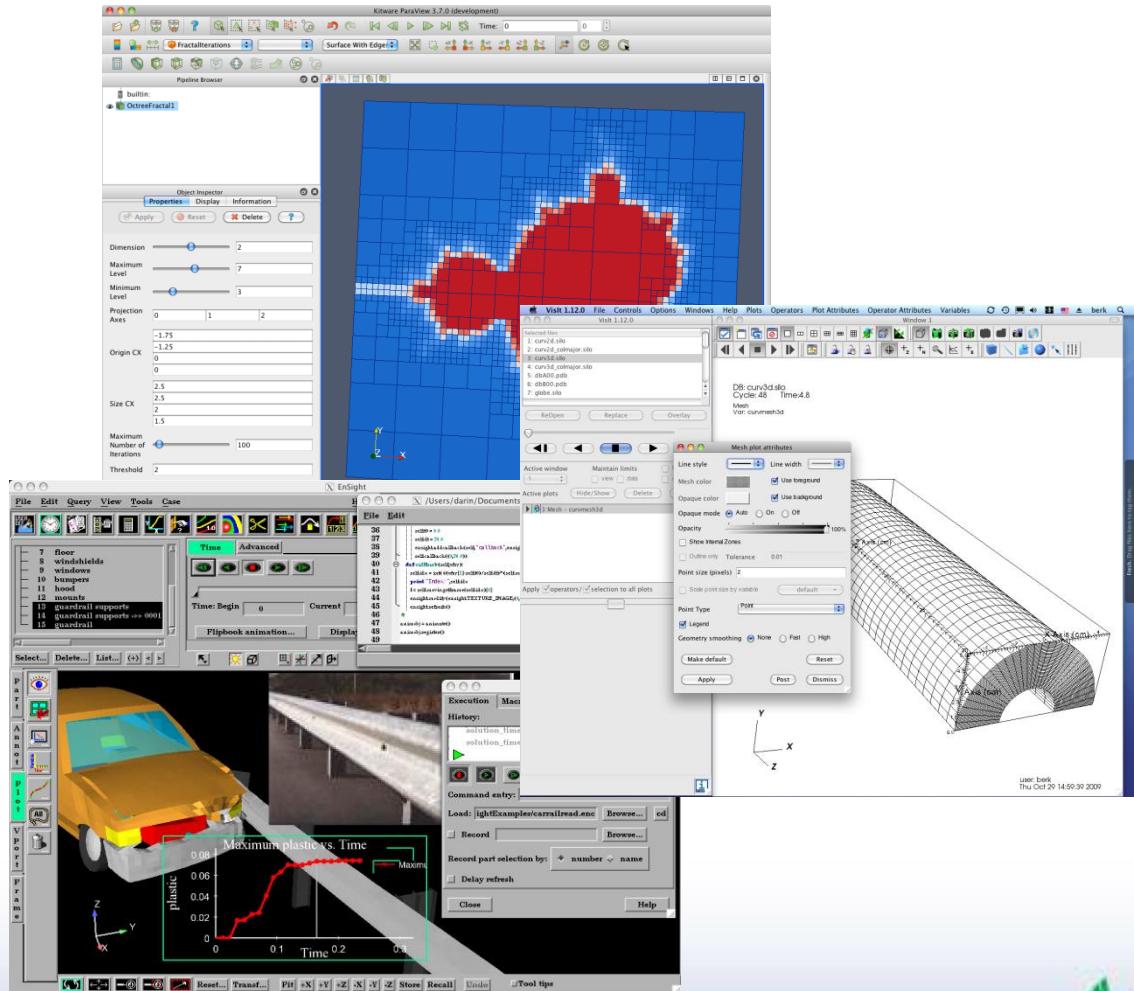
# ParaView

- [www.paraview.org](http://www.paraview.org)
- OpenSource (BSD)
- Based on VTK
- C++/Qt
- Cross-platform: Linux, Mac, Windows
- Python support
- Very active community (HPC Wire Award)
- Multi-core support (MPI)
- Co-processing (in-situ)
- More than 50 data readers



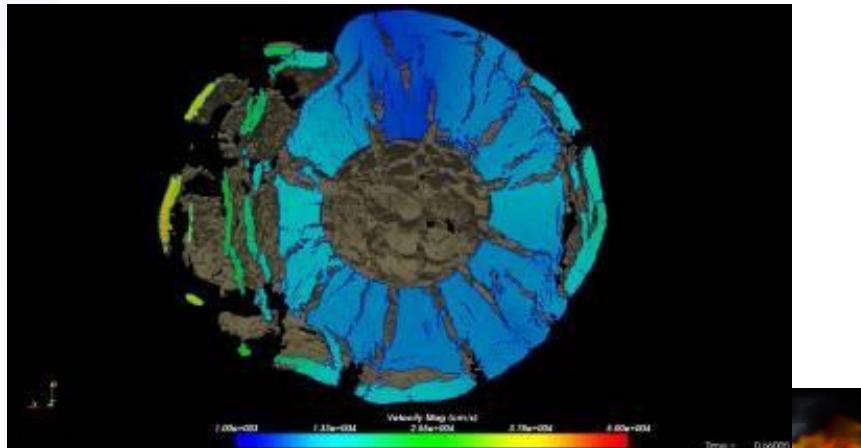
# General Purpose Tools

- EnSight
- ParaView
- VisIt
- FieldView
- TecPlot
- ...

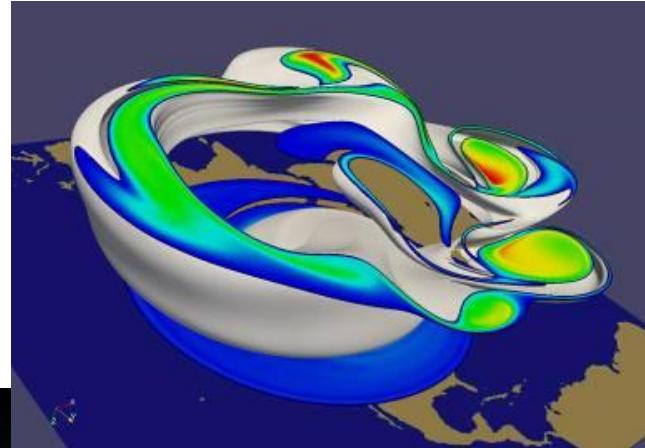


# ParaView

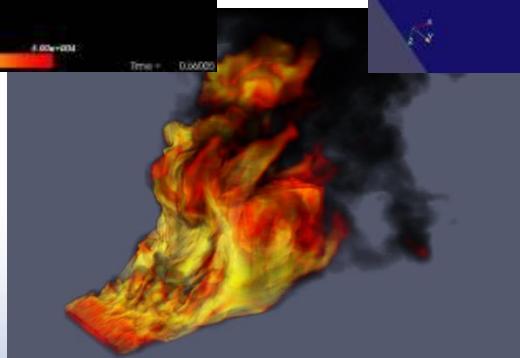
- An application and an architecture to visualize and analyze massive datasets
- A turn-key visualization application



1 billion cell asteroid  
detonation simulation

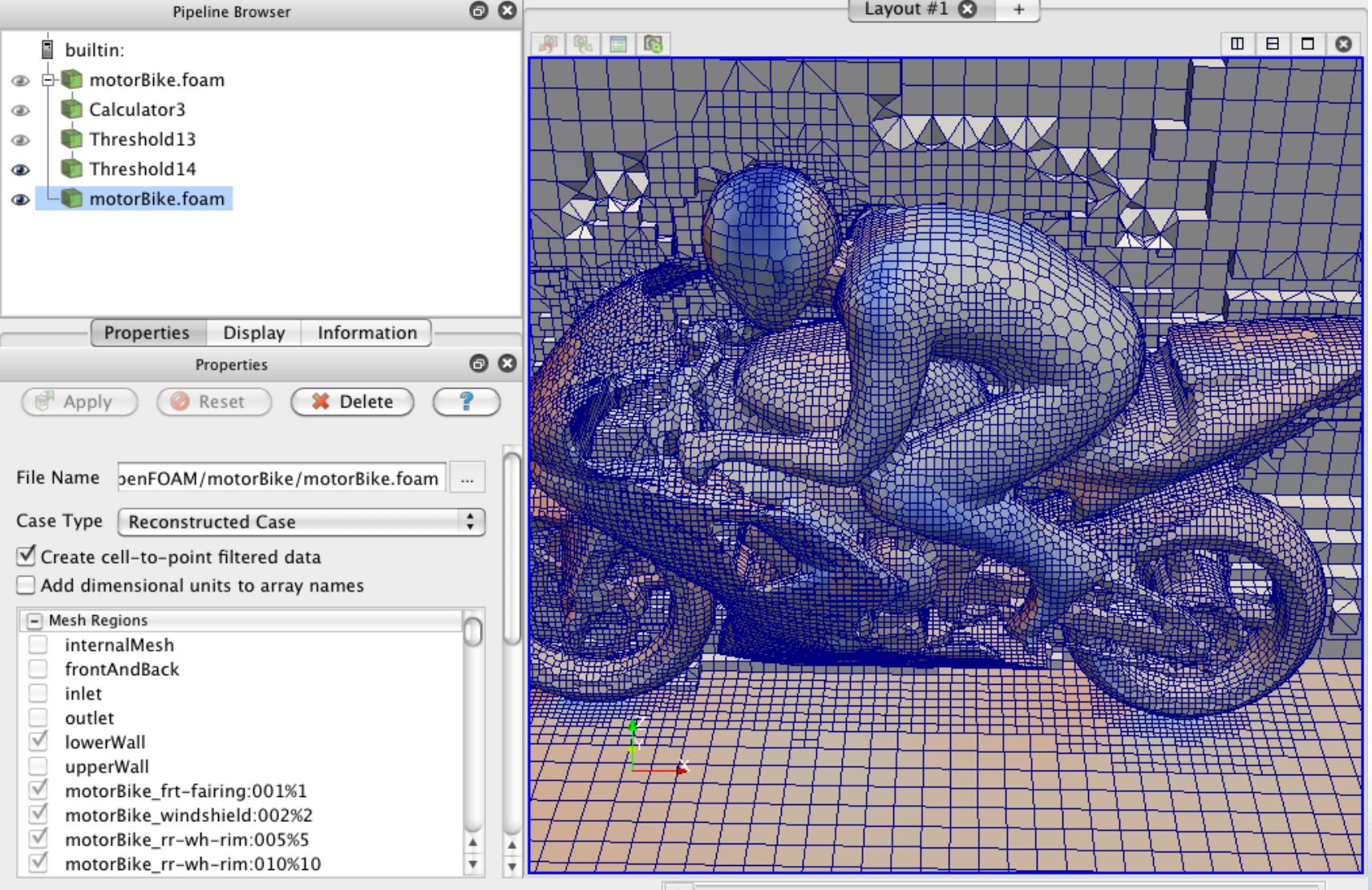
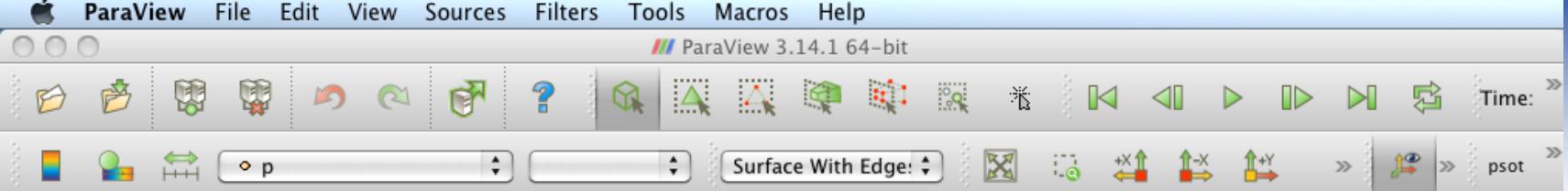


½ billion cell  
weather simulation



Fire simulation



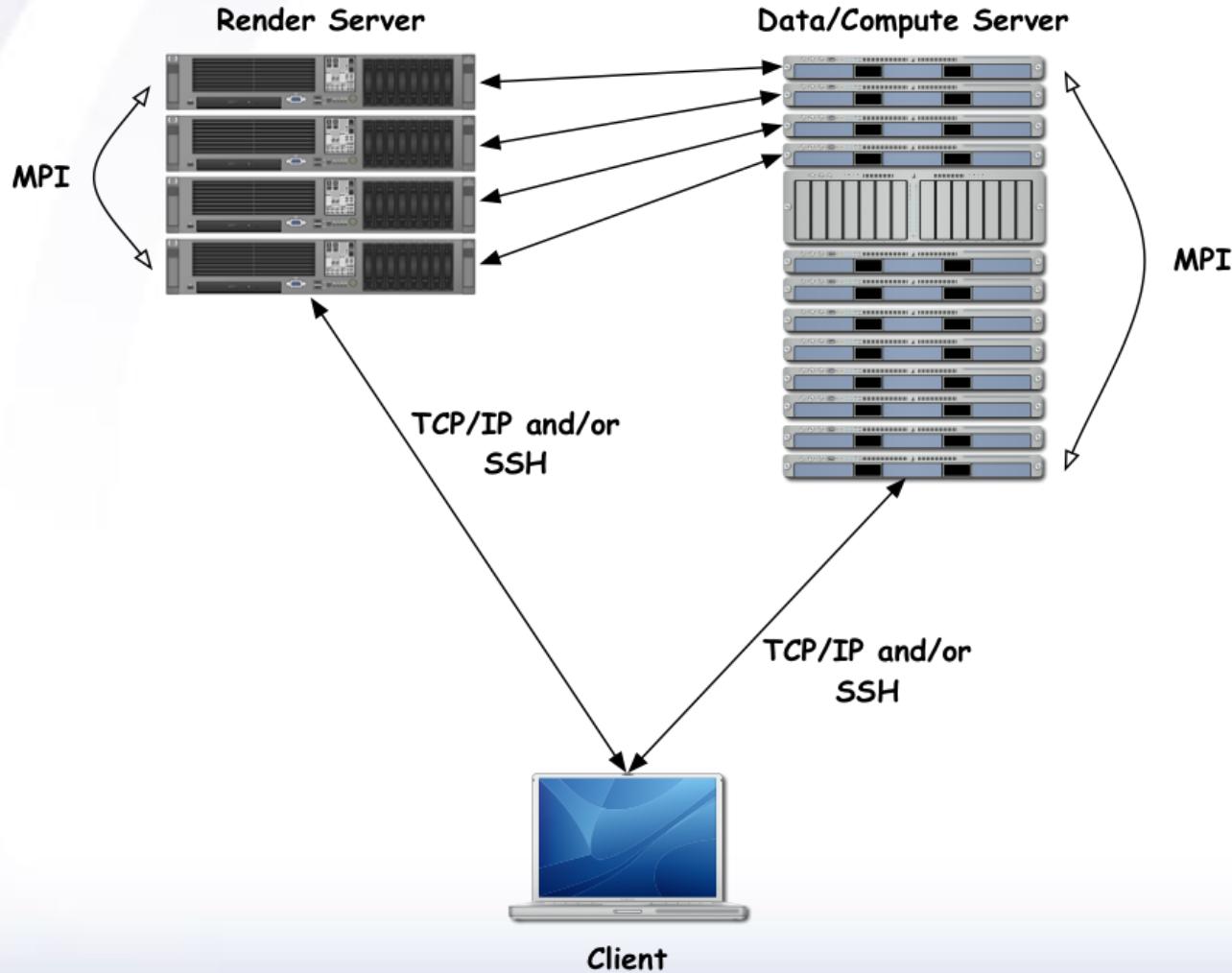


# ParaView is a Framework

- ParaView **extends VTK** to provide:
  - Client-server computing
  - State management
  - Python modules
  - Application/GUI framework
- ParaView framework can be used to **develop other applications**
- ParaView can be **embedded in other applications** and frameworks

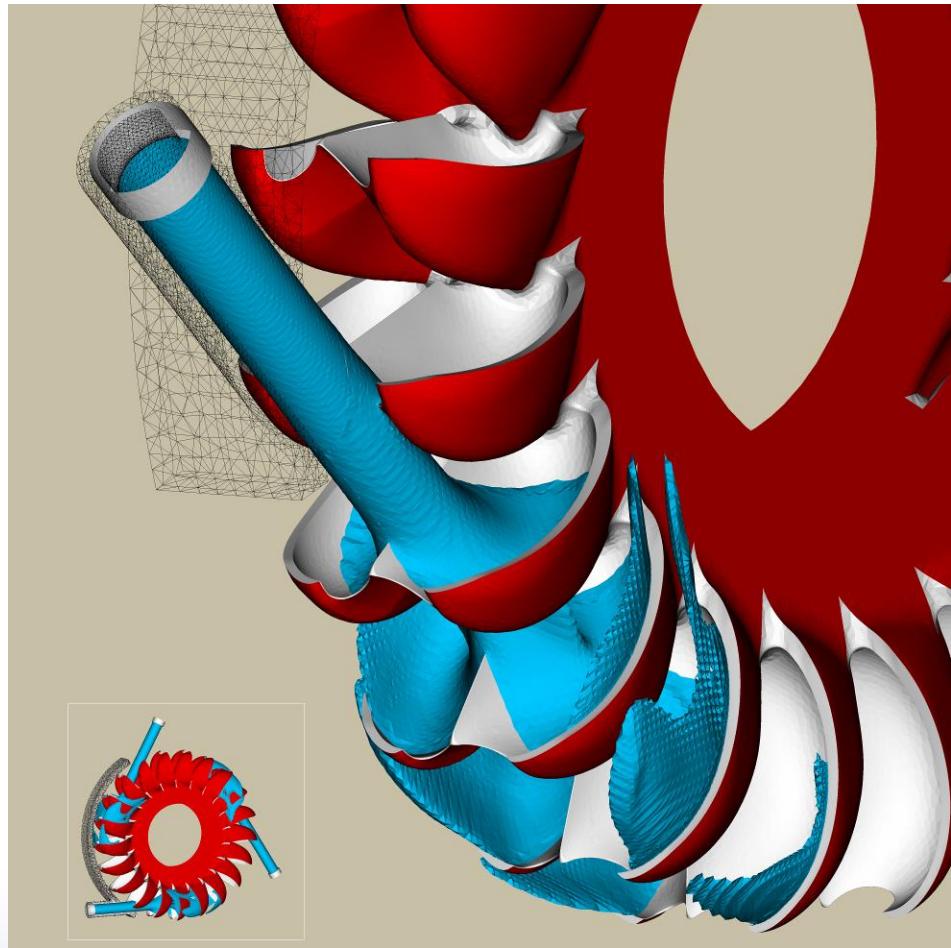
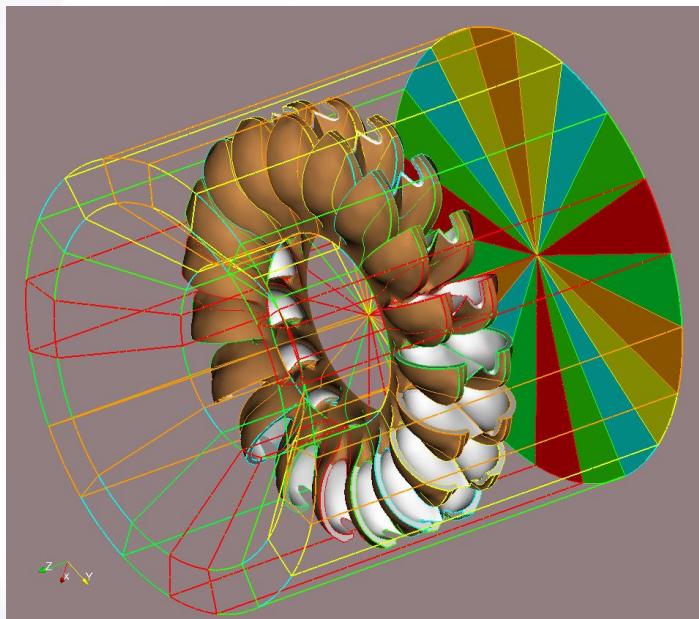


# ParaView Architecture



# Large Data - Unstructured

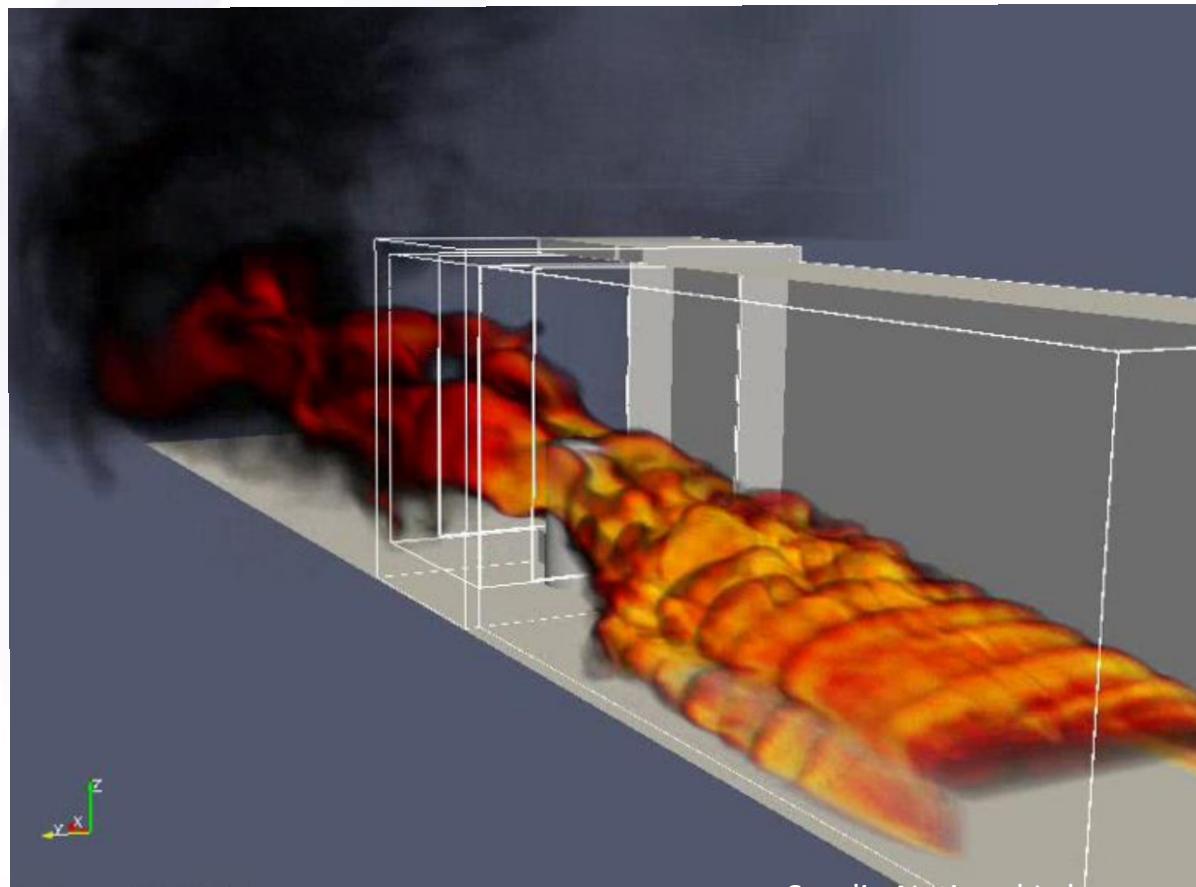
- CFD simulation
- 20-30 million elements
- Load balancing



source: Swiss supercomputing center



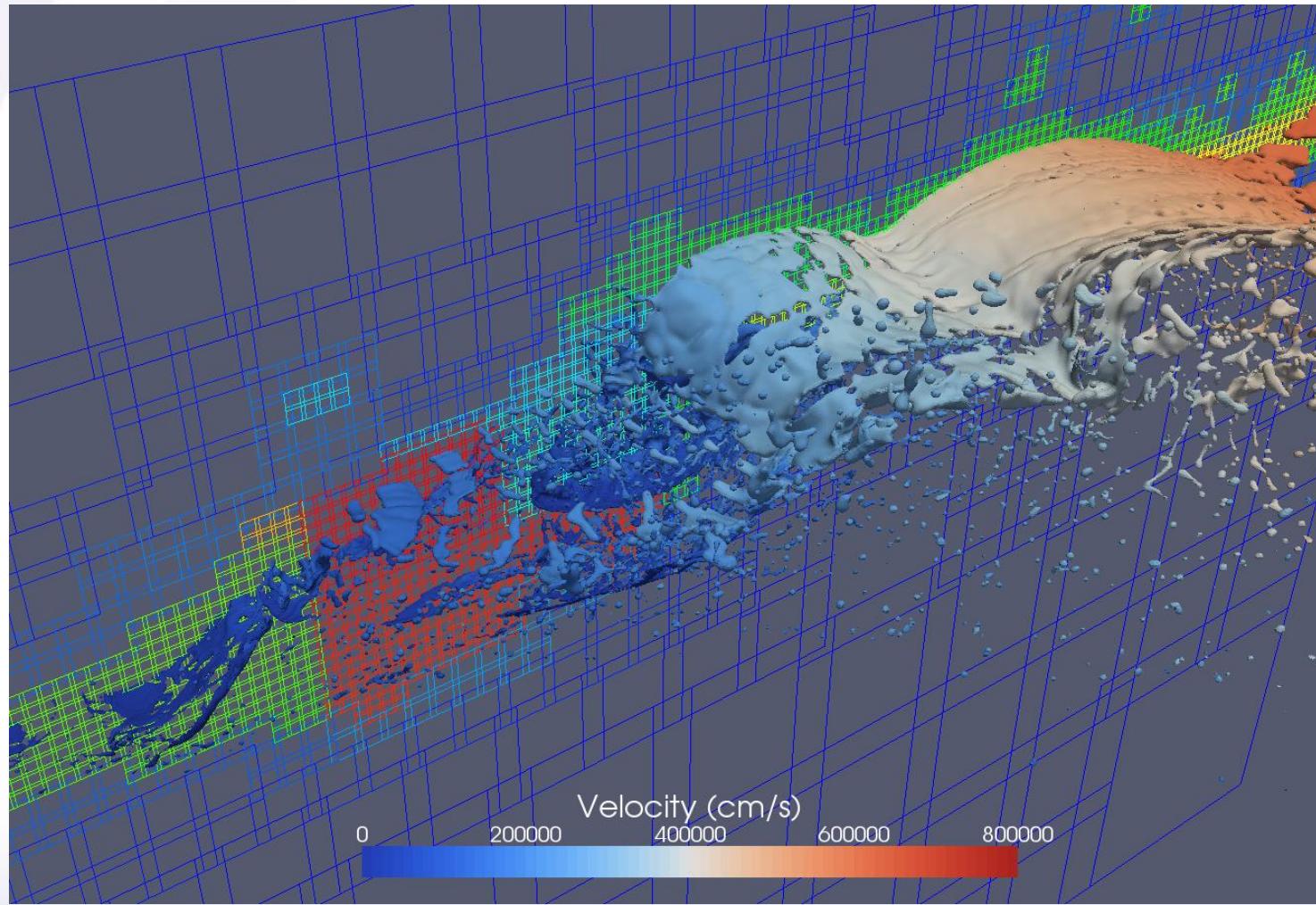
# Large Data - Unstructured



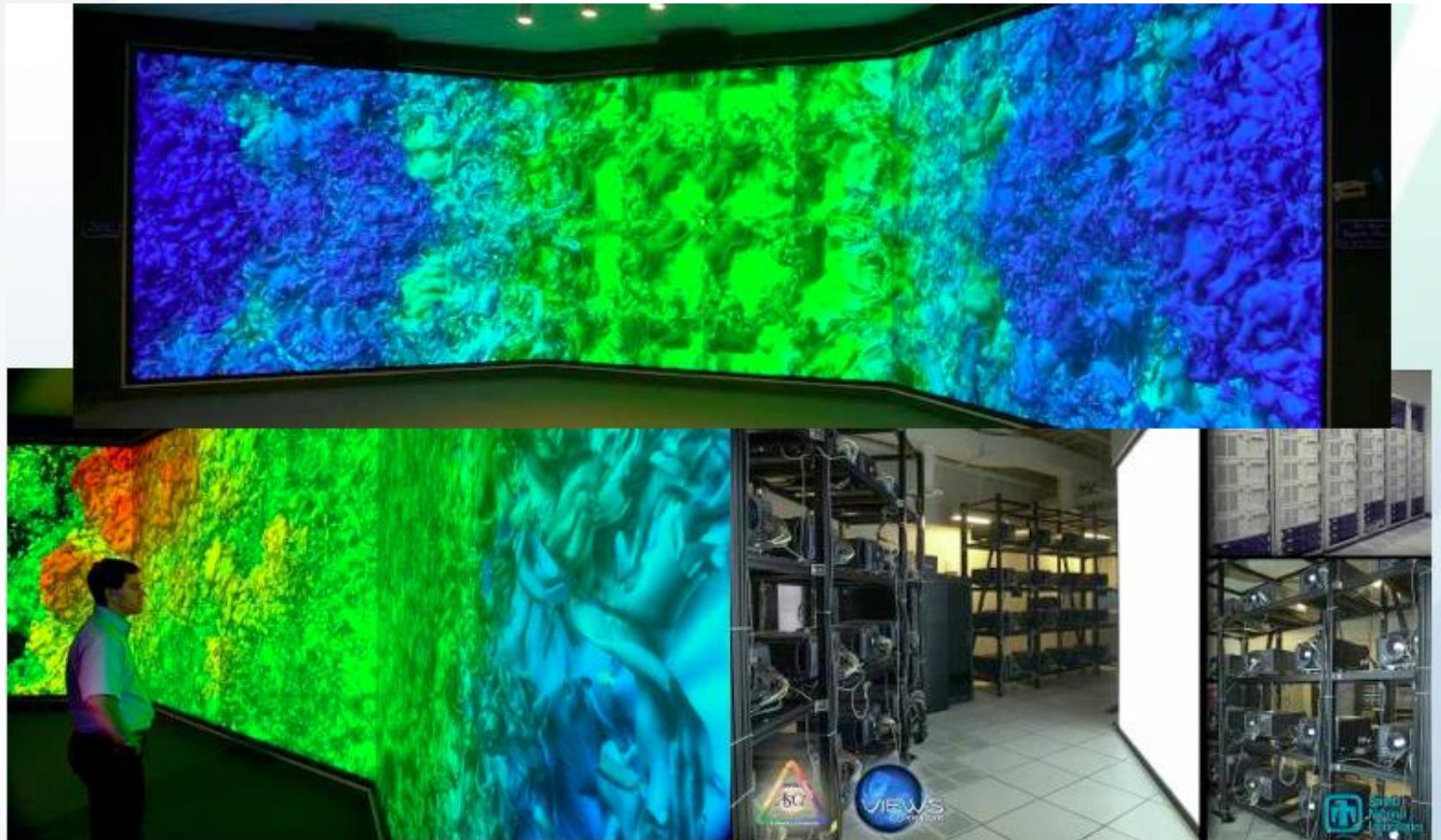
- Fire simulation
- 150 million elements



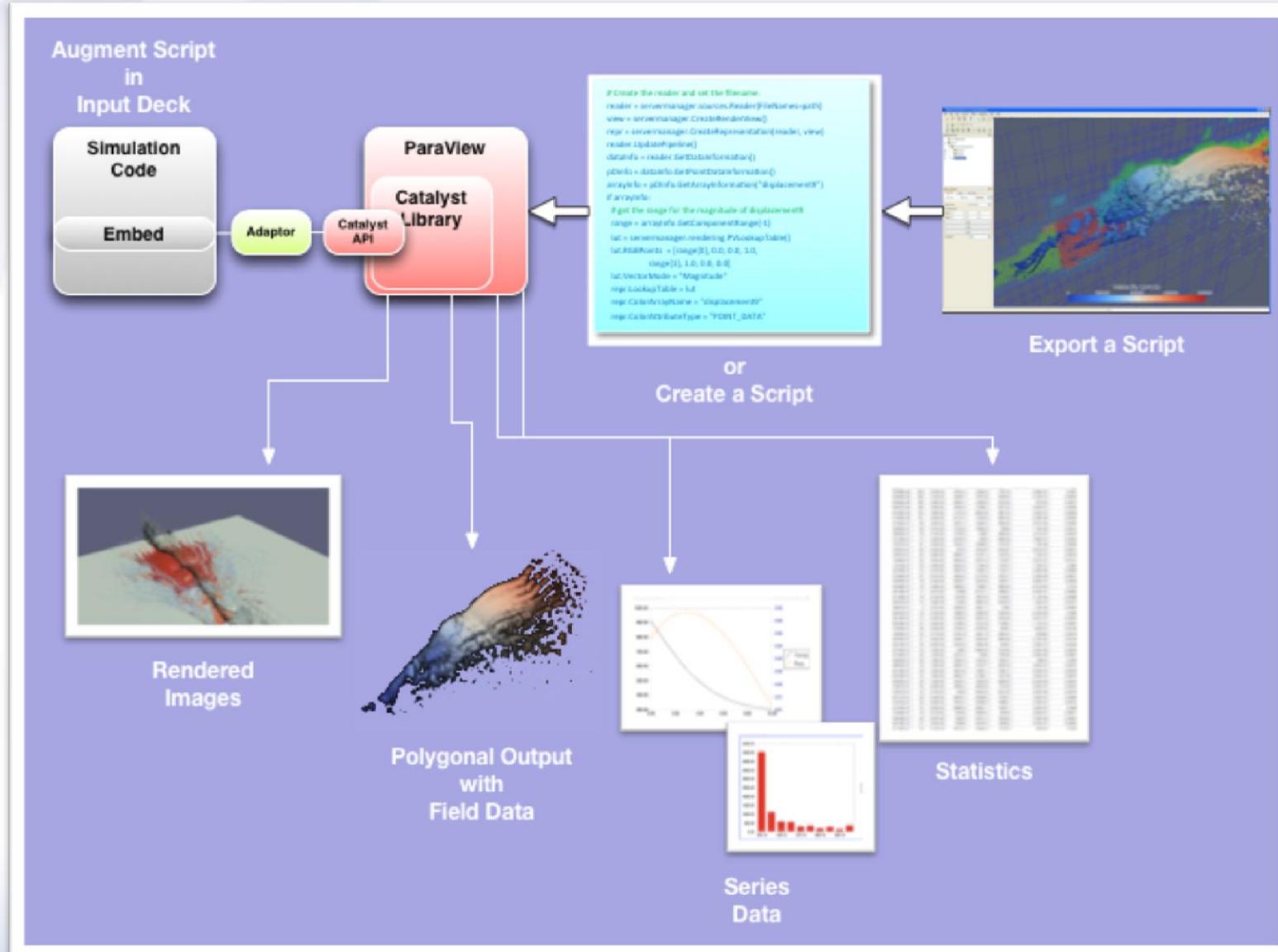
# Large-scale AMR



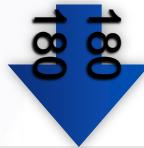
# ParaView in Use: Immersive Visualization



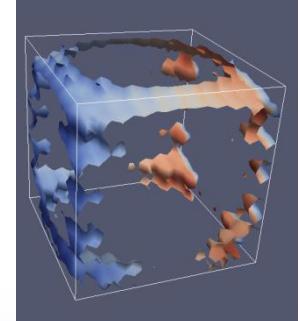
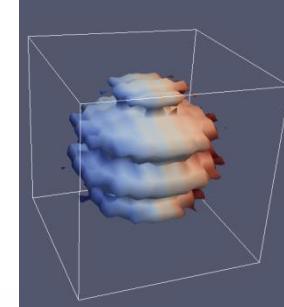
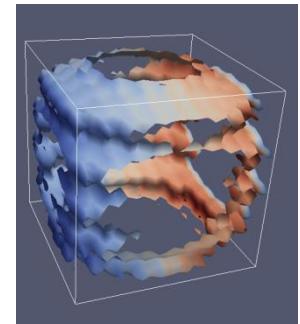
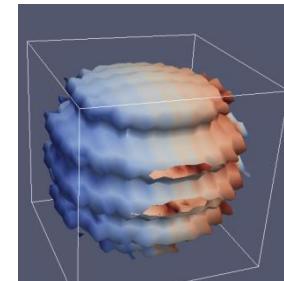
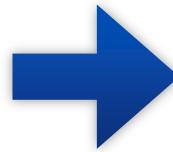
# Co-Processing/InSitu with ParaView Catalyst



```
andor:simulation ./run.sh --parameter1=1 --parameter2=1.5 --parameter3=123 --output=e1/file1.dat  
andor:simulation ./run.sh --parameter1=2 --parameter2=2.5 --parameter3=123 --output=e1/file2.dat  
andor:simulation ./run.sh --parameter1=4 --parameter2=1.5 --parameter3=123 --output=e1/file3.dat  
andor:simulation ...  
andor:simulation ./run.sh --parameter1=1 --parameter2=1.5 --parameter3=123 --output=e2/file1.dat  
andor:simulation ./run.sh --parameter1=2 --parameter2=2.5 --parameter3=123 --output=e2/file2.dat  
andor:simulation ./run.sh --parameter1=2 --parameter2=2.5 --parameter3=123 --output=e2/file3.dat  
andor:simulation ...  
andor:simulation
```



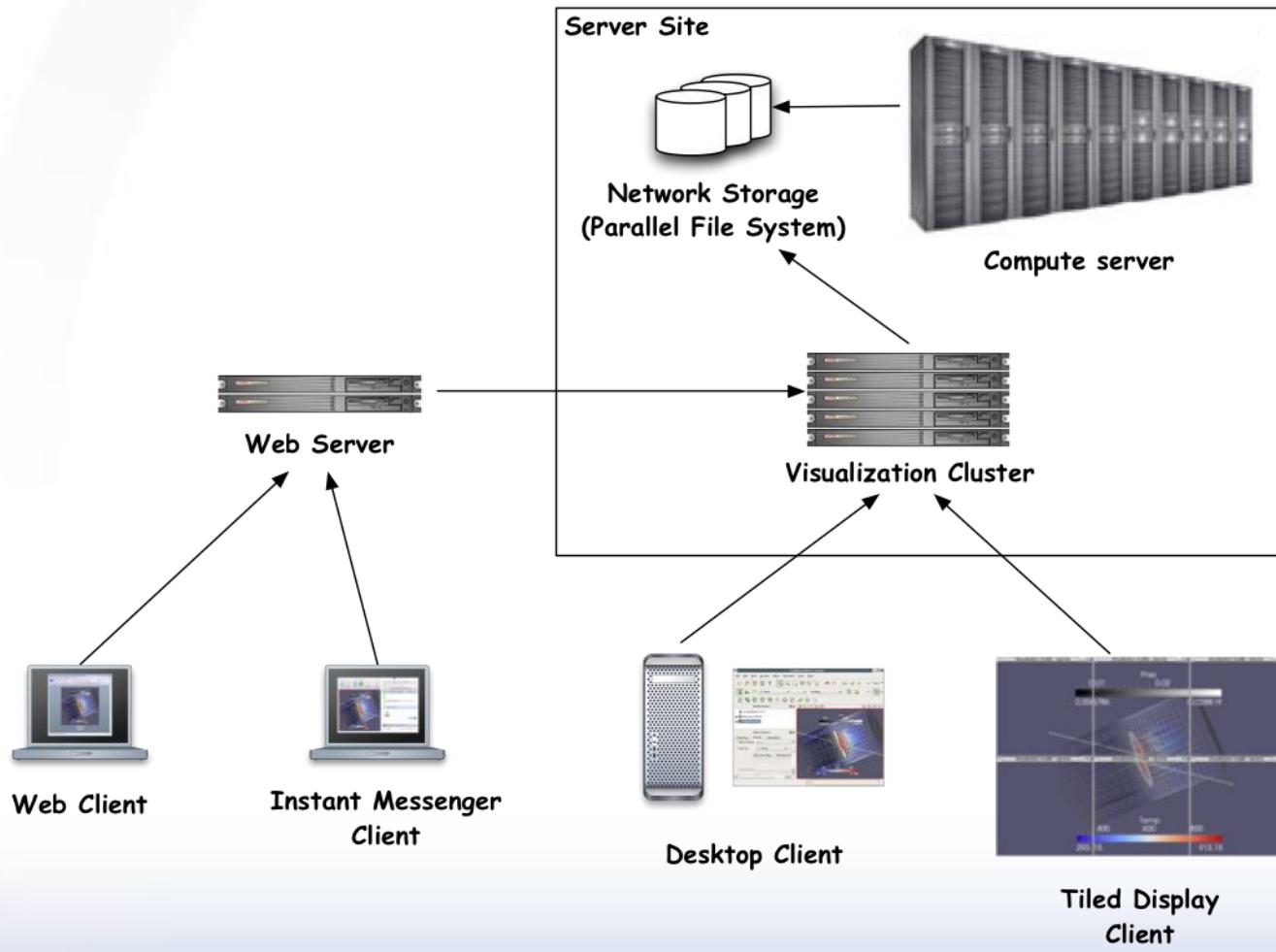
```
1 try: paraview.simple  
2 except: from paraview.simple import *  
3  
4 for idx in range(1, 100):  
5     reader = Read(FileName='e%d.dat' % idx)  
6  
7     DataRepresentation3 = Show()  
8  
9     Contour2 = Contour()  
10    Contour2.ContourBy = ['POINTS', 'Density']  
11    Contour2.Isosurfaces = [2.5850499793887138]  
12  
13    DataRepresentation4 = Show()  
14  
15    Render()  
16  
17    SaveImage("contour%d.png" % idx)
```



# **SOME OTHER TOOLS...**

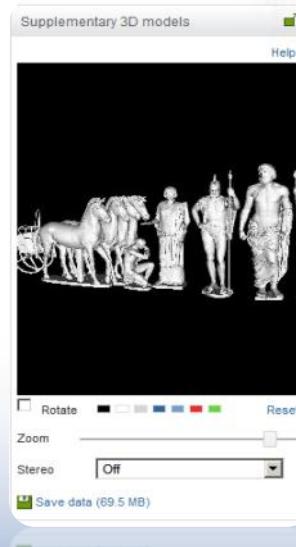
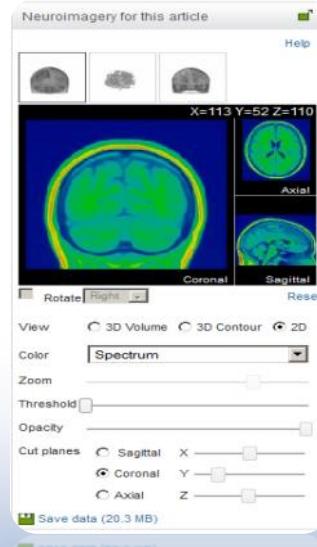
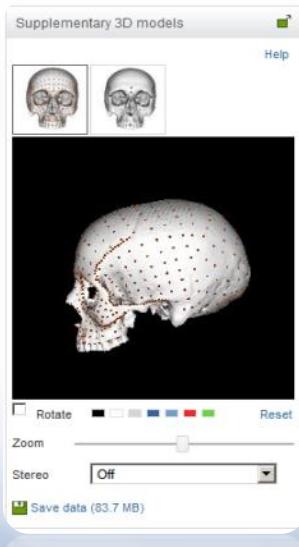


# ParaViewWeb - Collaboration

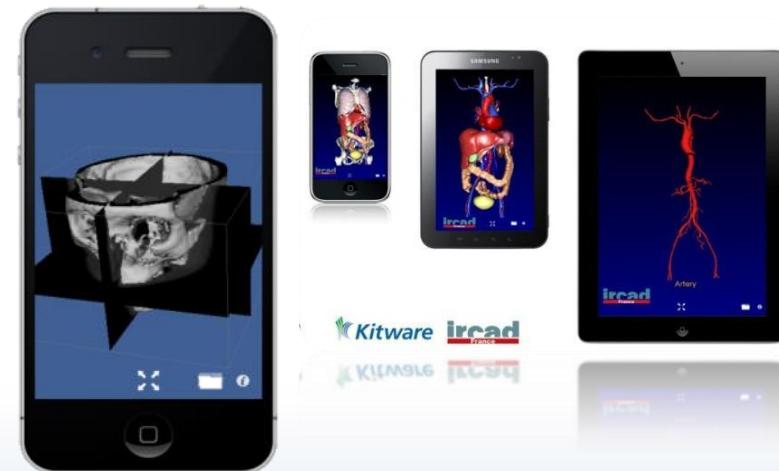
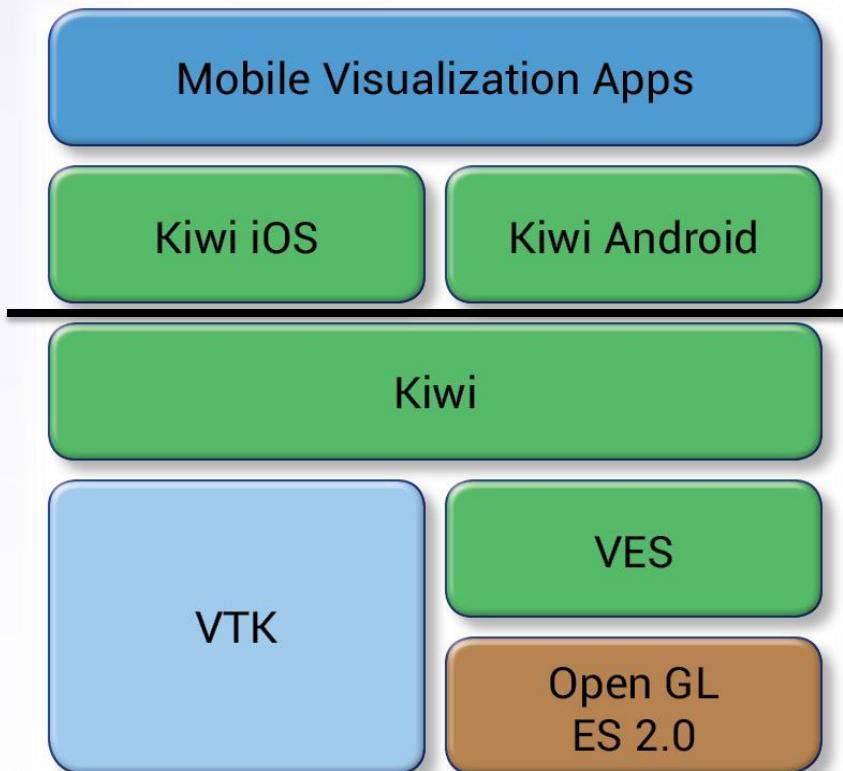


# Web Visualization – vtkWeb/ParaViewWeb

- <http://www.webviz.org>
- No plugin
- Works on all devices and browsers
- Instant visualization (fast loading)
- Fully interactive visualization



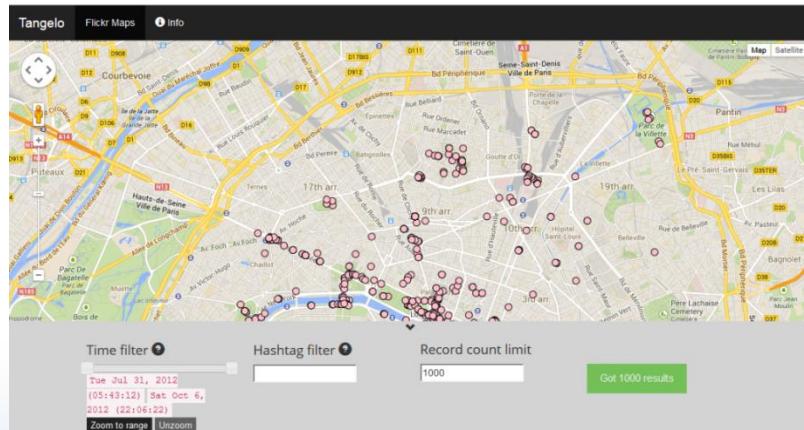
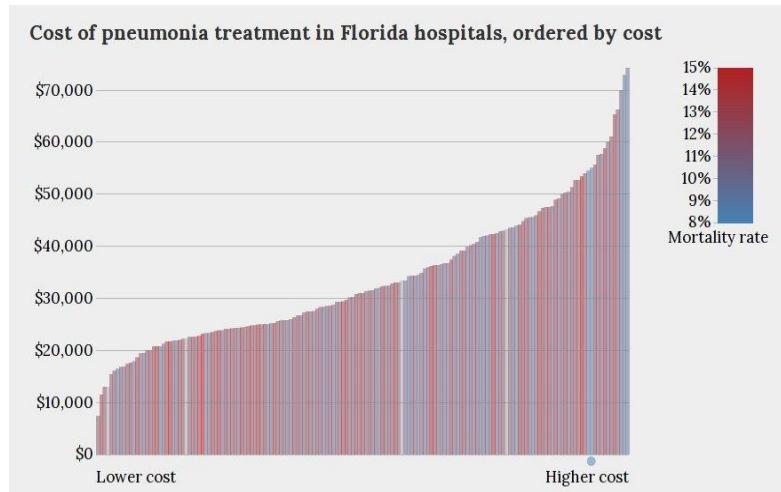
# Mobile Visualization: VES/VTK



# Tangelo

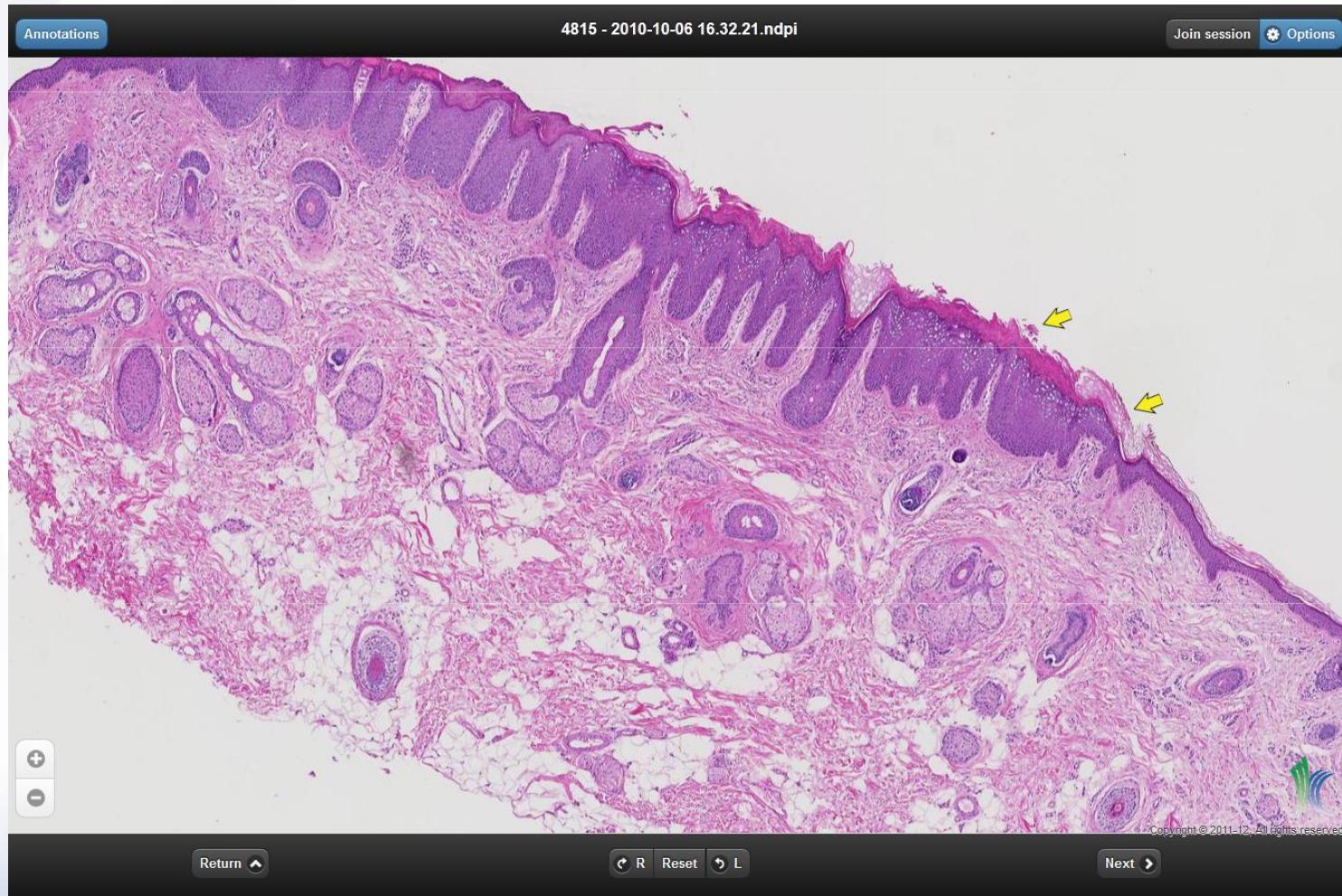


- <http://tangelo.kitware.com>
- Web framework
- HTML5 web architecture
- Packages several other frameworks too
  - Bootstrap, D3, Vega, MongoDB
- Facilitates development & deployment of web apps



# Digital Pathology

- <https://slide-atlas.org/>



# VTKWeb and Open Chemistry

Lumo -0.007

JPNPQ.

GLJLLE

Lumo -0.009

InChIKey

SMILES

Formula

Molecular Mass

Energy

Calculation

Basis

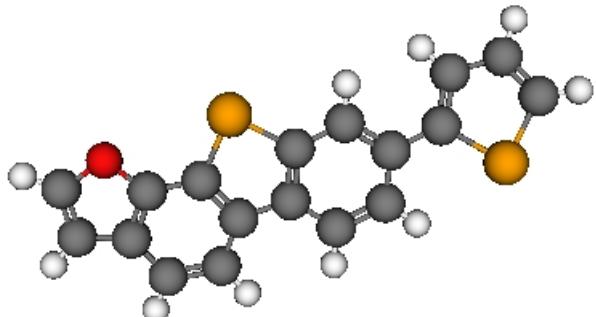
Download

Mass 400.196

Energy Homo -0.112

Mass 400.196

Energy Homo -0.113



IZRSVHMKZWKSSV-UHFFFAOYSA-N

c1cc2ccc3c4ccc(cc4[se]c3c2o1)-c1ccc[se]1

C<sub>18</sub>H<sub>10</sub>OSe<sub>2</sub>

400.196

Homo -0.112

Lumo -0.002

Gap 0.11

Total -5543.7168293588

Theory BP86

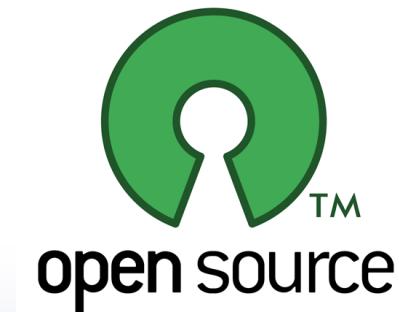
STO-6G

<http://data.openchemistry.org/>



# Benefits of Open-Source

- Extended support
  - The Visualization Toolkit: ~\$100M
- Active maintenance
  - Community-supported
- Access to expertise
- Reduce costs
  - Development
  - Maintenance
  - Evolution
  - No licensing fee





# Thank You!

Jérôme Velut  
[jerome.velut@kitware.com](mailto:jerome.velut@kitware.com)

Kitware Europe  
26 rue Louis Guérin  
69100 Villeurbanne  
France

