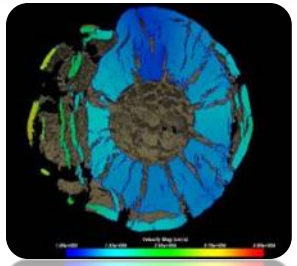
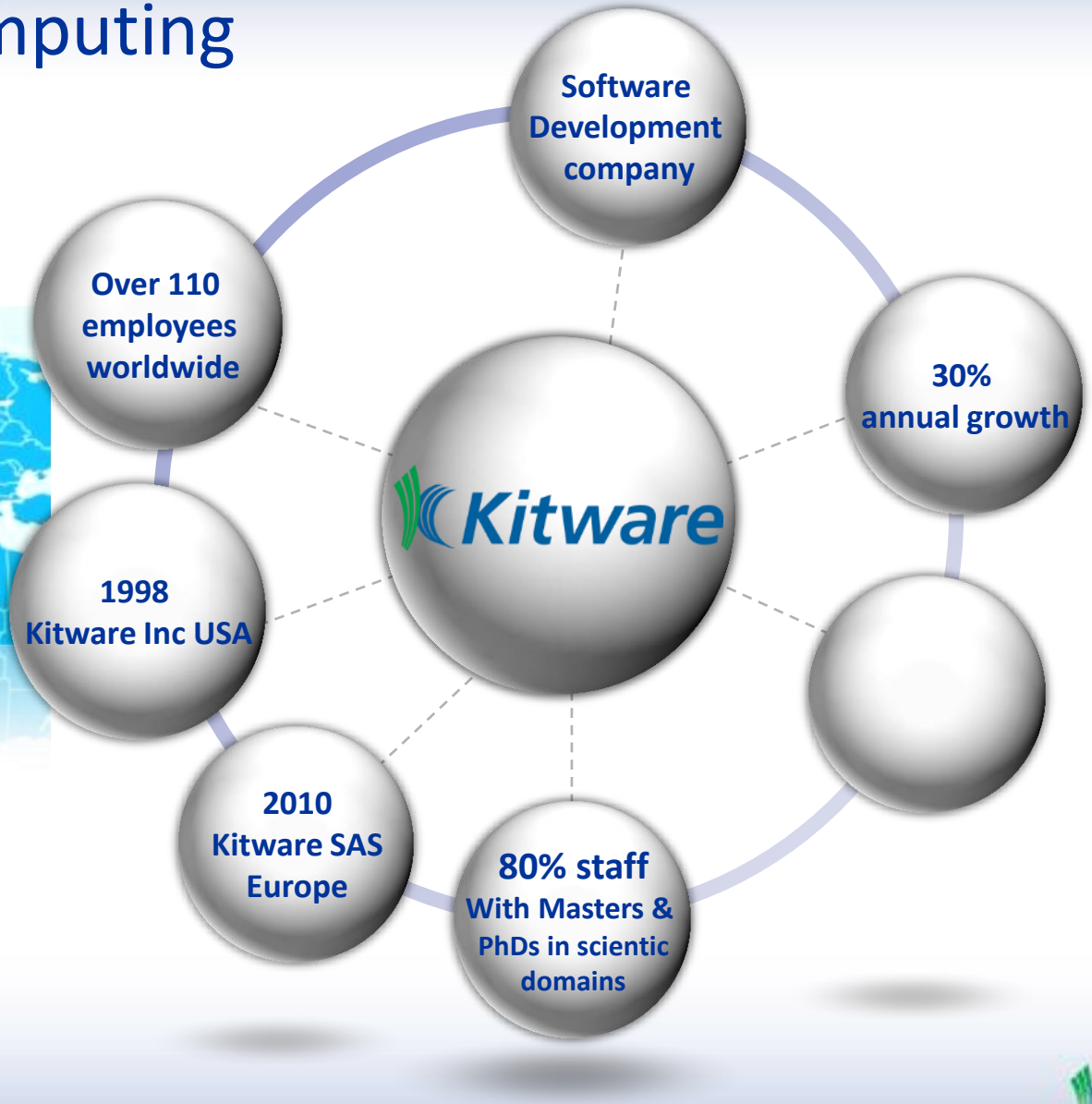
Post-processing and Visualization with Open-Source Tools



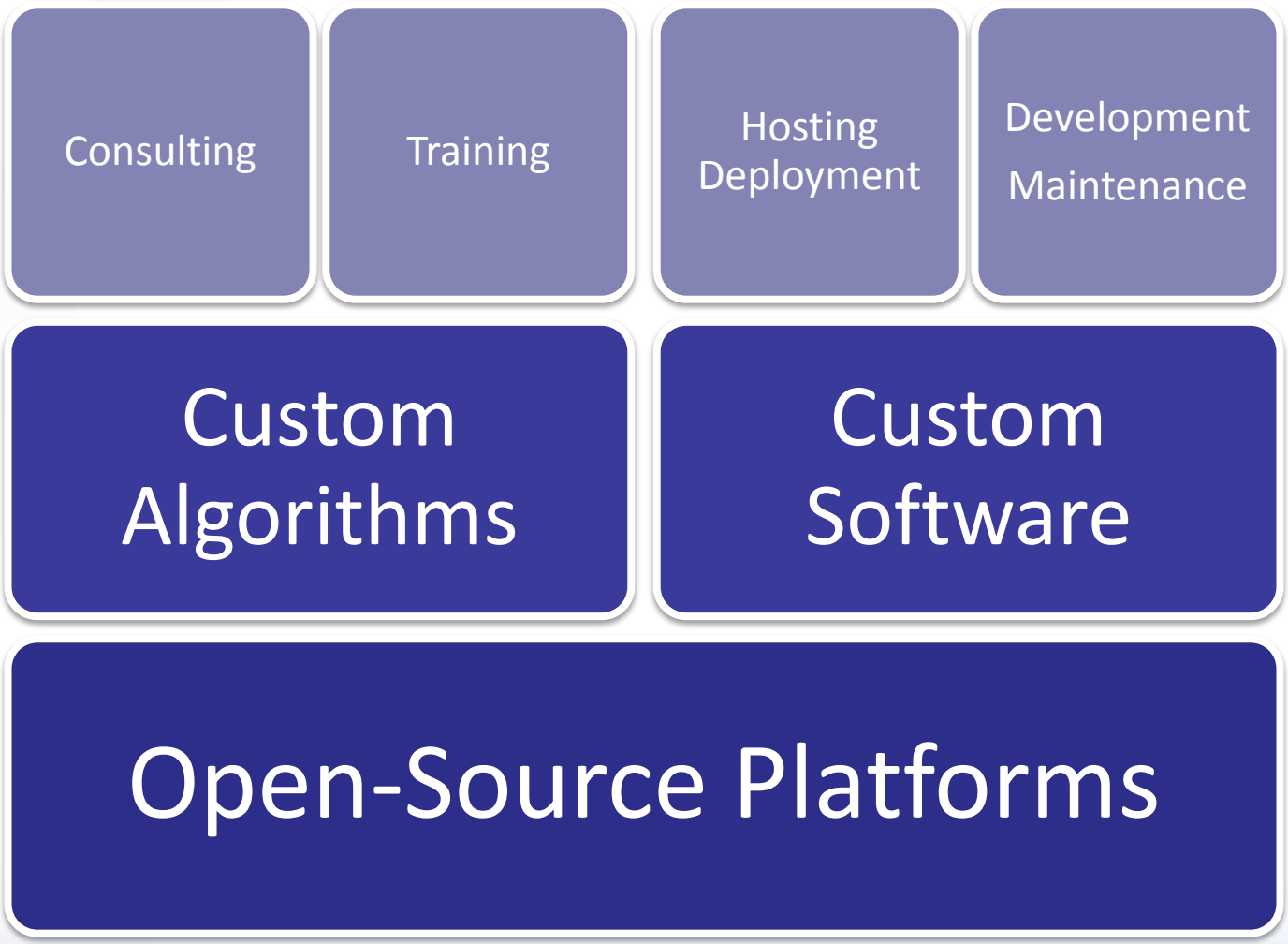
Journée Scientifique Centre Image

April 9, 2015 - Julien Jomier

Kitware - Leader in Open Source Software for Scientific Computing

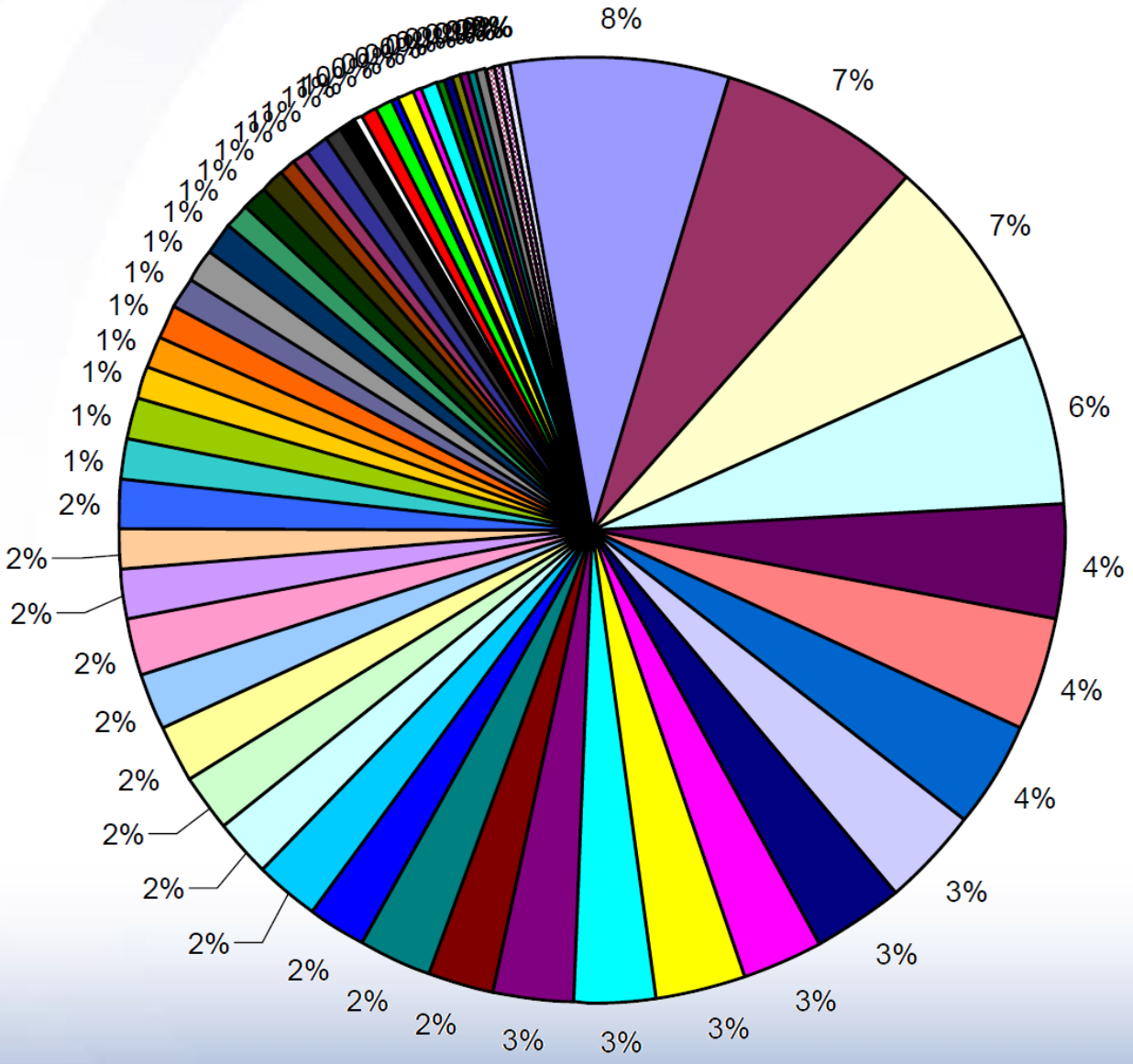


Kitware - Solutions based on Open-Source Software

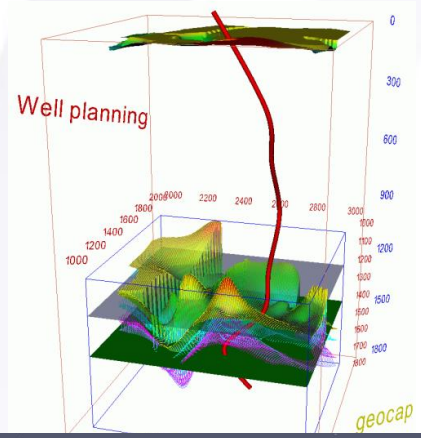


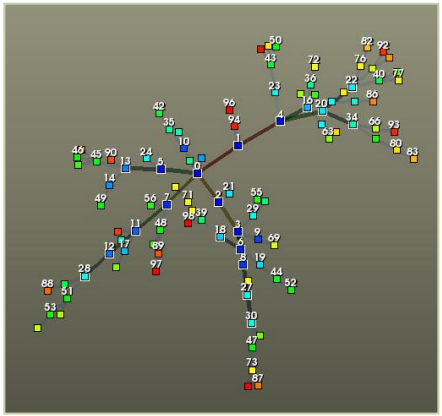
Kitware - Fields of Expertise

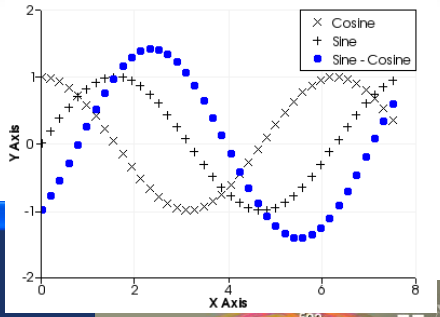
Kitware Open Source Projects

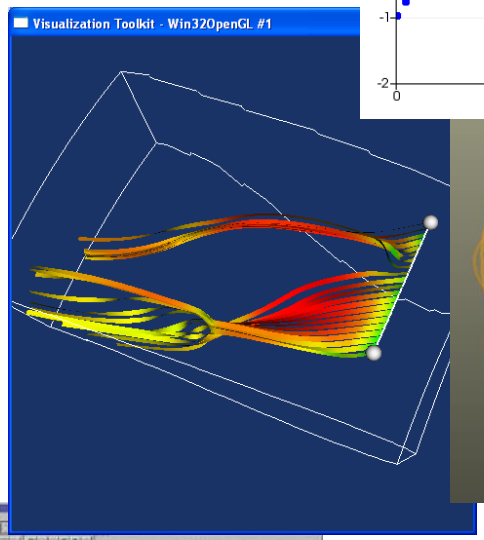


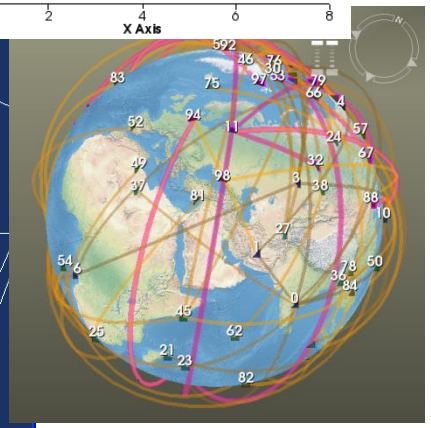
Challenges of Visualization

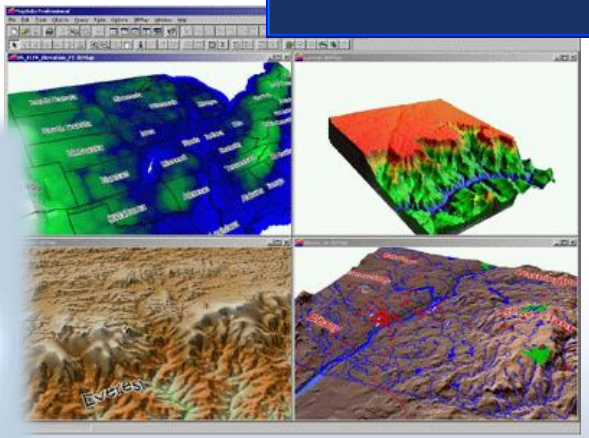


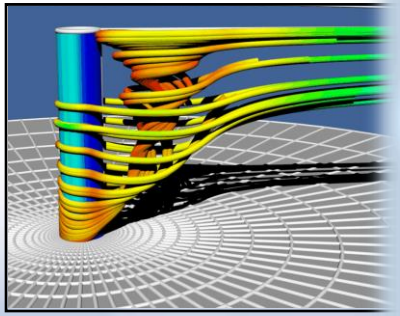


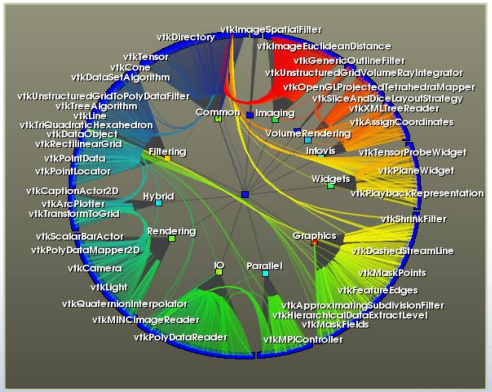












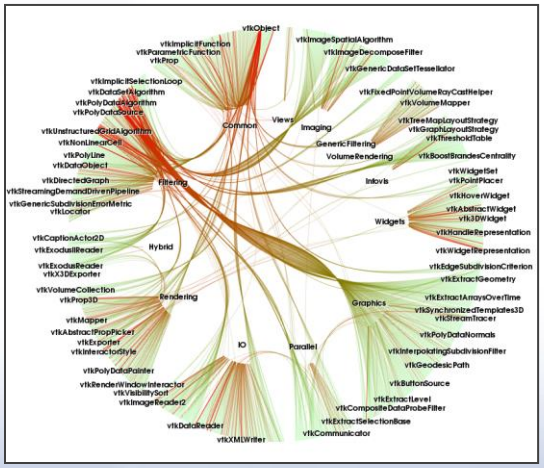
Challenges of Visualization

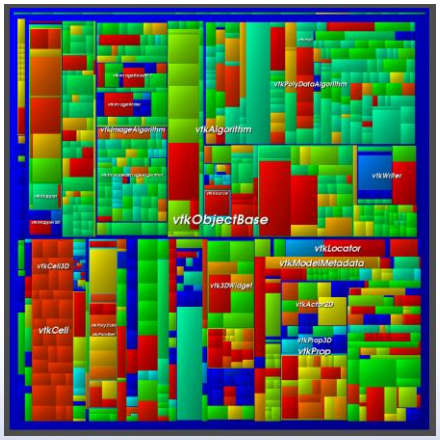
- Heterogeneous data
- Large/many/big data
- Distributed data
- Computing resources
- Domain-specific
- User-specific
- Heterogeneous devices
- Uncertainties

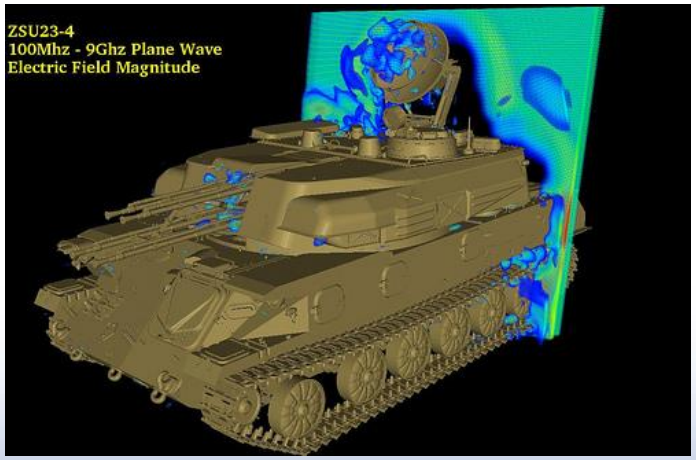
TOOLS CAN HELP OVERCOME THESE CHALLENGES

The Visualization Toolkit (VTK)

- www.vtk.org
- Started in 1993 at GE
- Visualization Library
  - Written in C++ (+8.2 million LOC) – BSD License
  - Automatic binding for Java, Python, C#, TCL
  - Portable by design: Linux, Windows, Mac OSX, Solaris…
- Very active community: 4000+ users on the mailing list

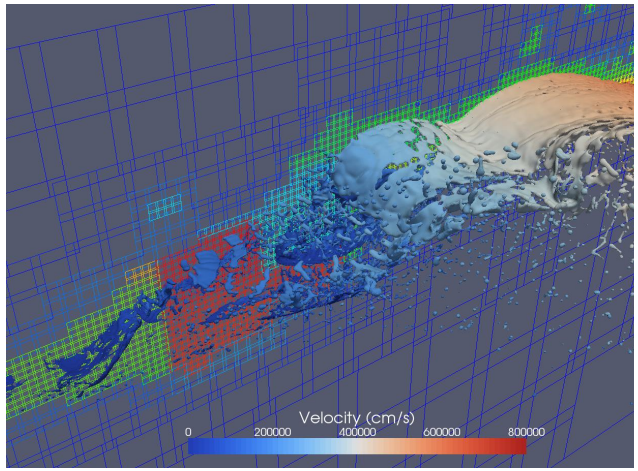


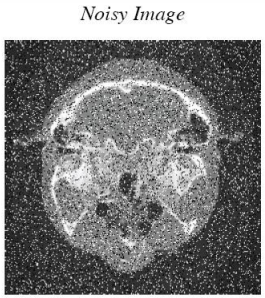


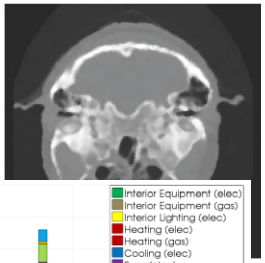


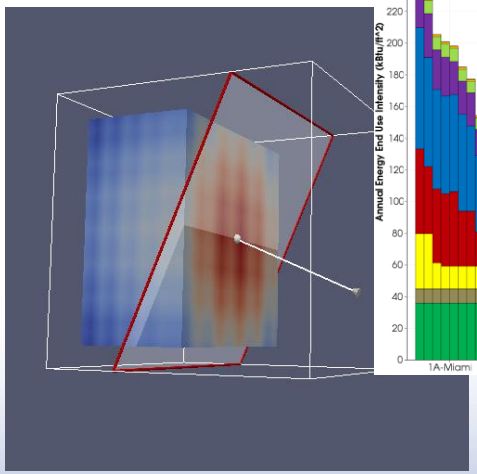
What can VTK do for me?

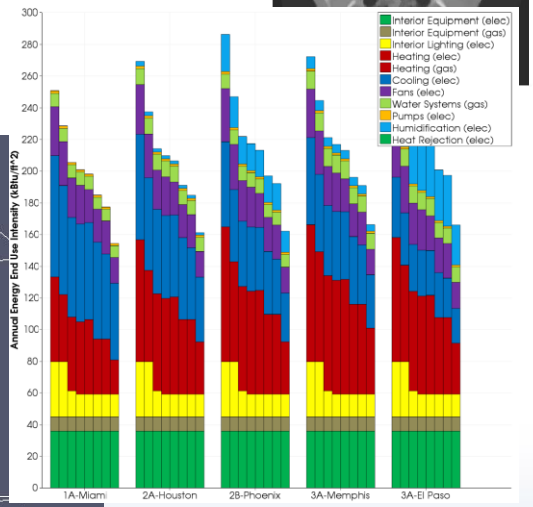
- **Scientific Visualization**
  - 2D to 4D data processing and (volume) rendering
  - Image processing
- **Information Visualization**
  - Charting/plotting
- **Application support**
  - GUI support, Widgets
- **Toolkit**
  - Meant to be integrated











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)

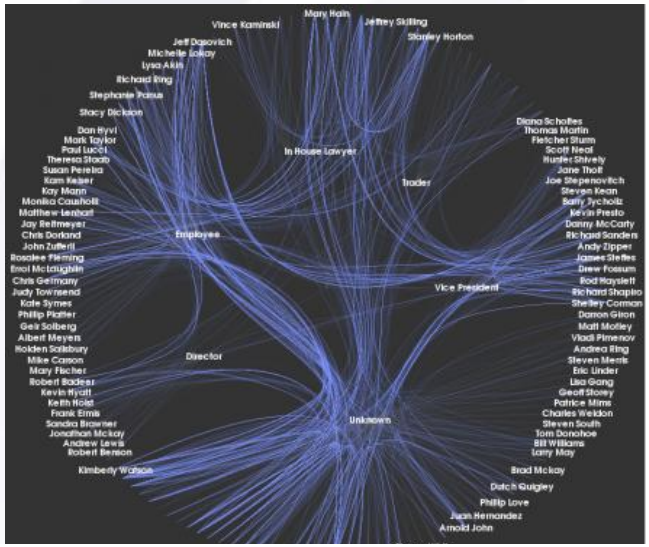
VTK Main Components

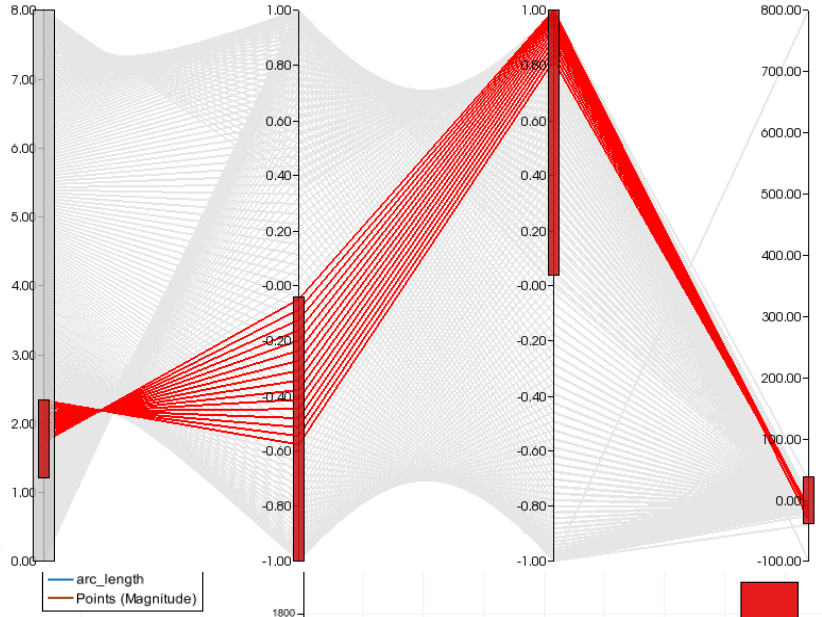
- **Data structures**
  - How VTK stores/provides access to arbitrary data
- **Algorithms/filtering pipeline**
  - Manipulate data
  - Readers, sources, filters, writers
- **Rendering classes**
  - Display that data on the screen
  - Mappers, actors, lights, cameras, renderers
- **Interaction classes**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

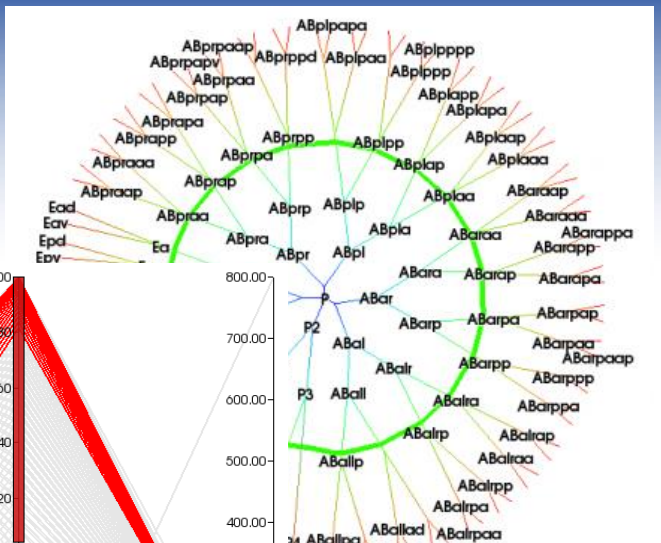
VTK Main Components

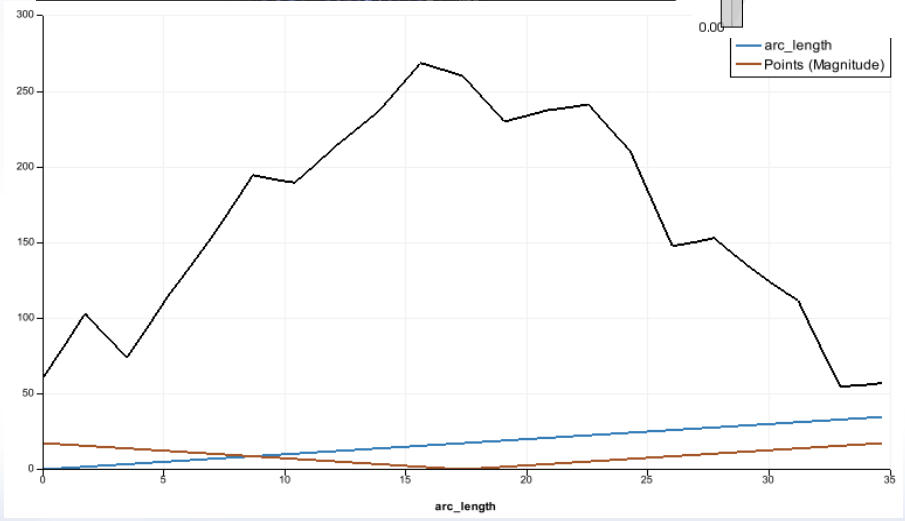
- **140 Readers**
  - STL, EnSight, TecPlot, BMP, JPEG…
- **150 Filters**
  - Contour, Subdivision, Delaunay, Elevation…
  - Statistics, Parallel, Geometry, FlowPaths, Extraction…
- **40 Widgets**
  - Distance, Angle, Plane, Seed, Checkerboard
- **Application domains**
  - Geo visualization
  - Chemistry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

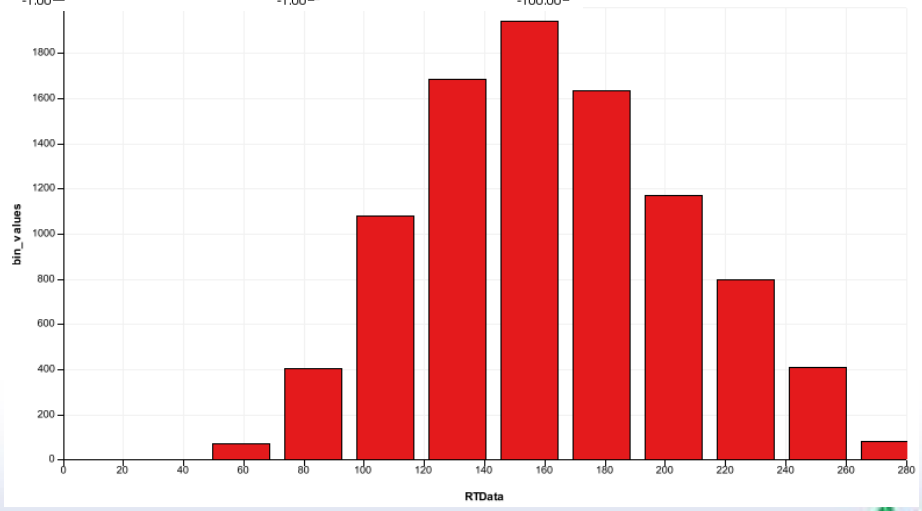
2D: Graphs and Charts



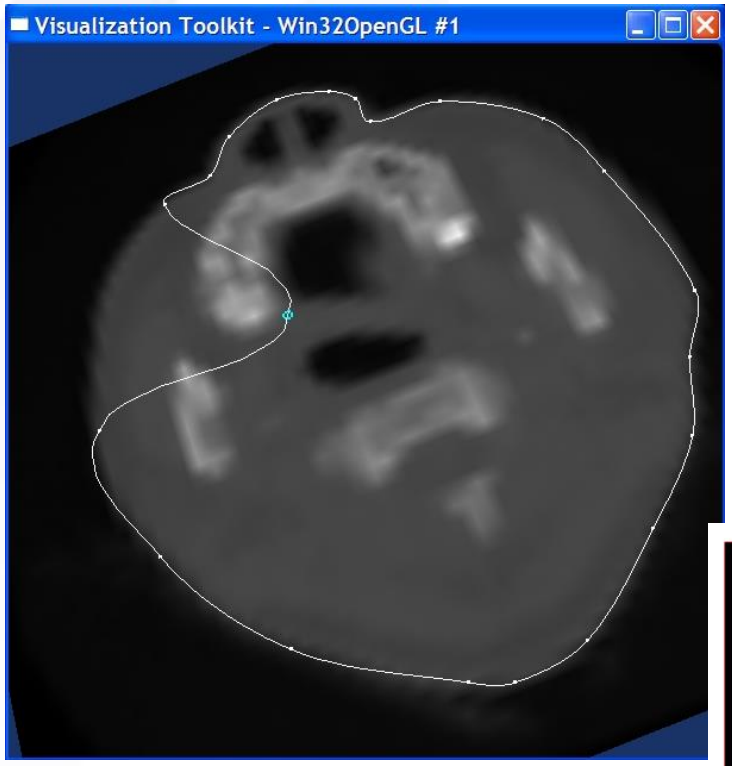


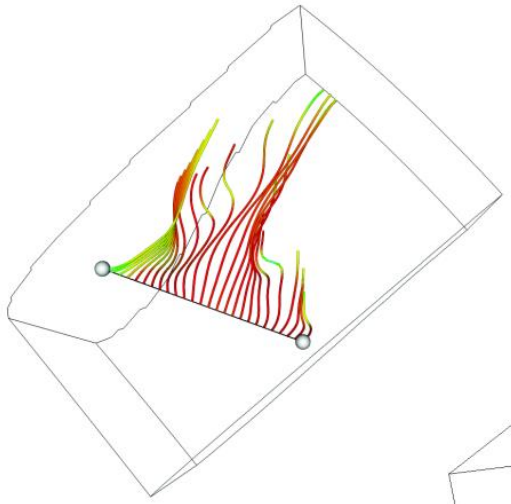


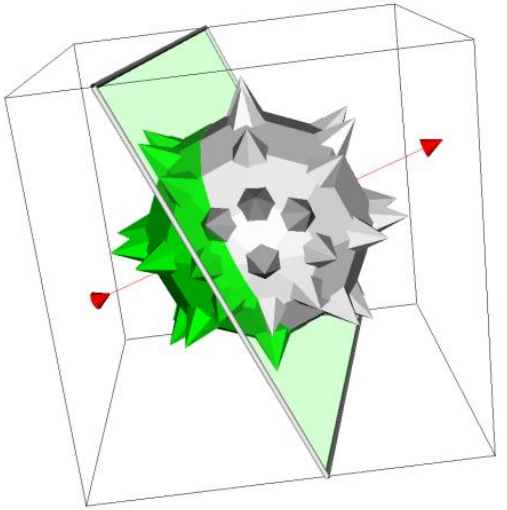


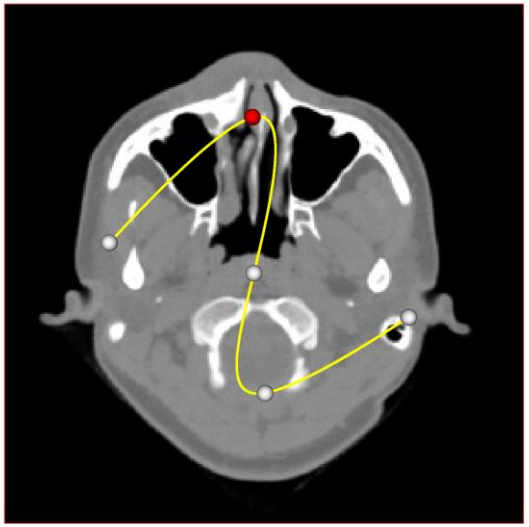


Interactive Widgets









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)
```

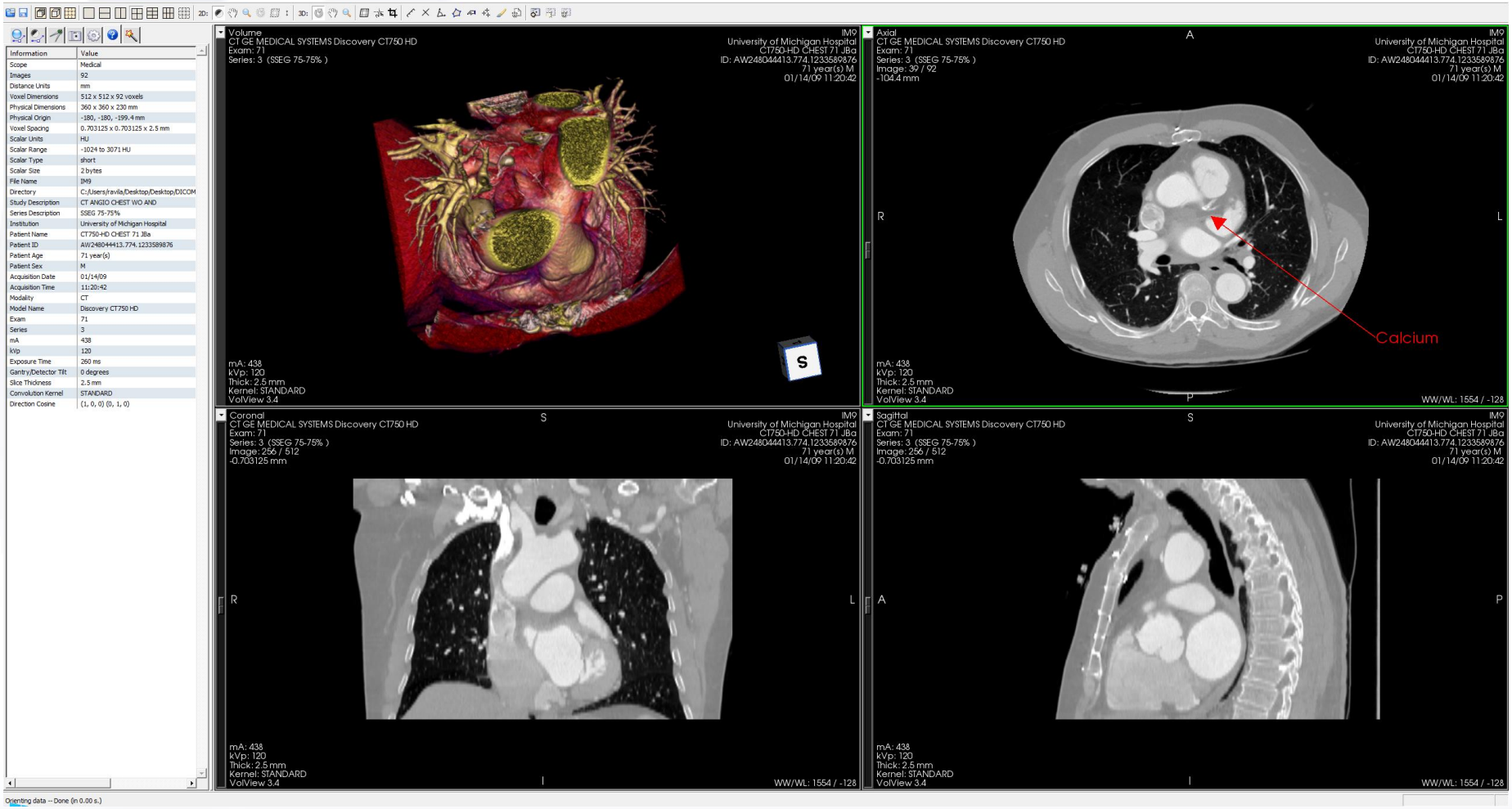
```
# Create a rendering window and renderer
ren = vtk.vtkRenderer()
renWin = vtk.vtkRenderWindow()
renWin.AddRenderer(ren)

# 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()
```

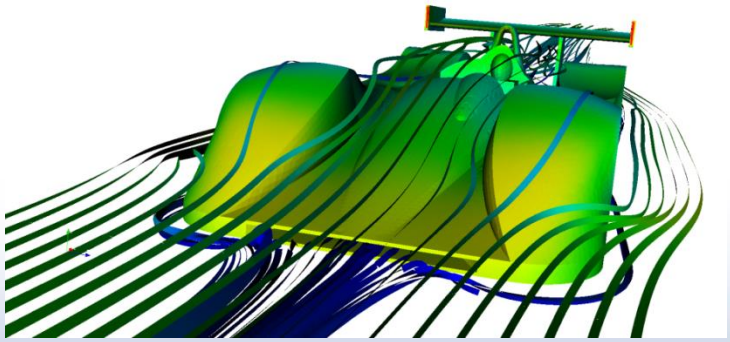
Volume Rendering



LARGE DATA VISUALIZATION

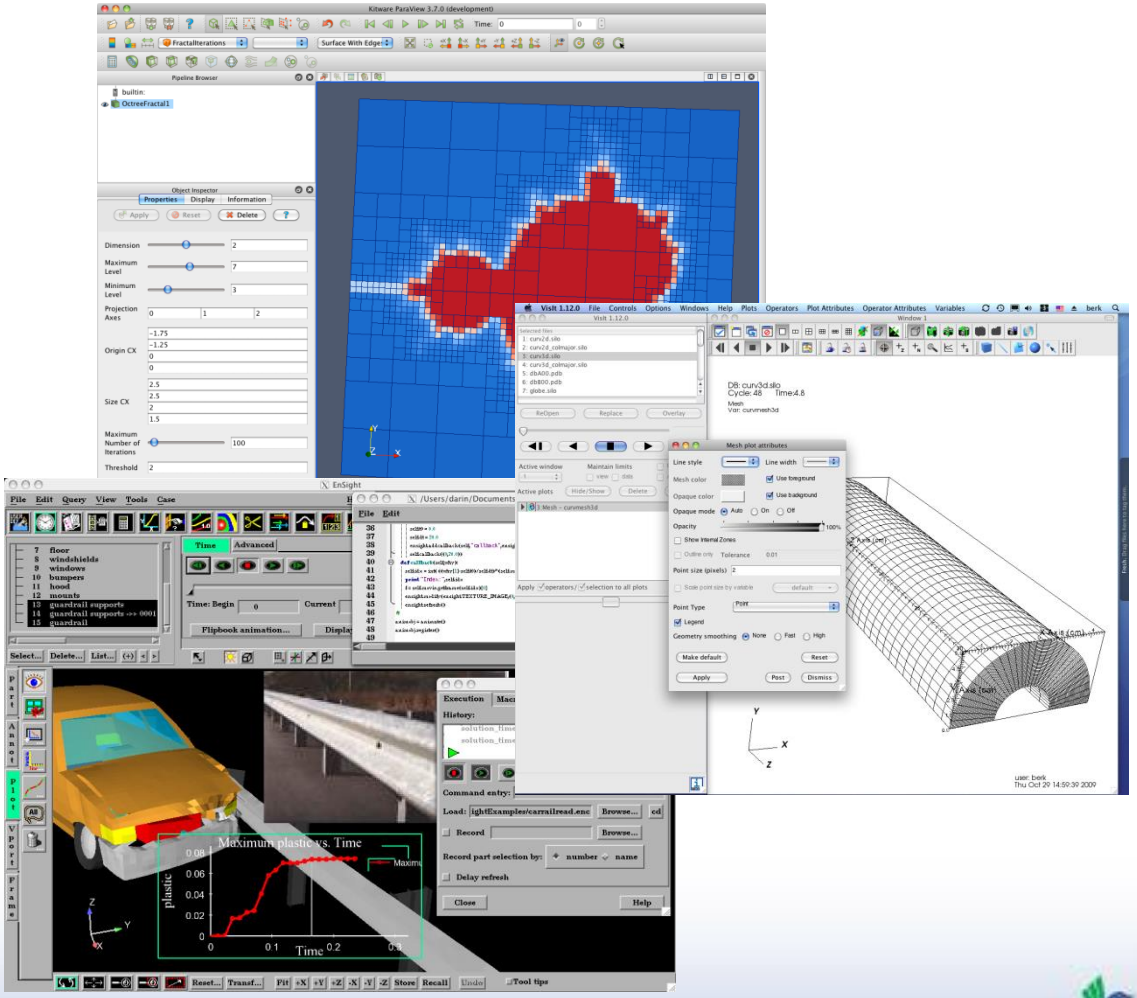
ParaView

- 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



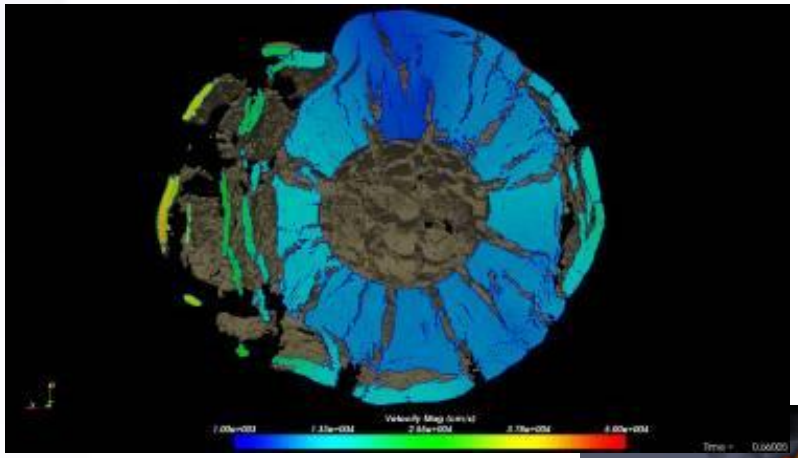
General Purpose Tools

- EnSight
- ParaView
- VisIt
- FieldView
- TecPlot
- …

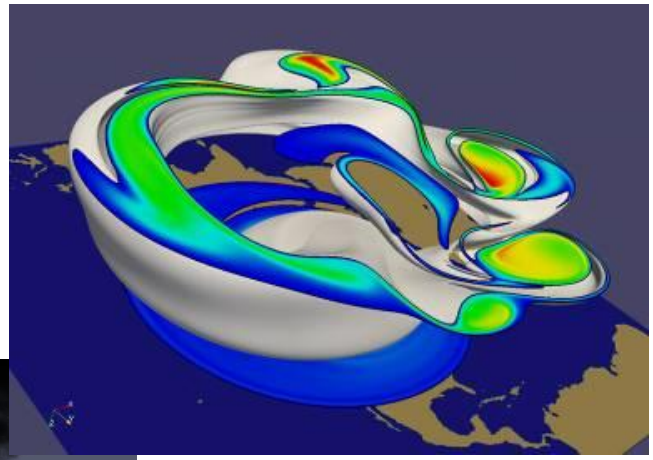


ParaView

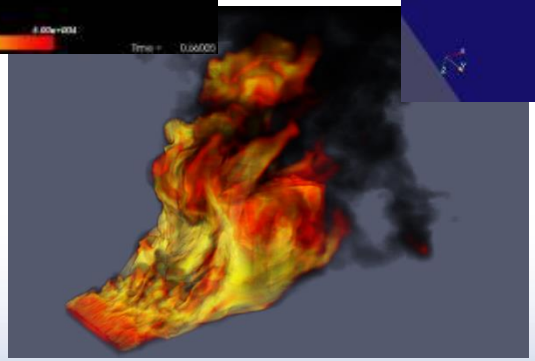
- An application and an architecture to visualize and analyze massive datasets
- A turn-key visualization application



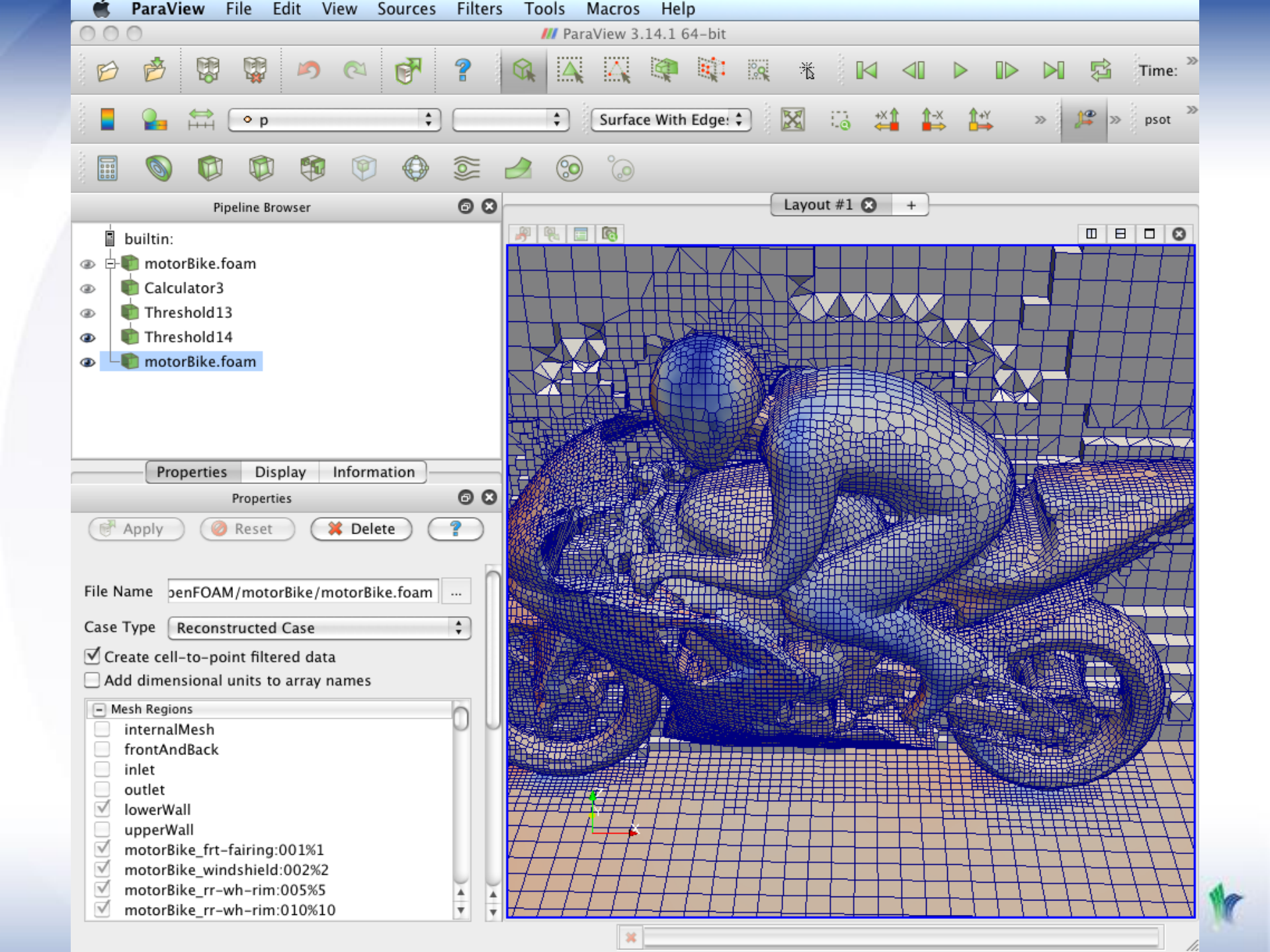
1 billion cell asteroid detonation simulation



½ billion cell weather simulation



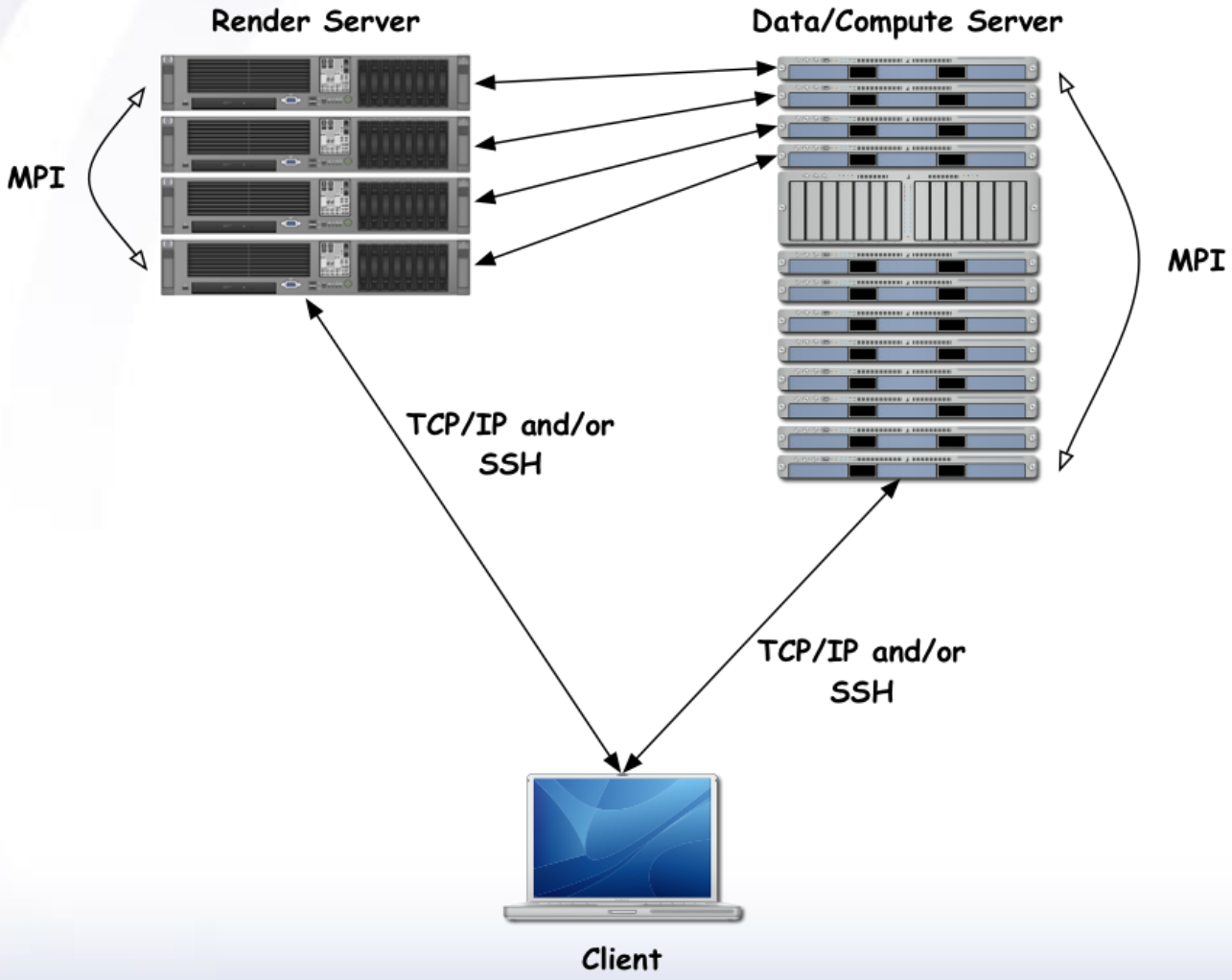
Fire simulation


ParaView File Edit View Sources Filters Tools Macros Help
ParaView 3.14.1 64-bit
Time:
p
Surface With Edge
psot
Pipeline Browser
builtin:
motorBike.foam
Calculator3
Threshold13
Threshold14
motorBike.foam
Properties Display Information
Properties
Apply
Reset
Delete
File Name
penFOAM/motorBike/motorBike.foam
Case Type
Reconstructed Case
Create cell-to-point filtered data
Add dimensional units to array names
Mesh Regions
internalMesh
frontAndBack
inlet
outlet
lowerWall
upperWall
motorBike_frt-fairing:001%1
motorBike_windshield:002%2
motorBike_rr-wh-rim:005%5
motorBike_rr-wh-rim:010%10
Layout #1

ParaView is a Framework

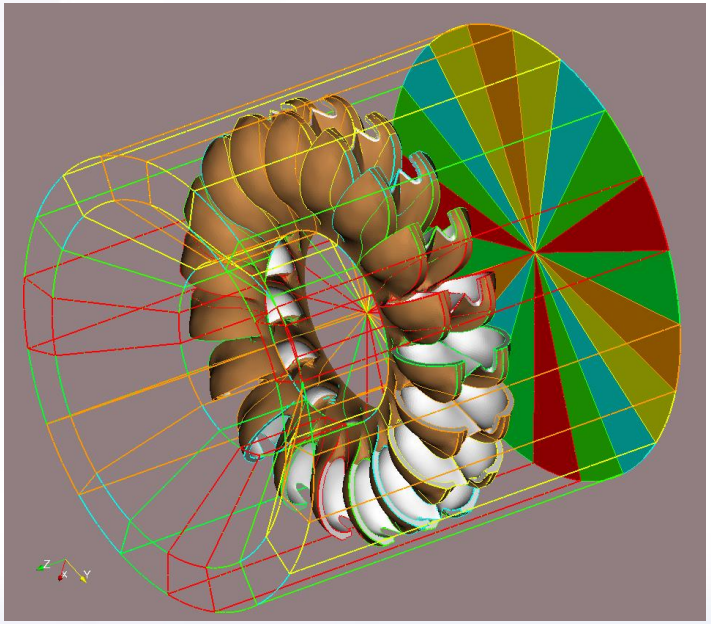
- ParaView **extends VTK** to provide:
  - Client-server computing
  - State management
  - Python modules
  - Application/GUI framework
- ParaView framework can be used to **develop other applications**
- ParaView can be **embedded in other applications** and frameworks

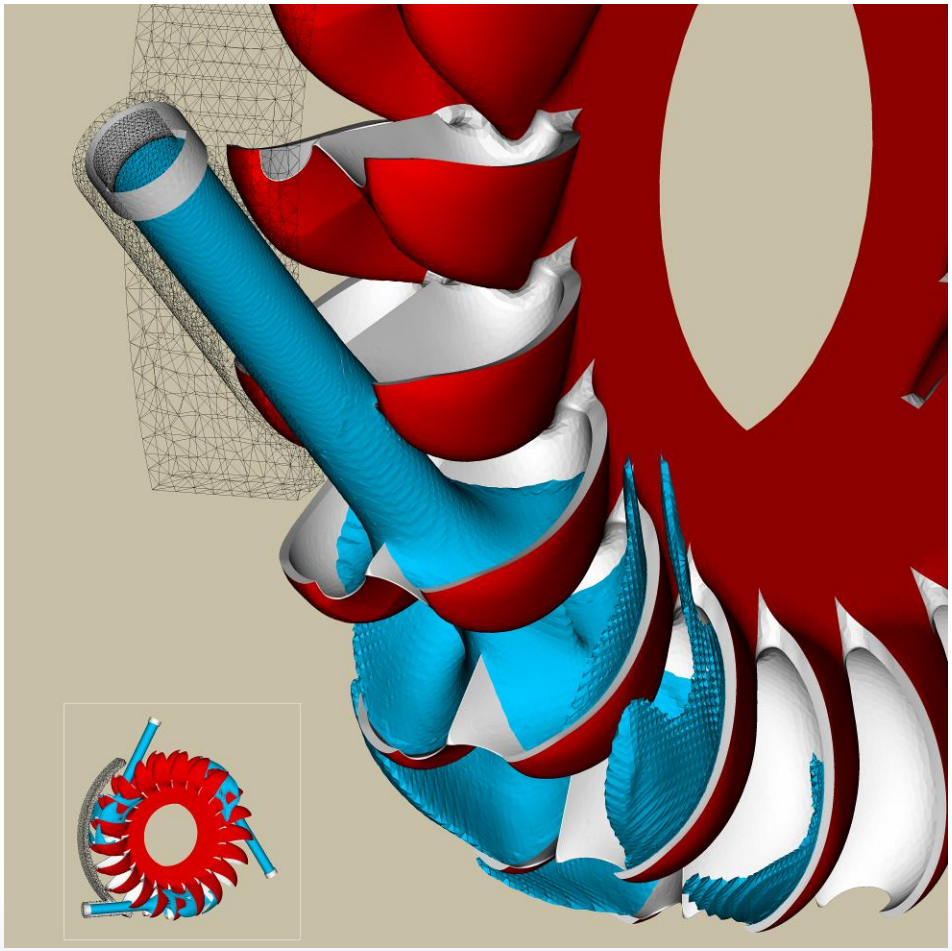
ParaView Architecture



Large Data - Unstructured

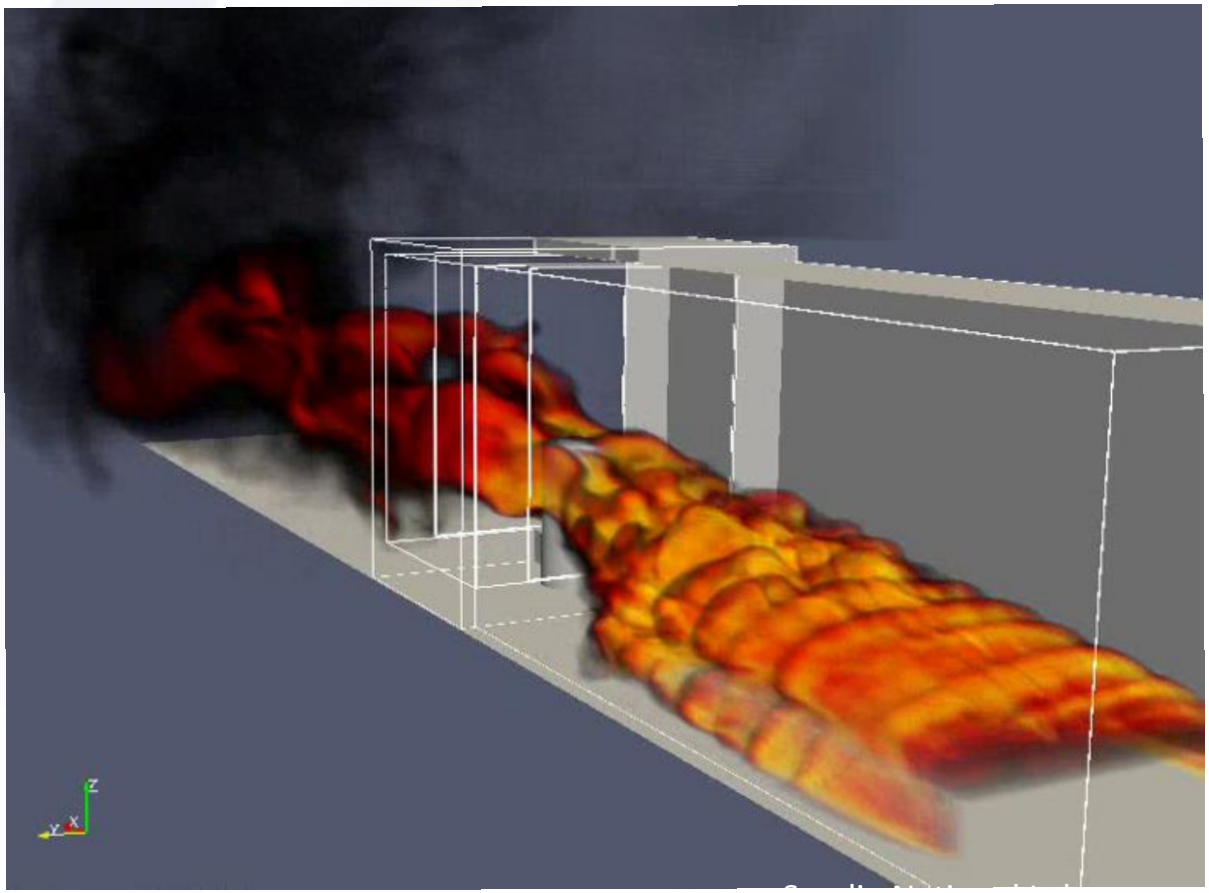
- CFD simulation
- 20-30 million elements
- Load balancing





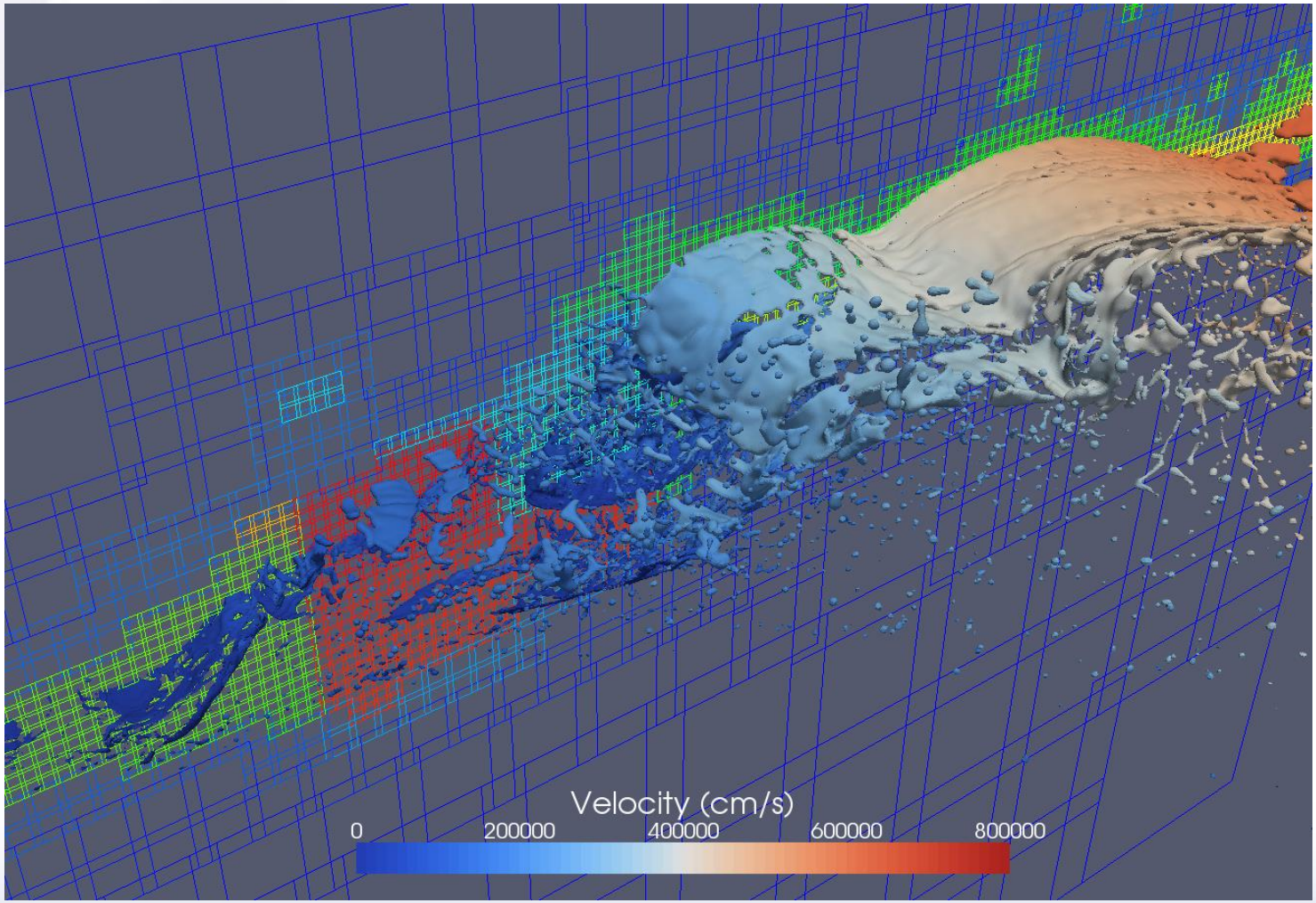
source: Swiss supercomputing center

Large Data - Unstructured

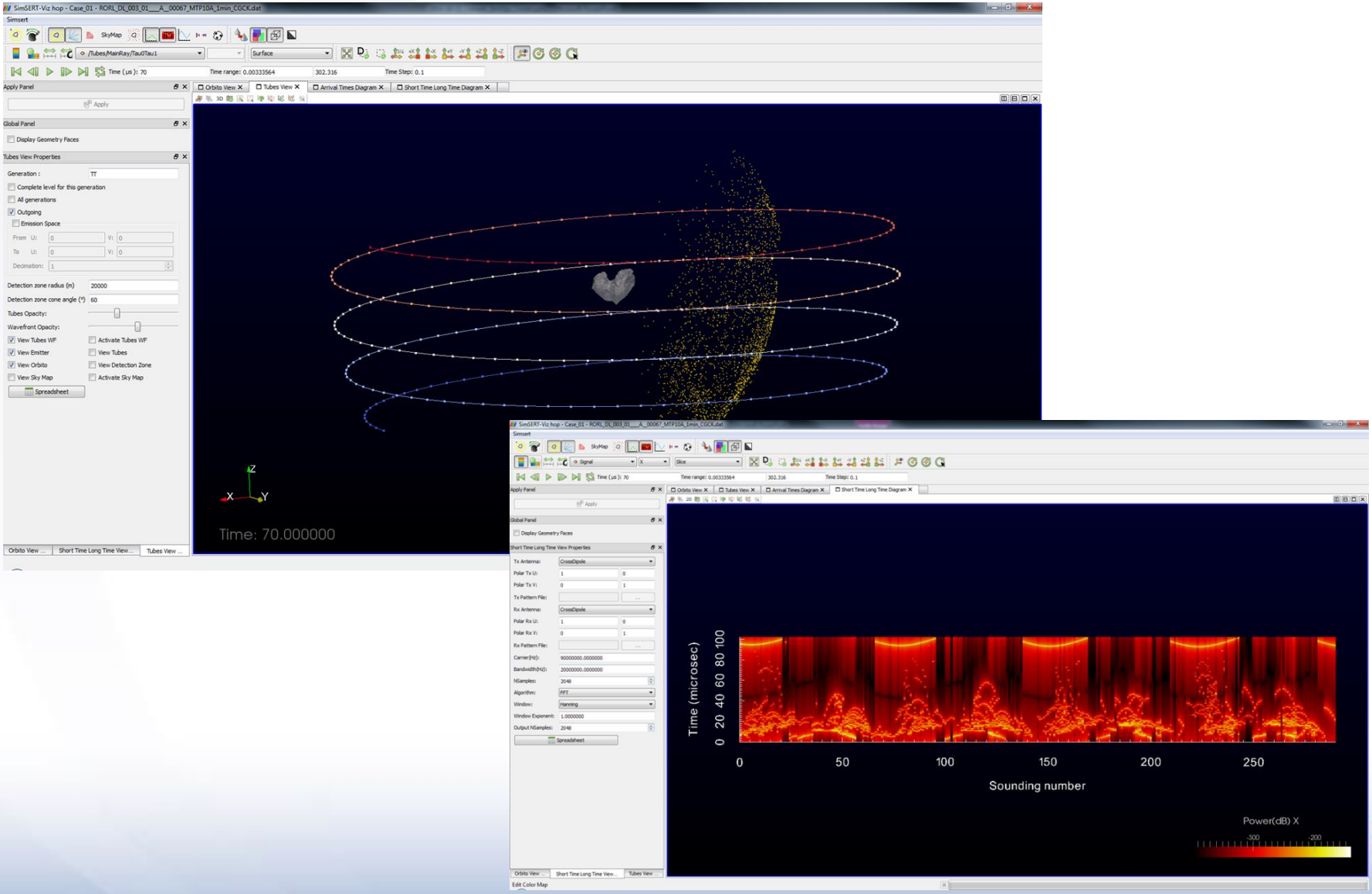


- Fire simulation
- 150 million elements

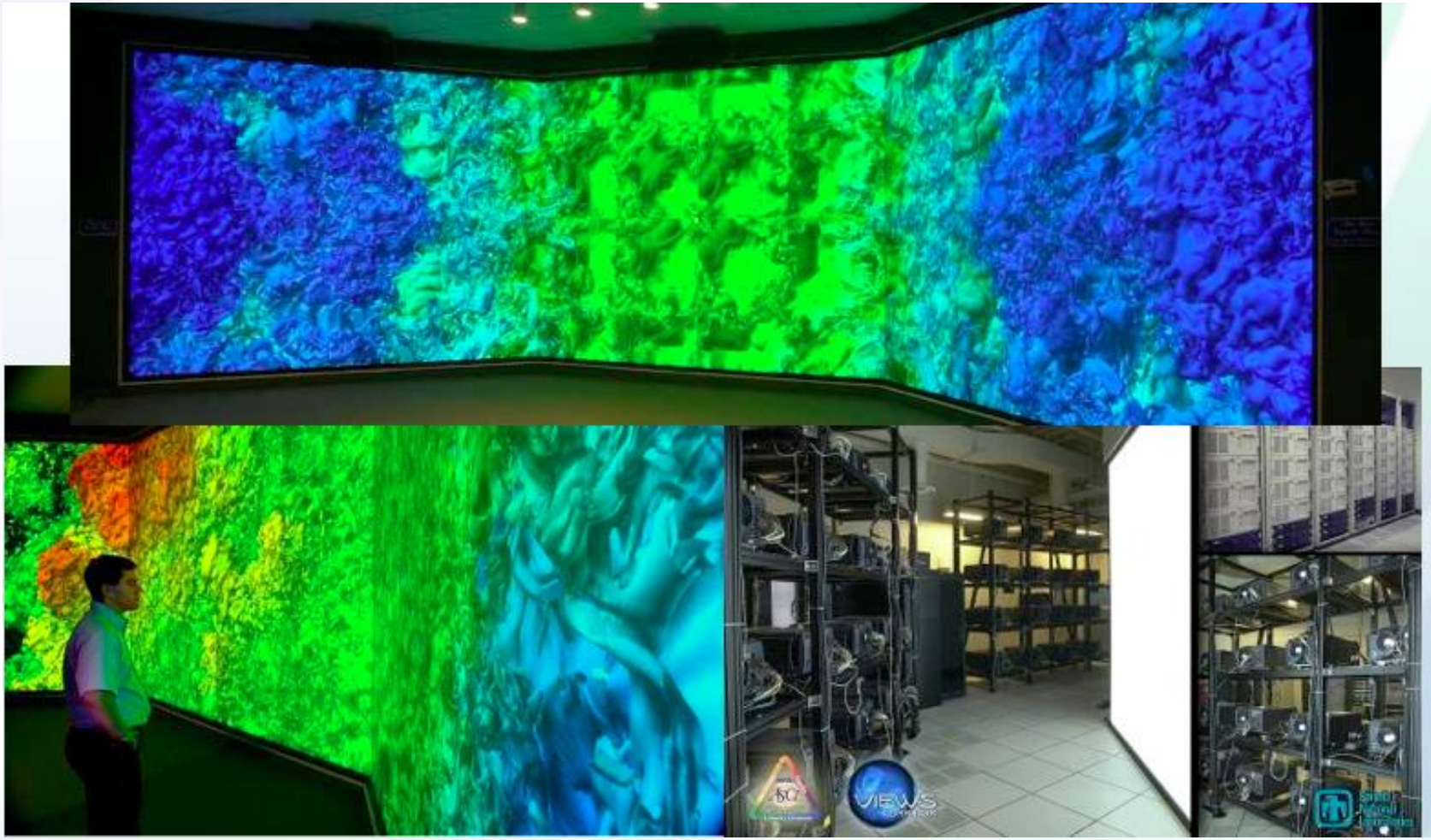
Large-scale AMR



ESA - Rosetta

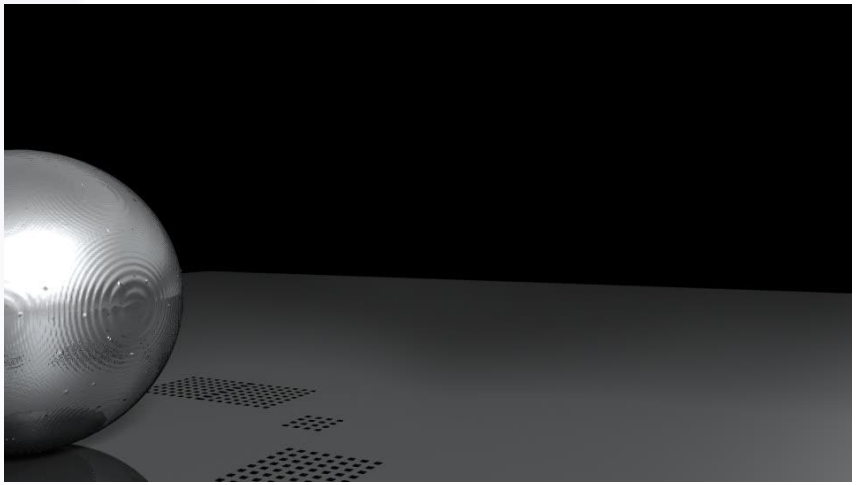

Time: 70.000000
Time (microsec)
0 20 40 60 80 100
0 50 100 150 200 250
Sounding number
Power(dB) X

ParaView in Use: Immersive Visualization

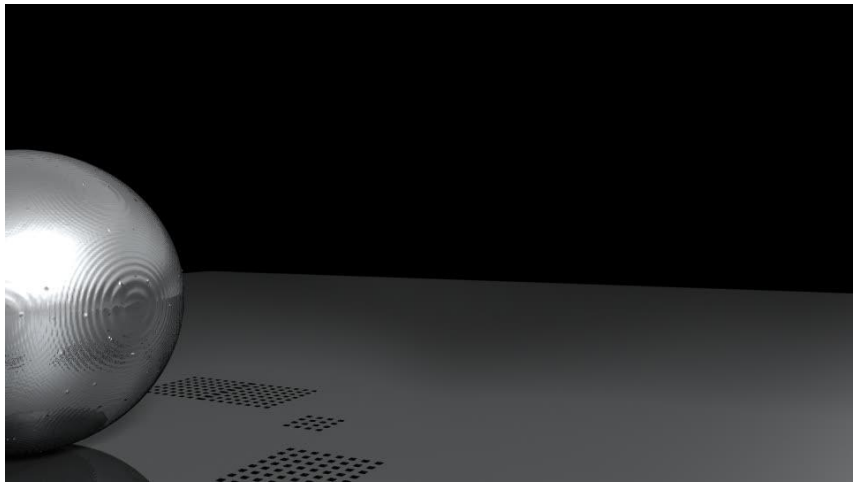


In-Situ : Access to More/Richer Data

Post-Processing
(every 100 time steps)

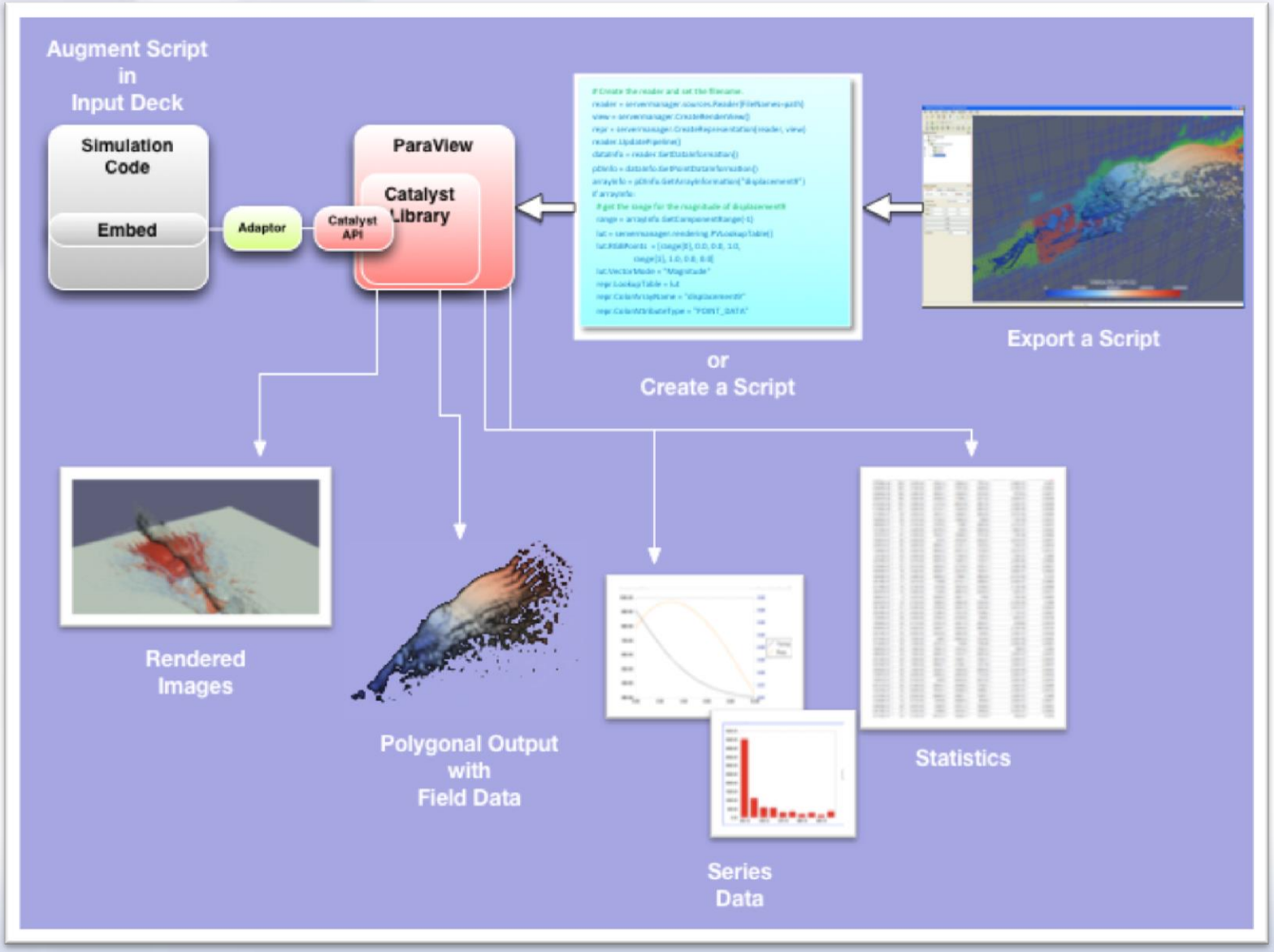


In-situ
(every time step)



Note: Reflections and shadows added in post-processing

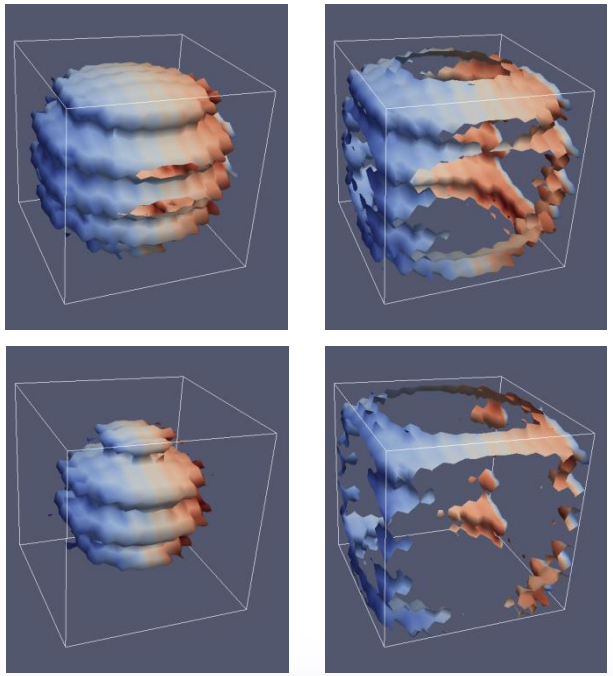
Co-Processing/InSitu with ParaView Catalyst



```
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
```

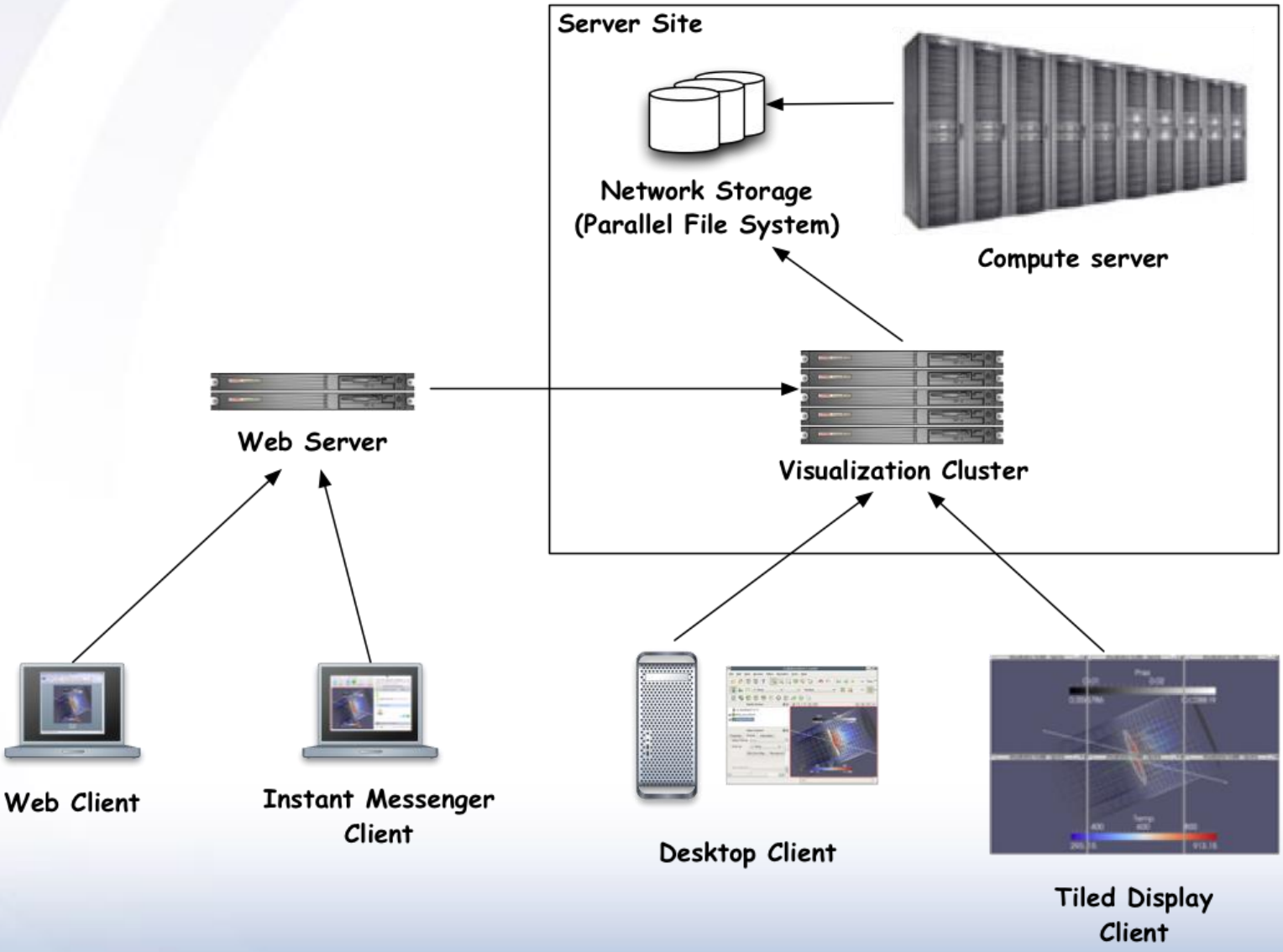
180
180

```
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)
18
```



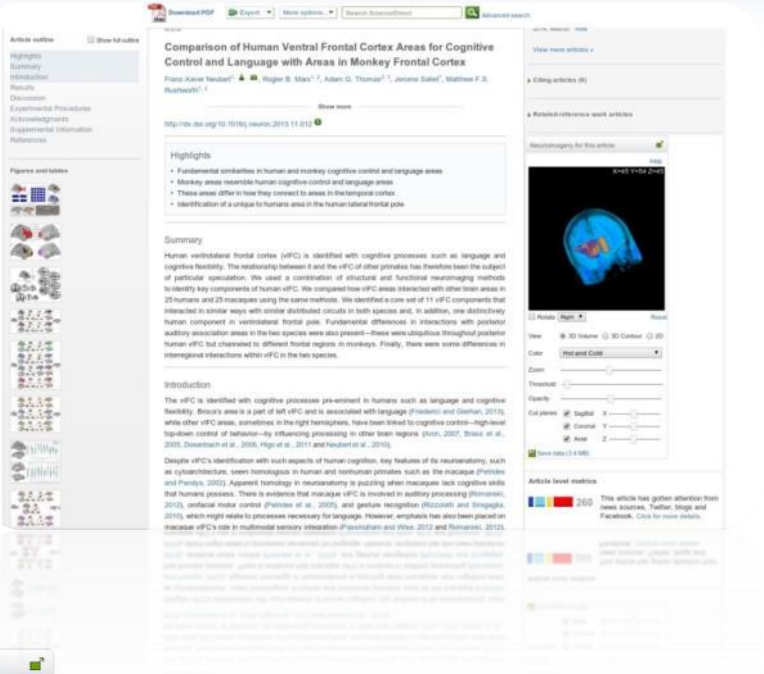
OTHER VISUALIZATION TOOLS

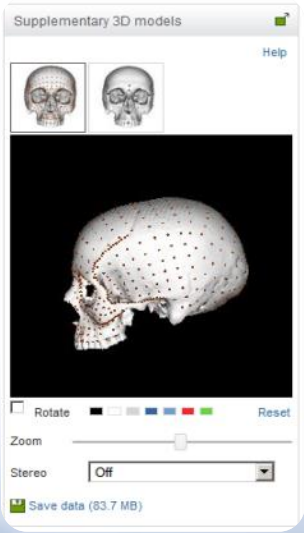
ParaViewWeb - Collaboration

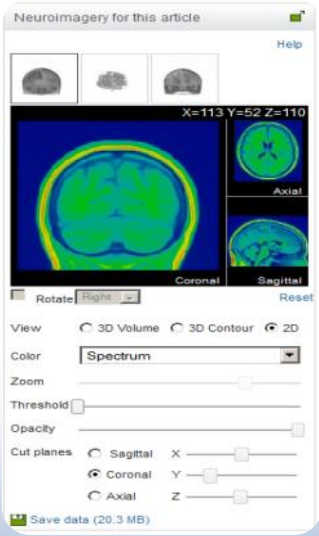


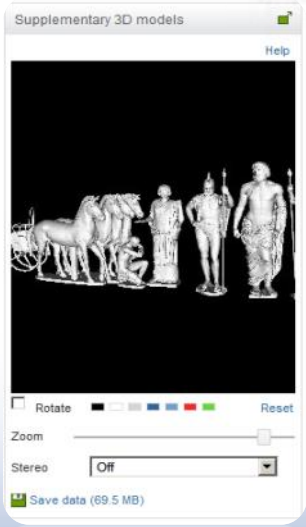
Web Visualization – vtkWeb/ParaViewWeb

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

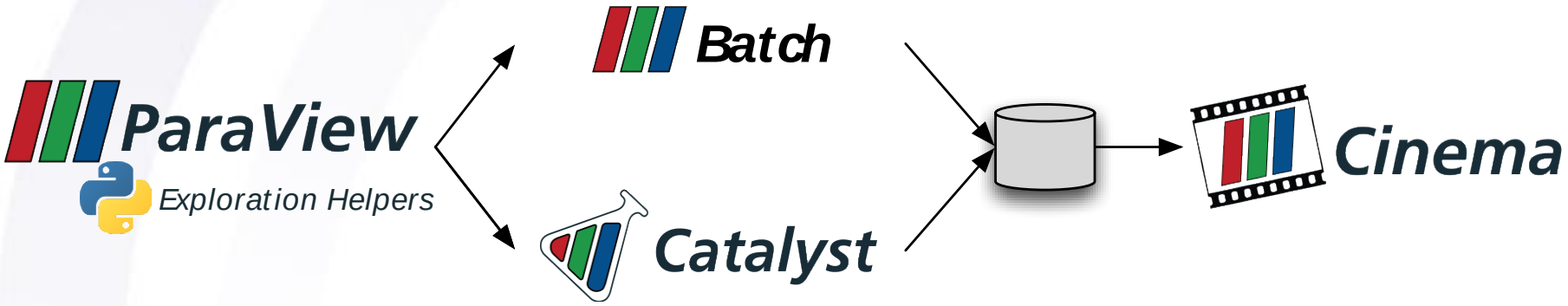


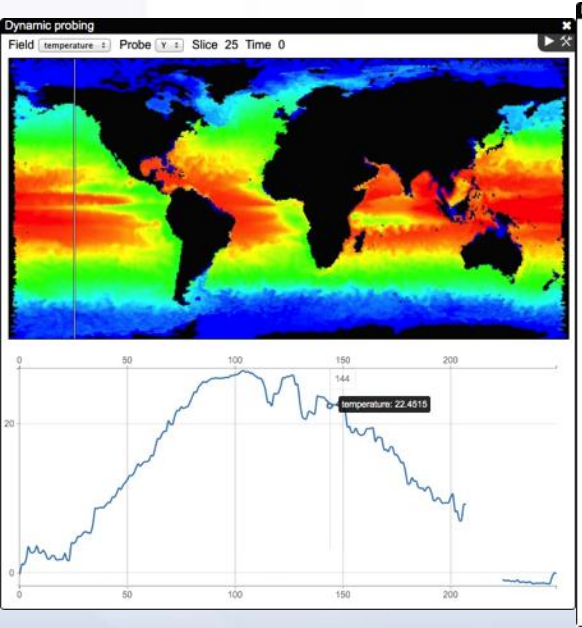


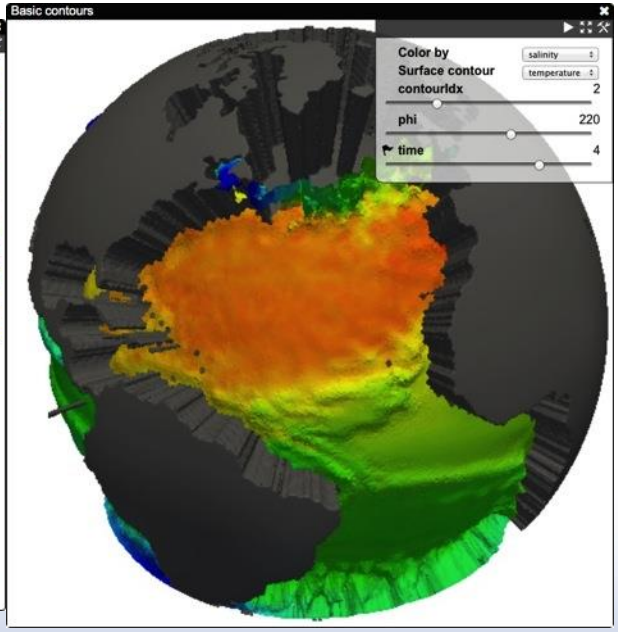


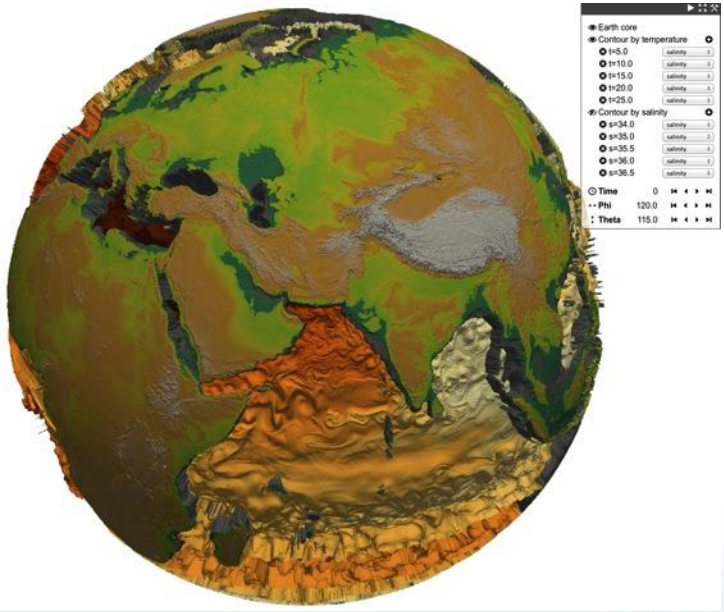


ParaView - Cinema

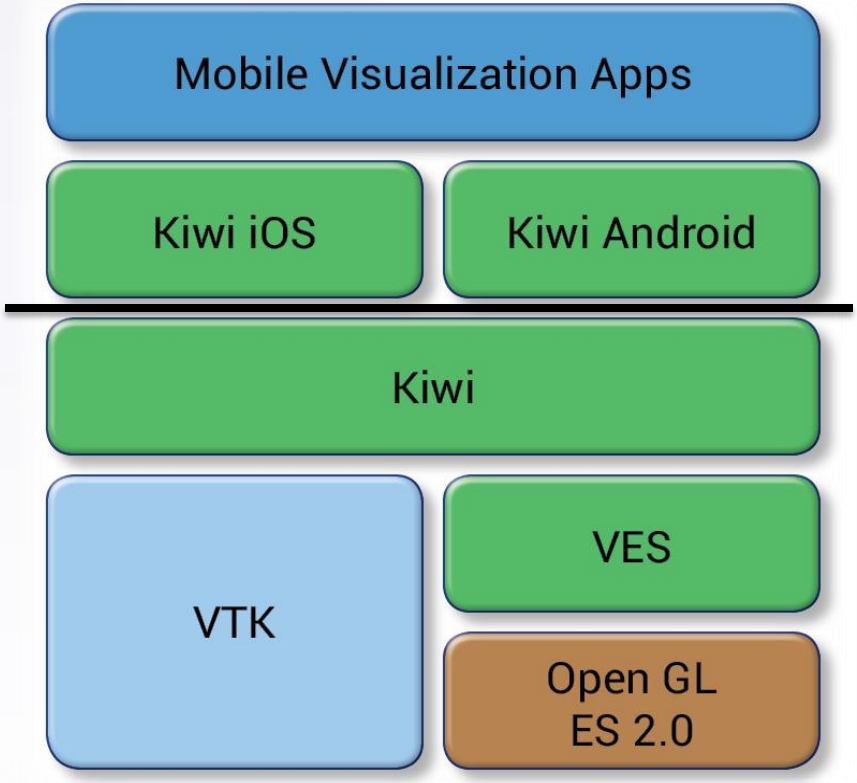






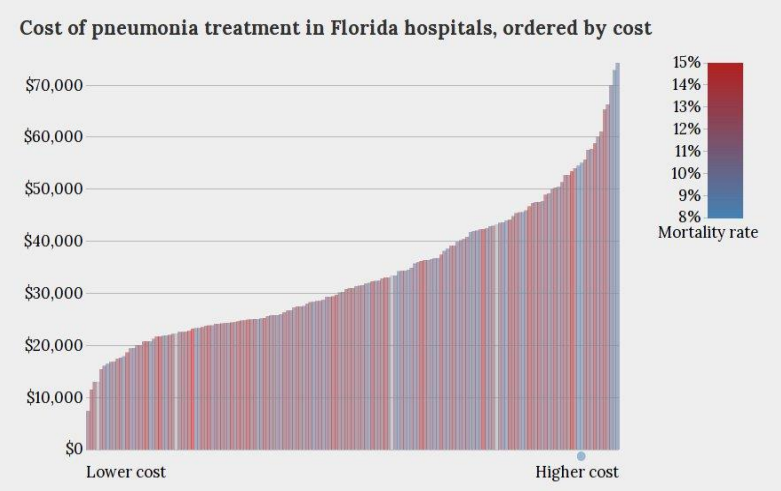


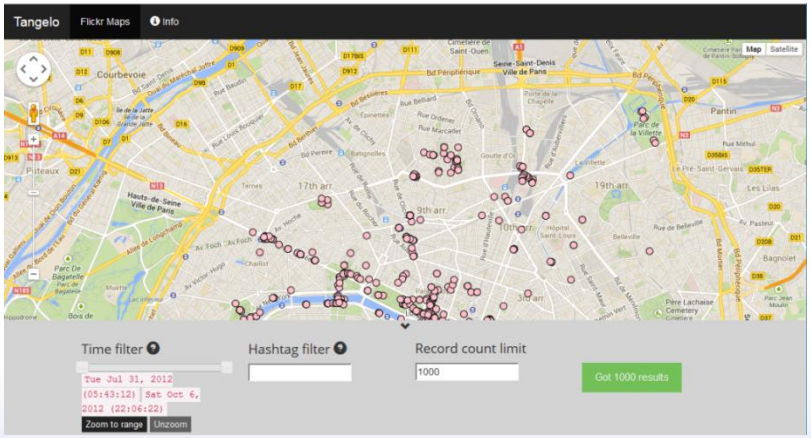
Mobile Visualization: VES/VTK



Tangelo

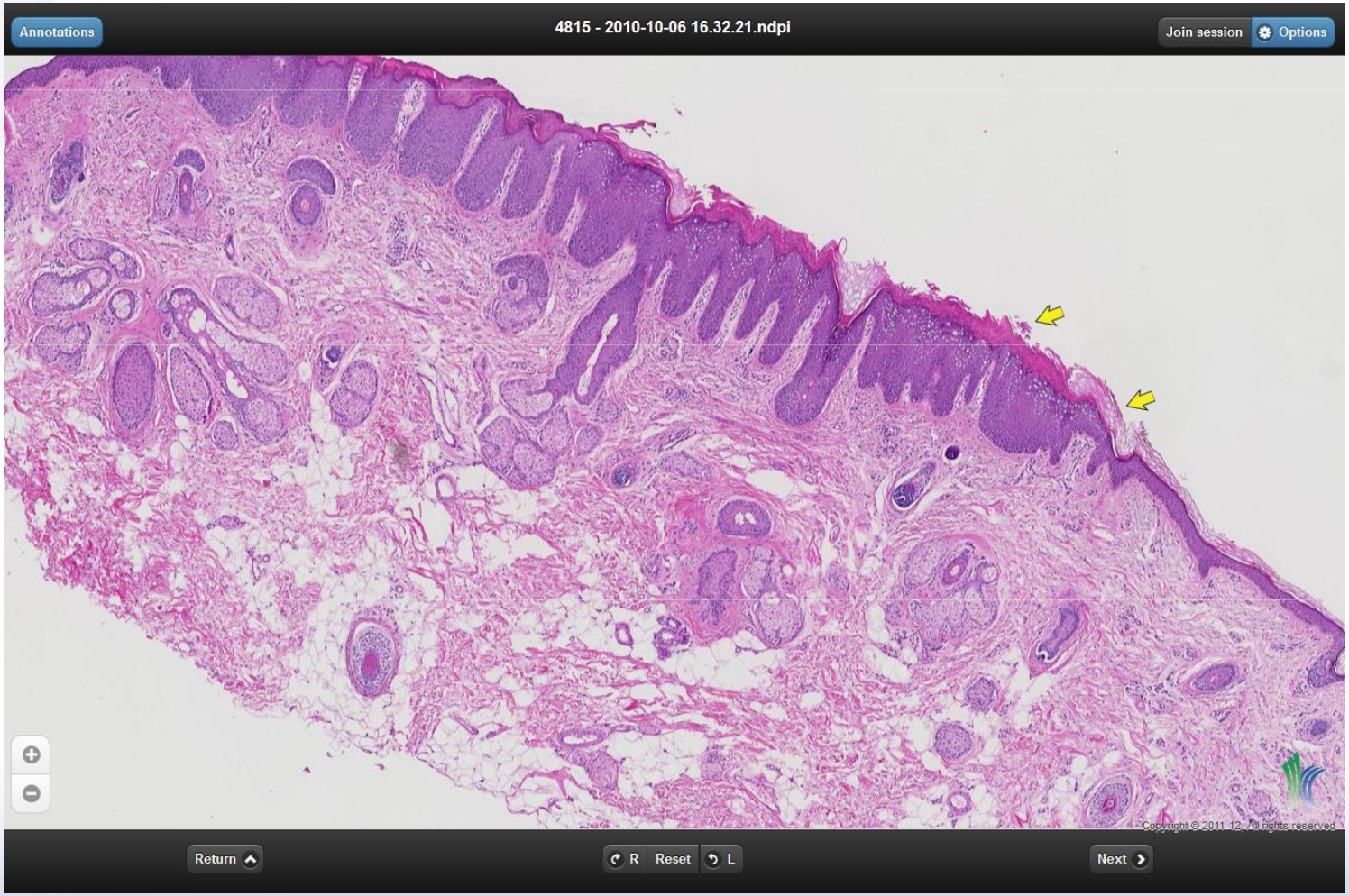
- htttp://tangelo.kitware.com
- Web framework
- HTML5 web architecture
- Packages several other frameworks too
  - Bootstrap, D3, Vega,MongoDB
- Facilitates development & deployment of web apps



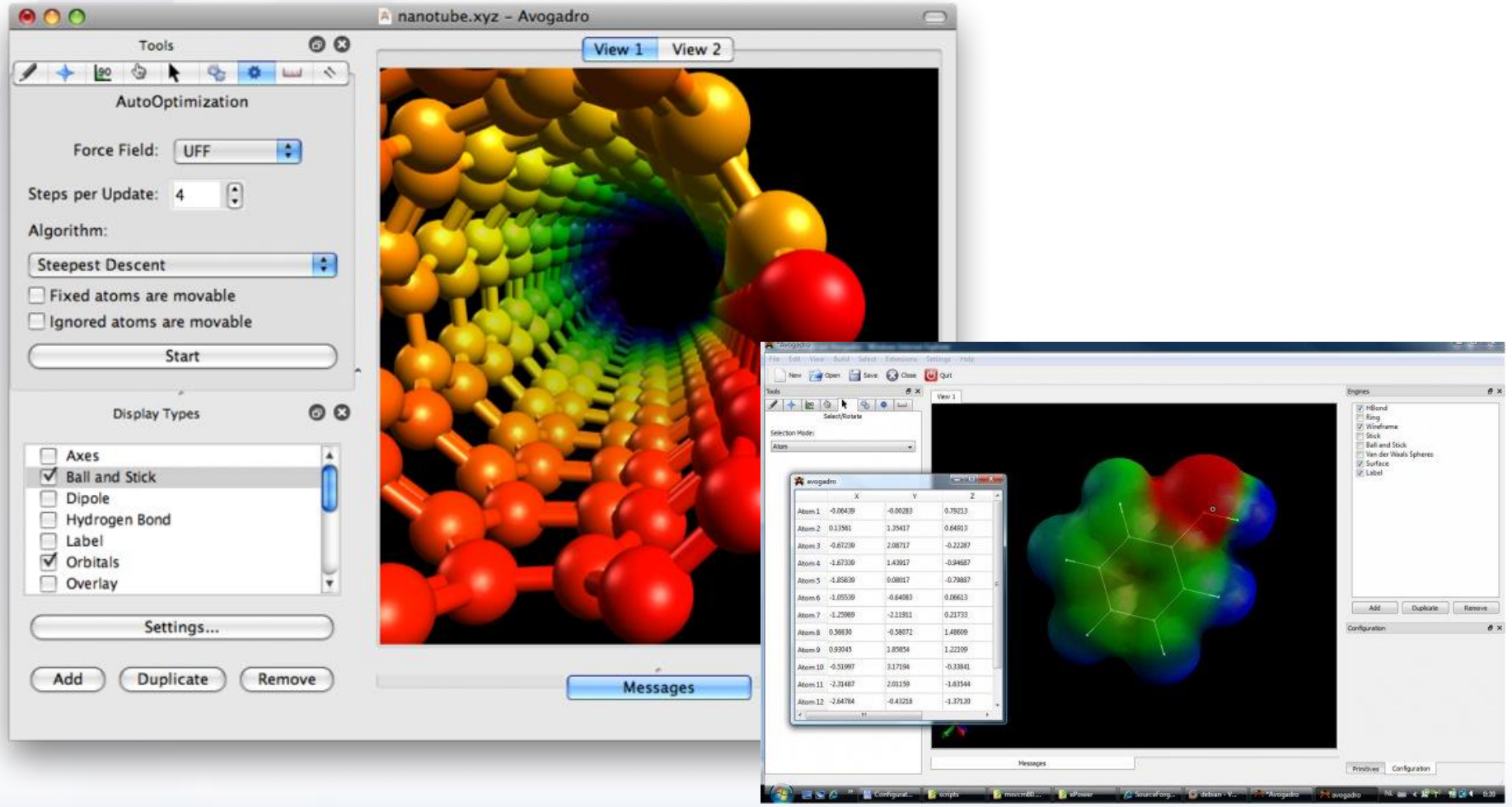


Visualization of large 2D images

- Digital Pathology: https://slide-atlas.org/

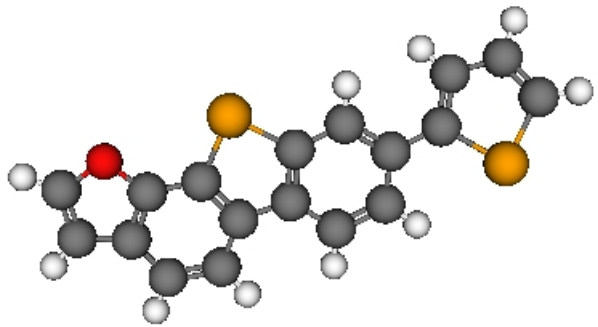


Open Chemistry



http://www.openchemistry.org/

VTKWeb and Open Chemistry



InChIKey	IZRSVHMKZWKSSV-UHFFFAOYSA-N
SMILES	c1cc2ccc3c4ccc(cc4[se]c3c2o1)-c1ccc[se]1
Formula	$C_{18}H_{10}OSe_2$
Molecular Mass	400.196
Energy	Homo -0.112
	Lumo -0.002
	Gap 0.11
	Total -5543.7168293588
Calculation	Theory BP86
	Basis STO-6G

Download

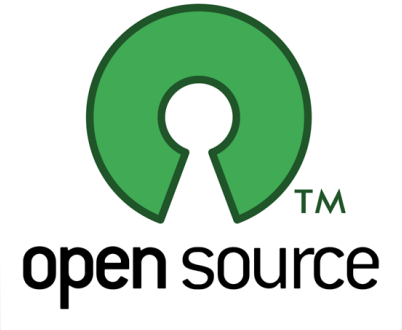
Mass	400.196
Energy	Homo -0.112

Mass	400.196
Energy	Homo -0.113

http://data.openchemistry.org/

Benefits of Open-Source

- **Extended support**
  - The Visualization Toolkit: ~$140M
- **Active maintenance**
  - Community-supported
  - Training
- **Access to expertise**
- **Reduce costs**
  - Development
  - Maintenance
  - Evolution
  - No licensing fee



Thank You!

julien.jomier@kitware.com

Kitware Europe
26 rue Louis Guérin
69100 Villeurbanne
France