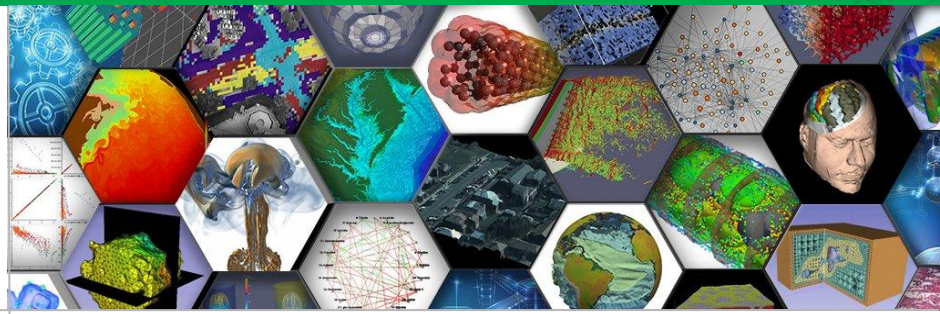
Web Visualization

Sebastien Jourdain @ Kitware

For tutorial

Make sure you have Python 3.6+
Python 3.9 is mandatory for ParaView

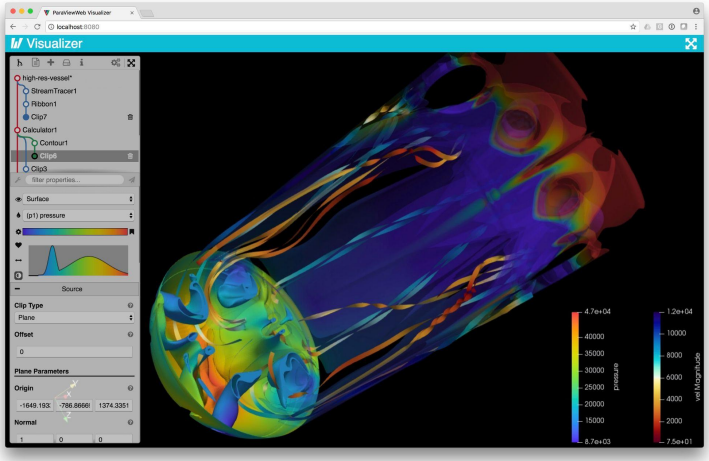
Kitware - Open Source softwares

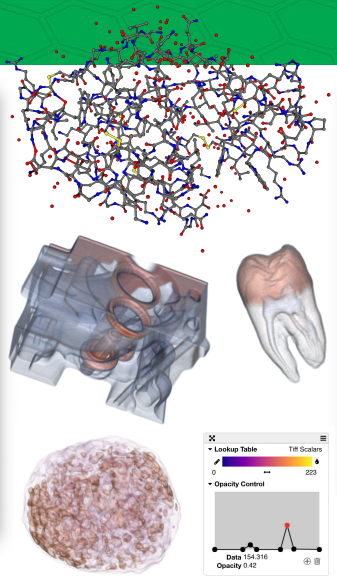


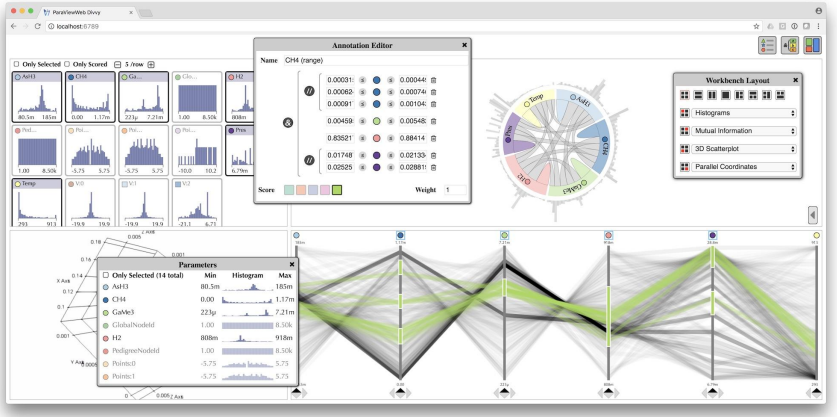
Five core areas of expertise

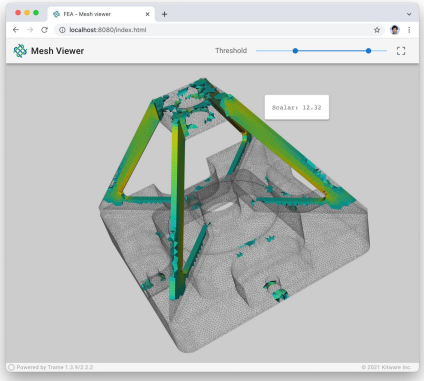
- Computer Vision — KWIVER, TeleSculptor
- Data and Analytics — Resonant
- HPC and Visualization — VTK, ParaView, CMB, tomviz
- Medical Computing — ITK, 3DSlicer, Pulse Physiology Engine, mstk
- Software Process — CMake

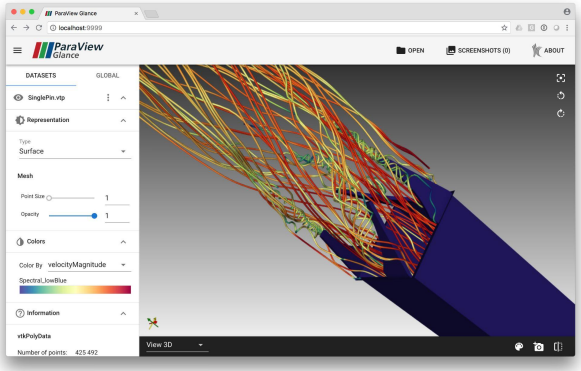
Visualization in the Web

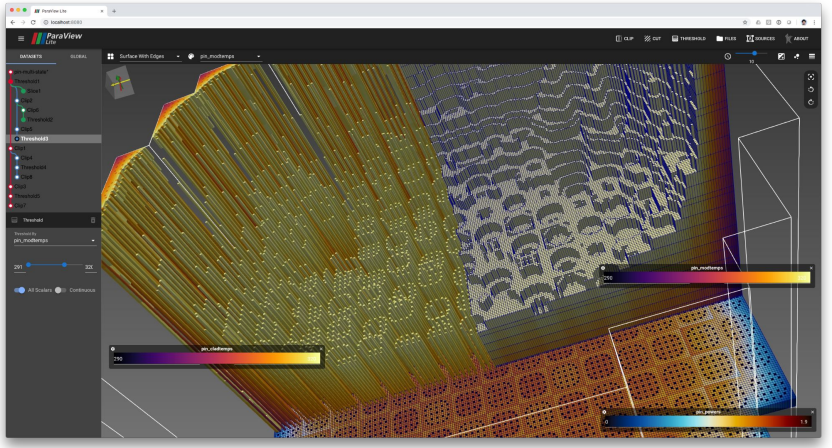






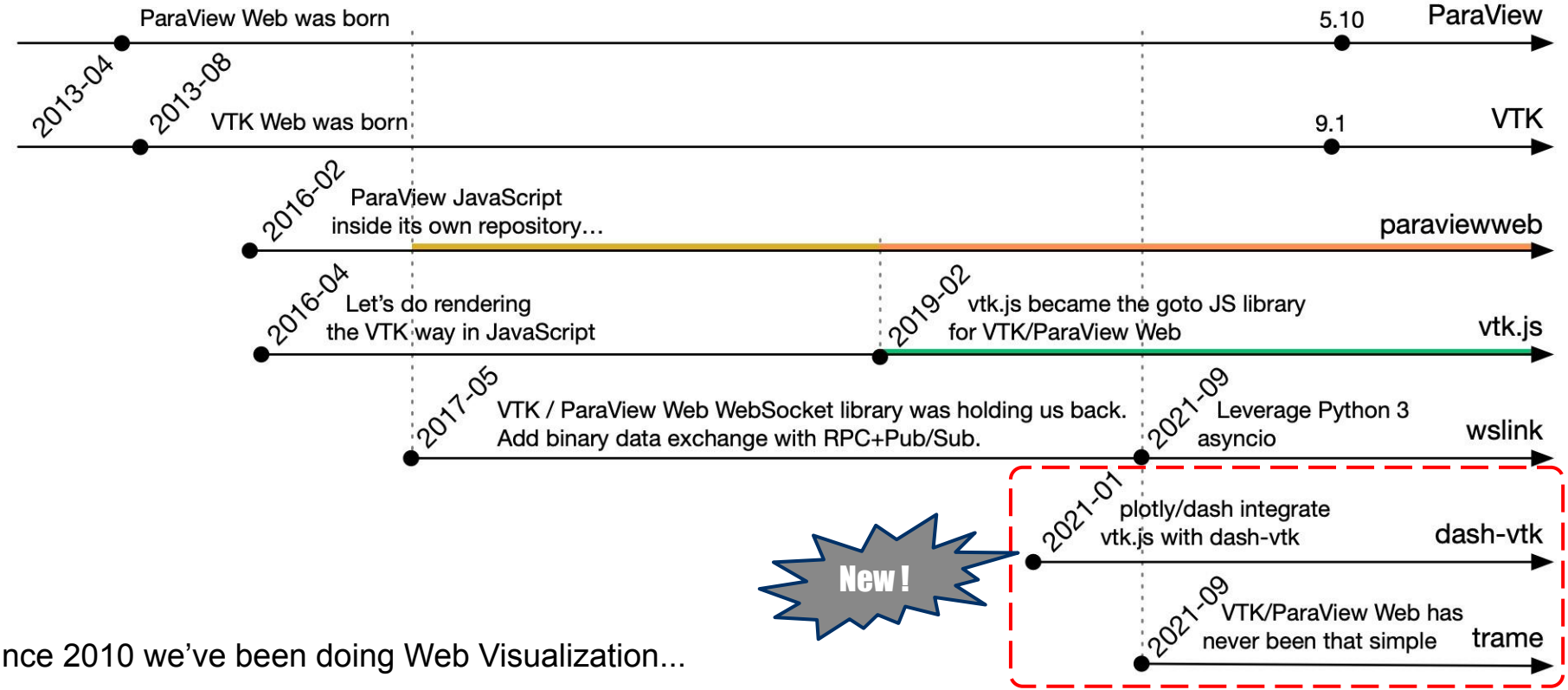






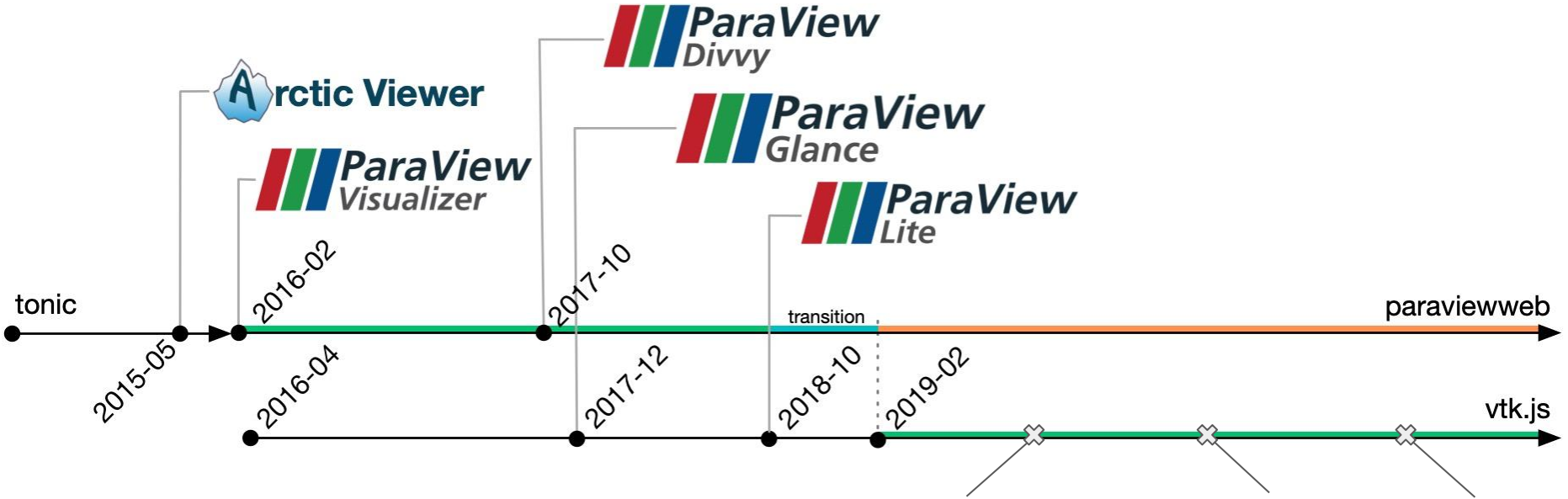
Kitware

History and background



Since 2010 we've been doing Web Visualization...

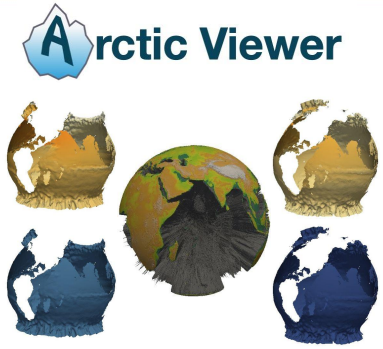
Web Visualization Applications/Demonstrators

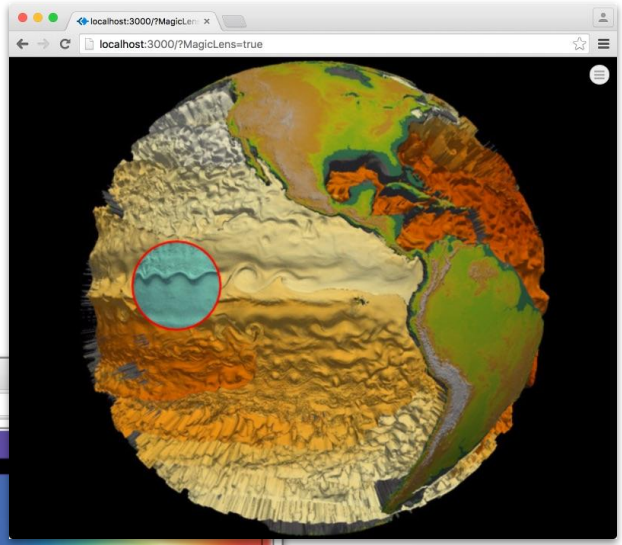


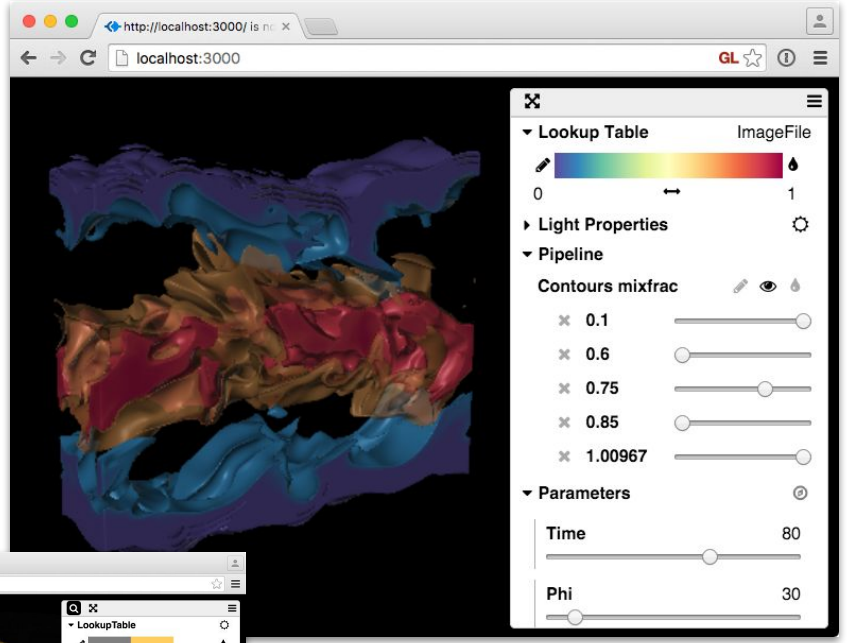
Applications / Demonstrators

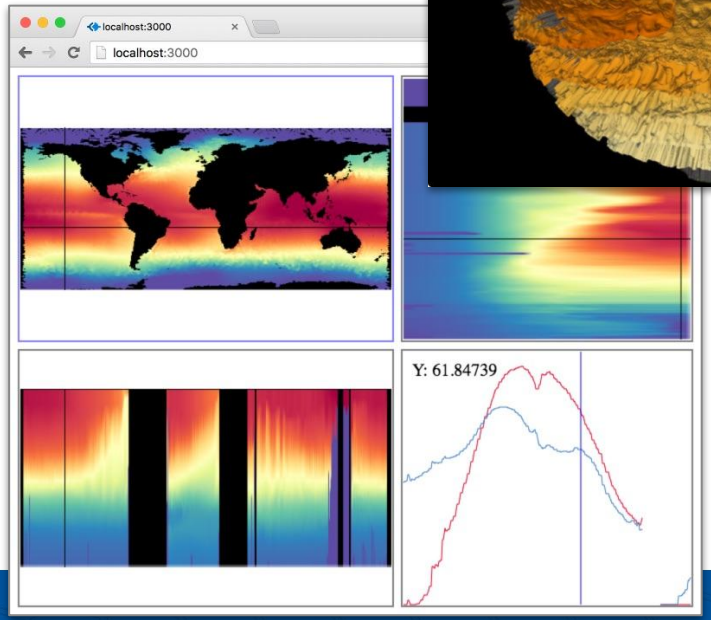
- Arctic Viewer (2015)
  - Fast and interactive viewer for ParaView based pre-processed assets
- ParaView Visualizer (2016)
  - The ParaView like application in the Web
- ParaView Divvy (2017)
  - Interactive data analytic with ParaView that does not look like ParaView
- Glance (2018)
  - The ultimate vtk.js viewer in a serverless fashion
- ParaView Lite (2018)
  - A fresh look at what ParaView in the Web could look like

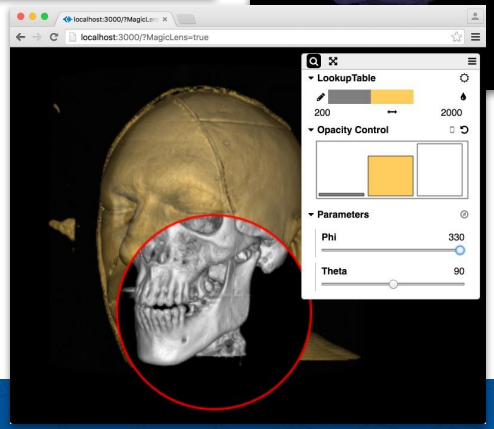
Fast and interactive viewer for ParaView/Catalyst based pre-processed assets

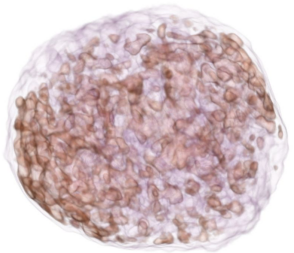


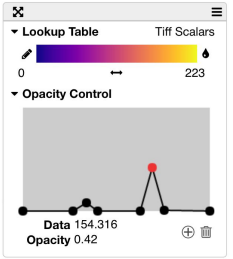








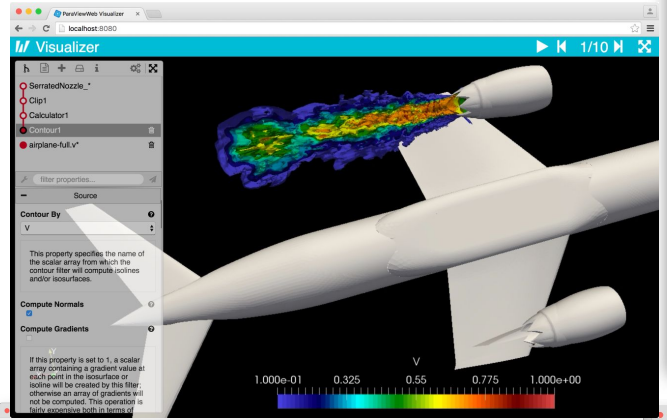


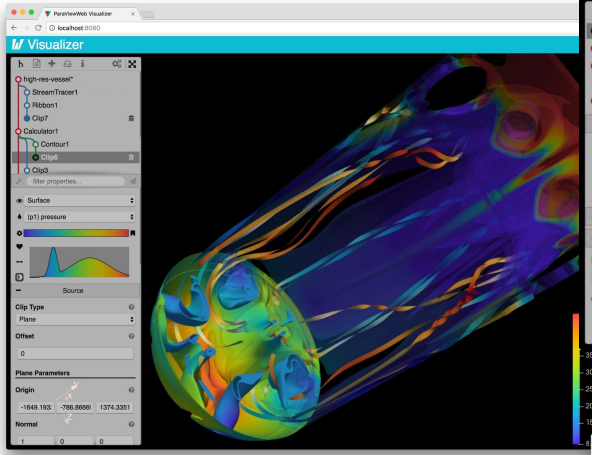


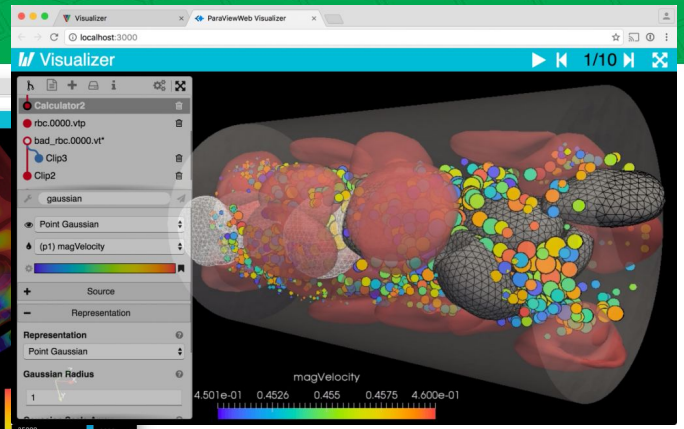
Kitware

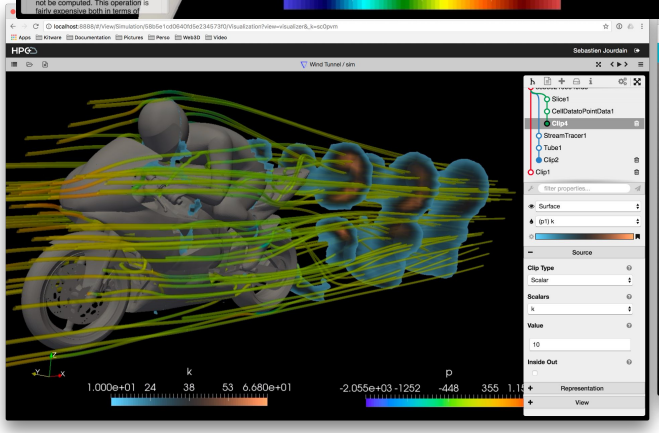
The ParaView like application in the Web

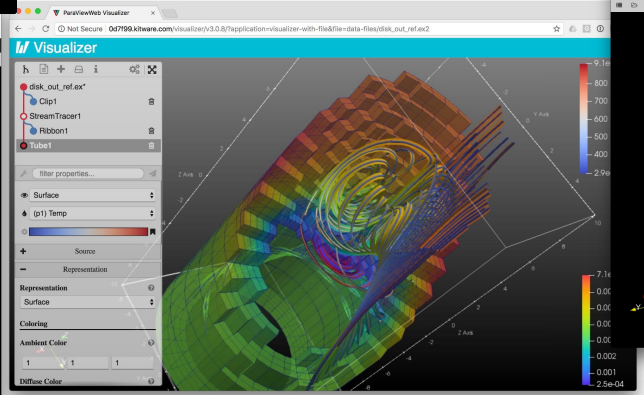
ParaView Visualizer

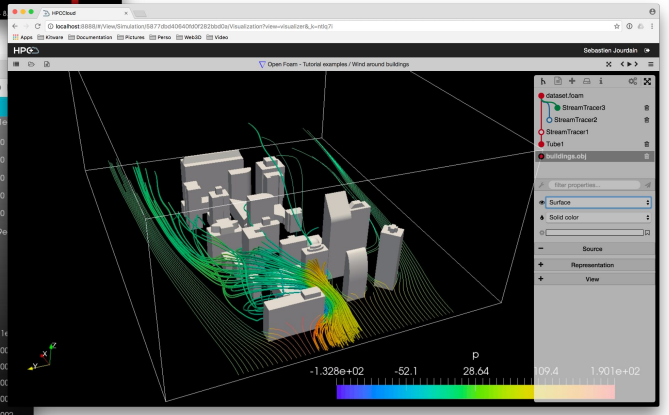








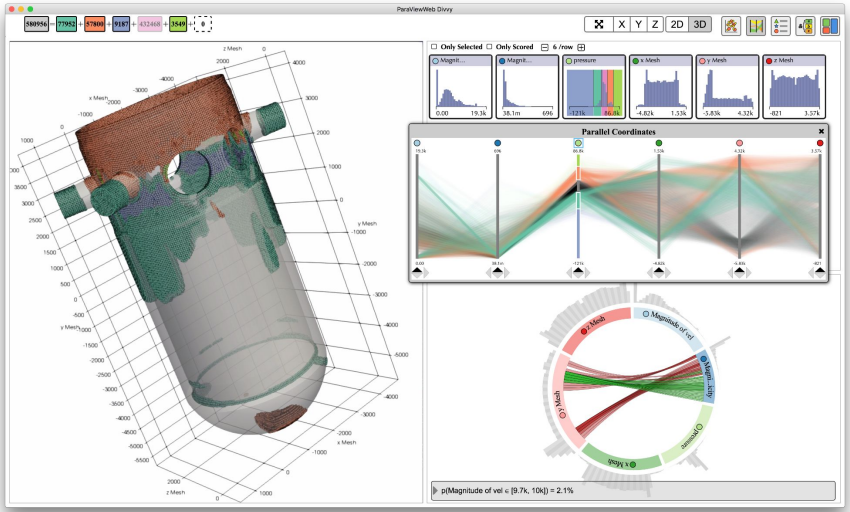


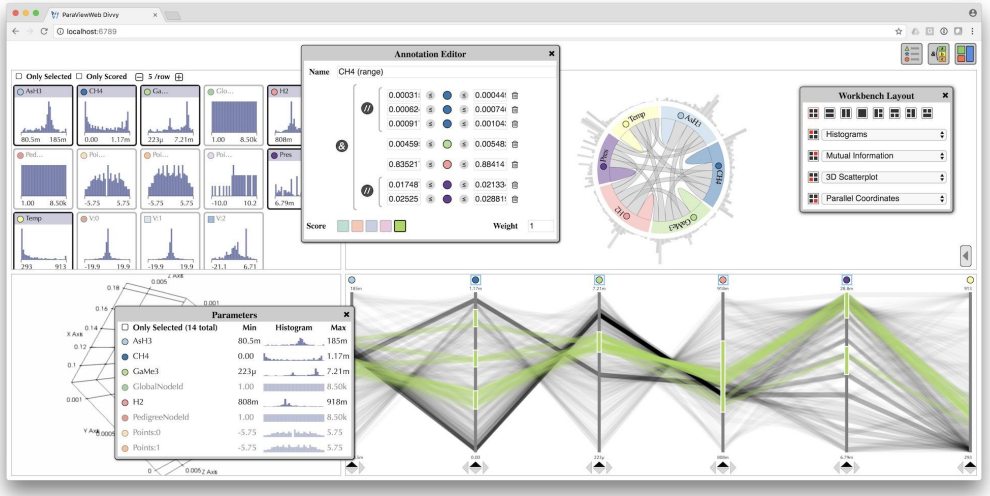


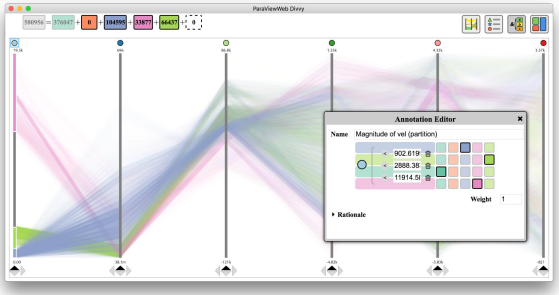
Kitware

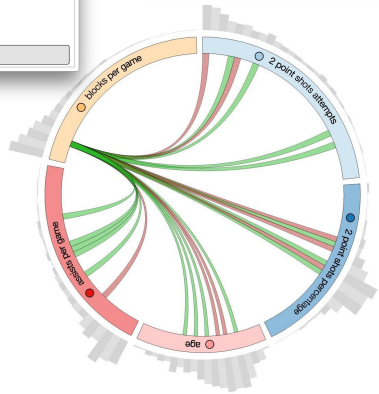
Interactive data analytic with ParaView that does not look like ParaView

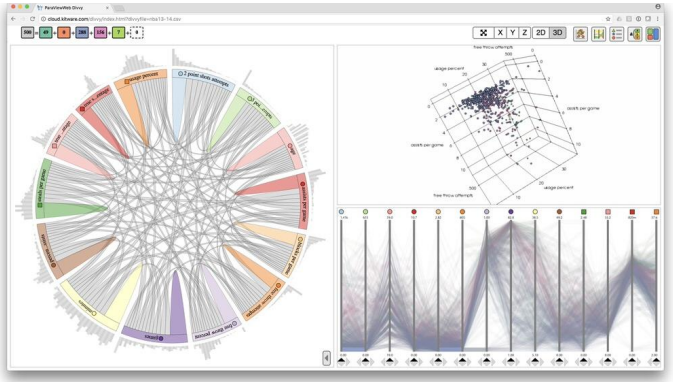
ParaView Divvy







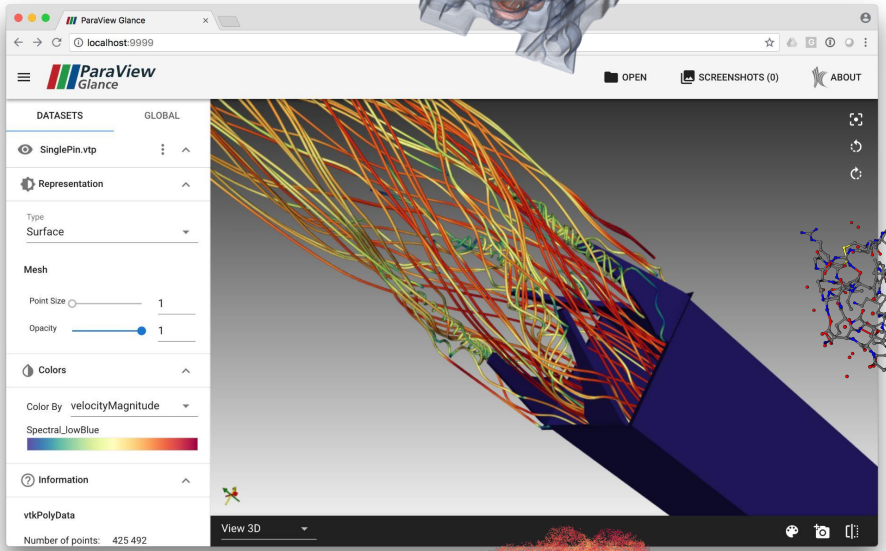


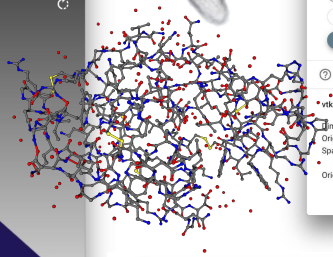


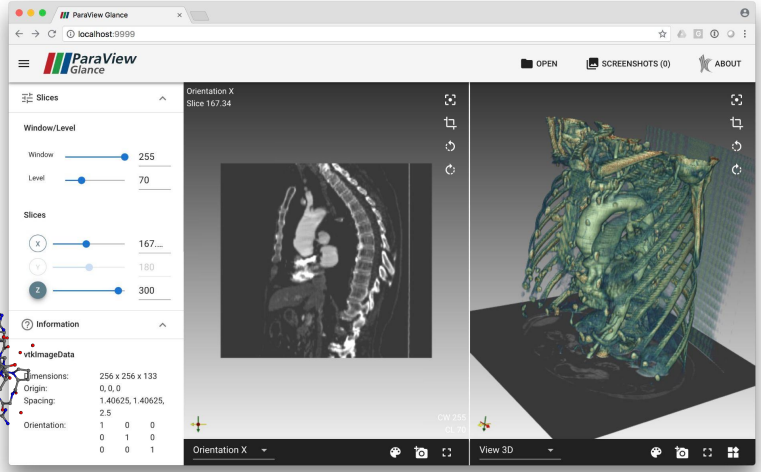
Kitware

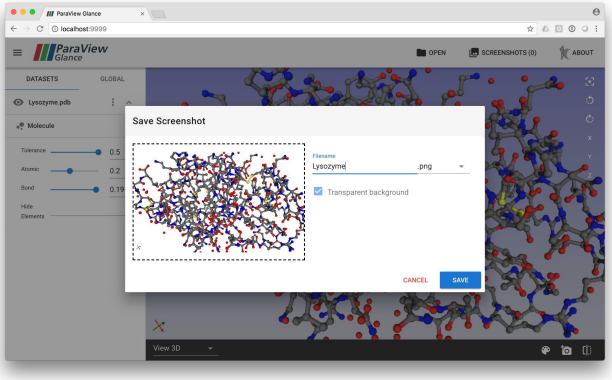
The ultimate vtk.js viewer in a serverless fashion

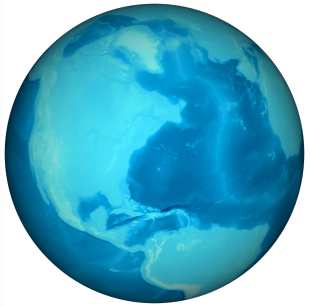
ParaView Glance



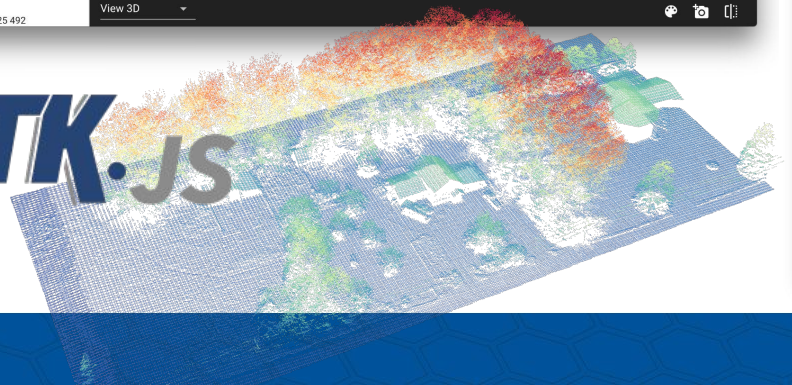








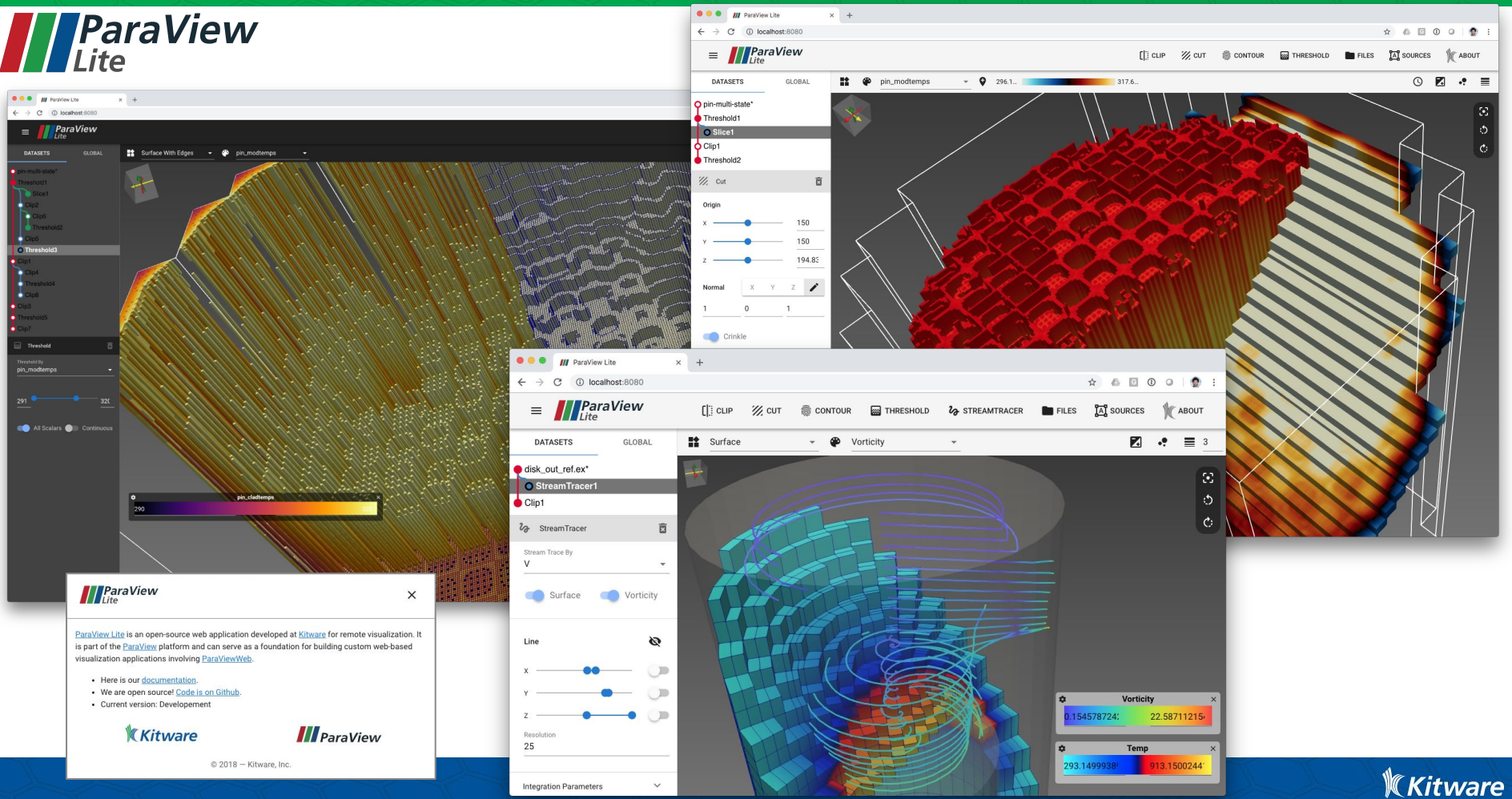
VTK.JS



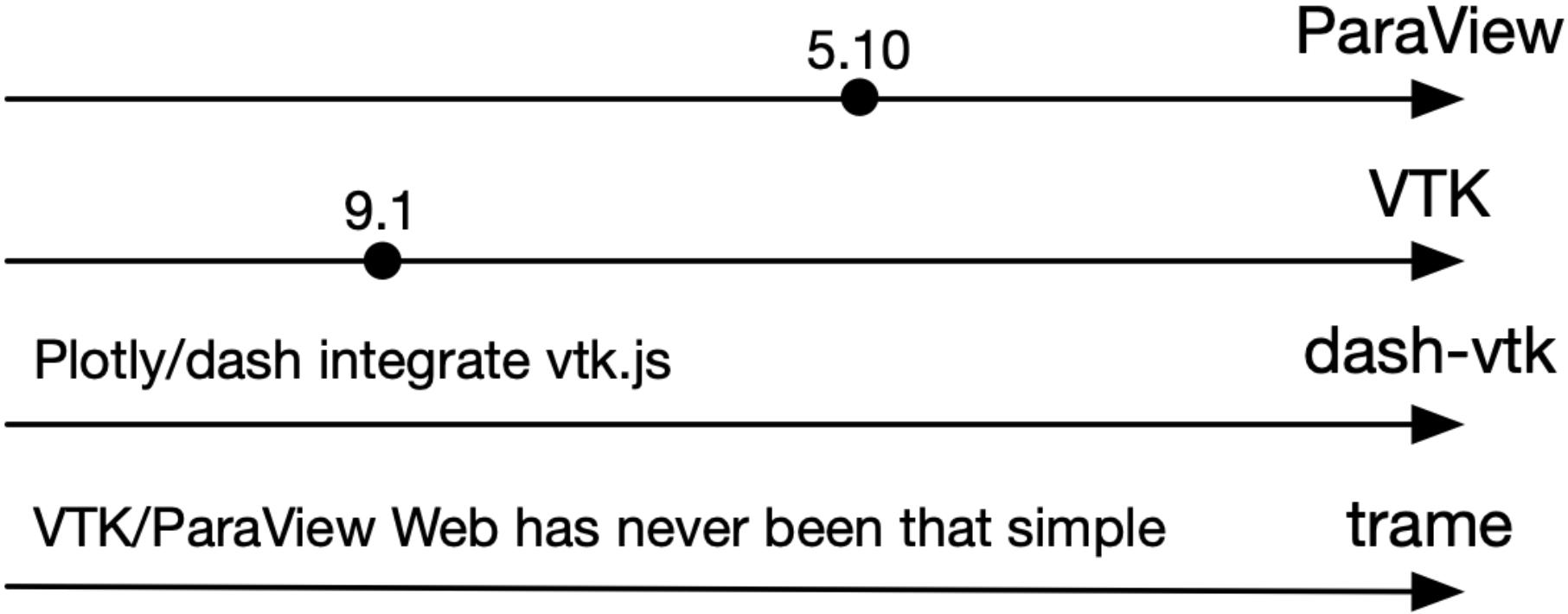
Kitware

A fresh look at what ParaView in the Web could look like

ParaView Lite



What's new?


5.10
ParaView
9.1
VTK
Plotly/dash integrate vtk.js
dash-vtk
VTK/ParaView Web has never been that simple
trame

ParaView (5.10) and VTK (9.1) releases

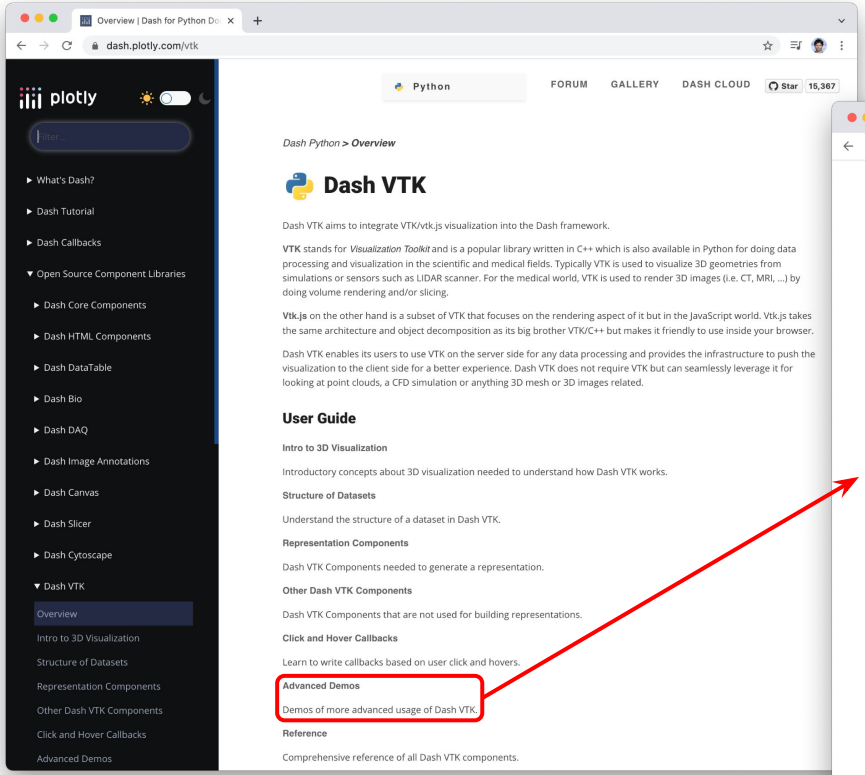
They both include the latest version of ***wslink***

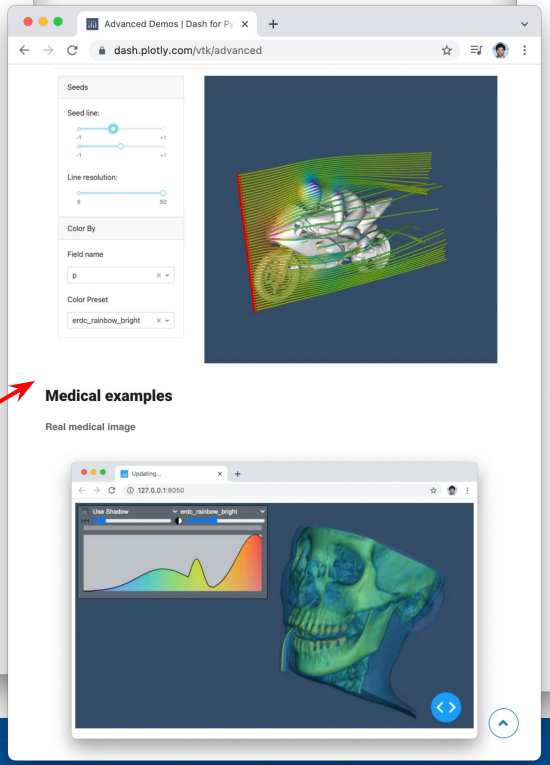
- Python 3.6+ with ***asyncio*** and ***aiohttp***

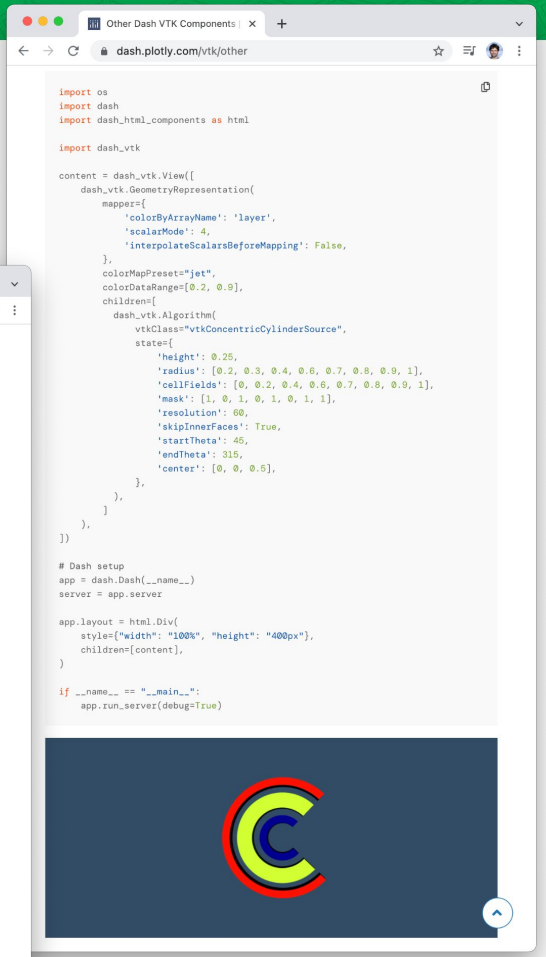
They both include improvements and fixes for their Web protocols

- Mouse wheel handling
- Support for vtkRenderView synchronization with vtk.js for
  - Volume Rendering, Cube Axes, Scalar Bar Actor

Plotly/Dash add support for 3D Viz with vtk.js







Supported languages: Python, R and Julia

Kitware

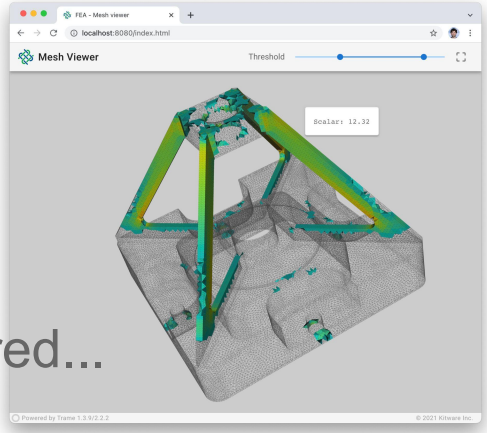
trame - VTK/ParaView Web has never been that simple!

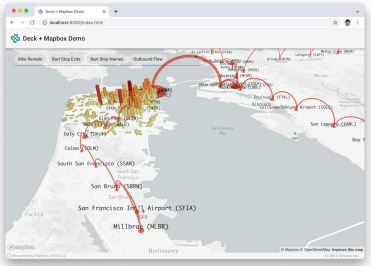
trame provides the frame for building your next Web application

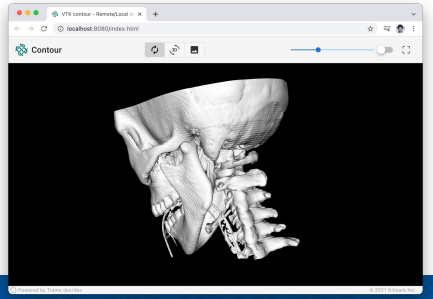
- Python is all you need to know

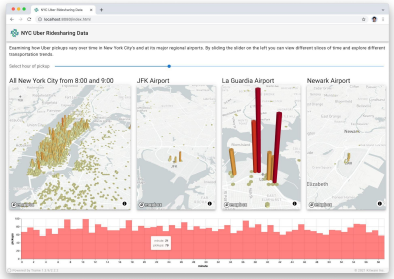
Make your first Web app today - (tutorial coming next)

- Rich set of UI components
- Play well with VTK 9.1 and ParaView 5.10
- Charts, 3D maps or 3D views, trame has you covered...










```
pip install “vtk>=9.1.0”
pip install “trame”
```

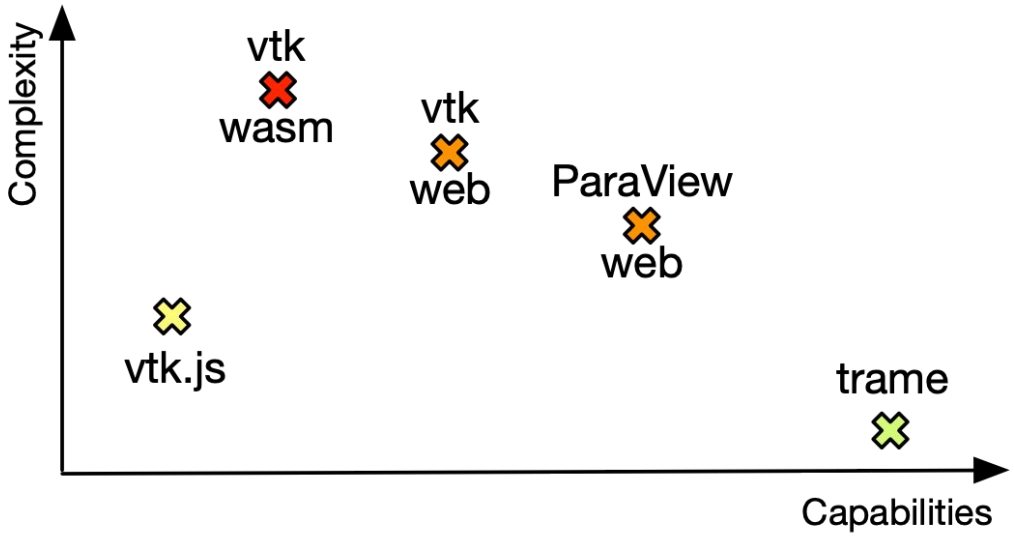

Kitware

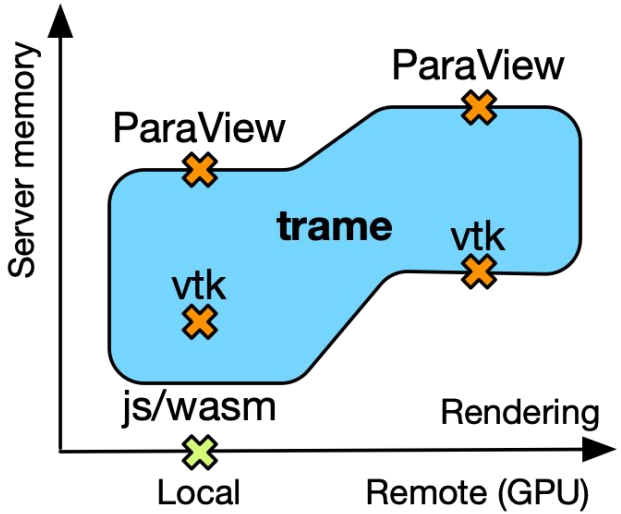
But wait a minute...

Why do we have so many options?

Blend of needs and constraints

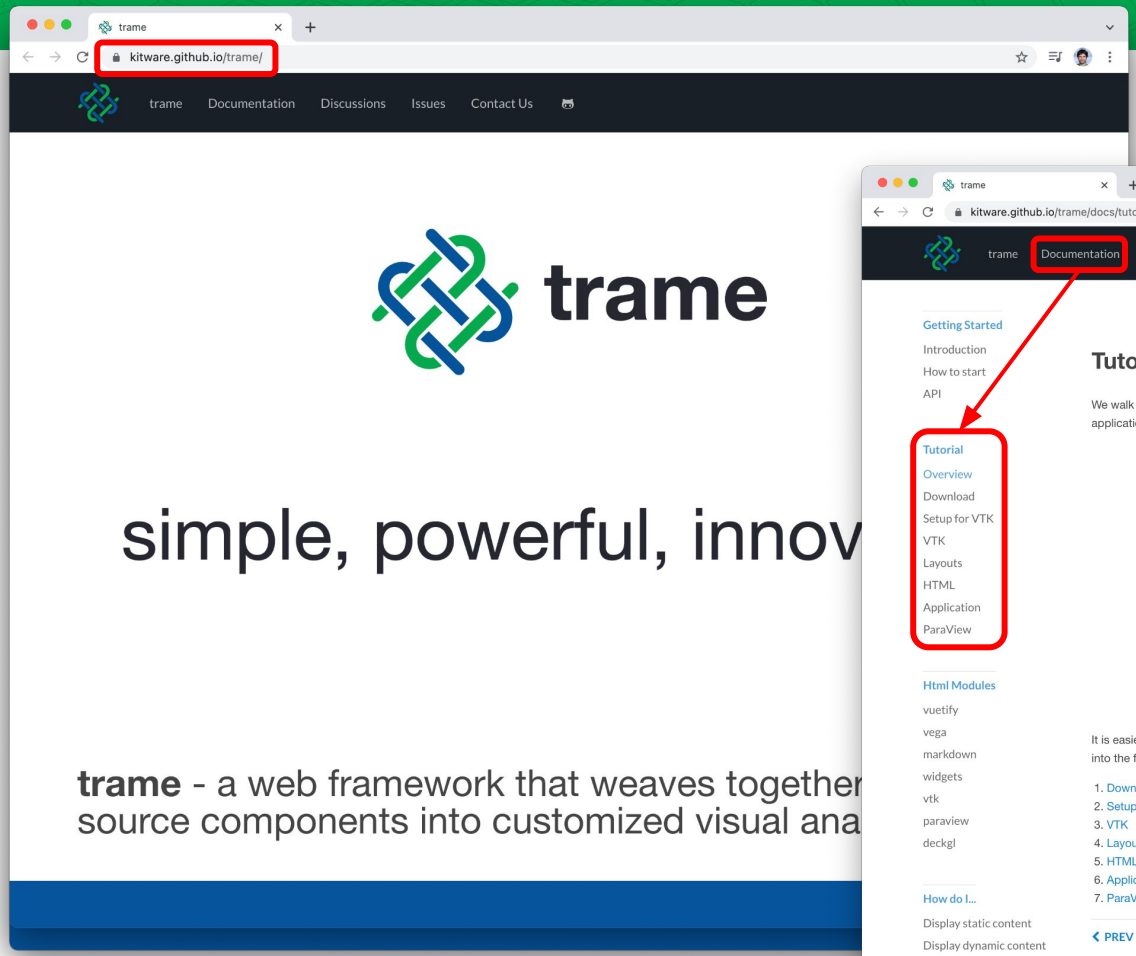
- Client vs client+server
- Resources (knowledge, hardware, operating cost…)

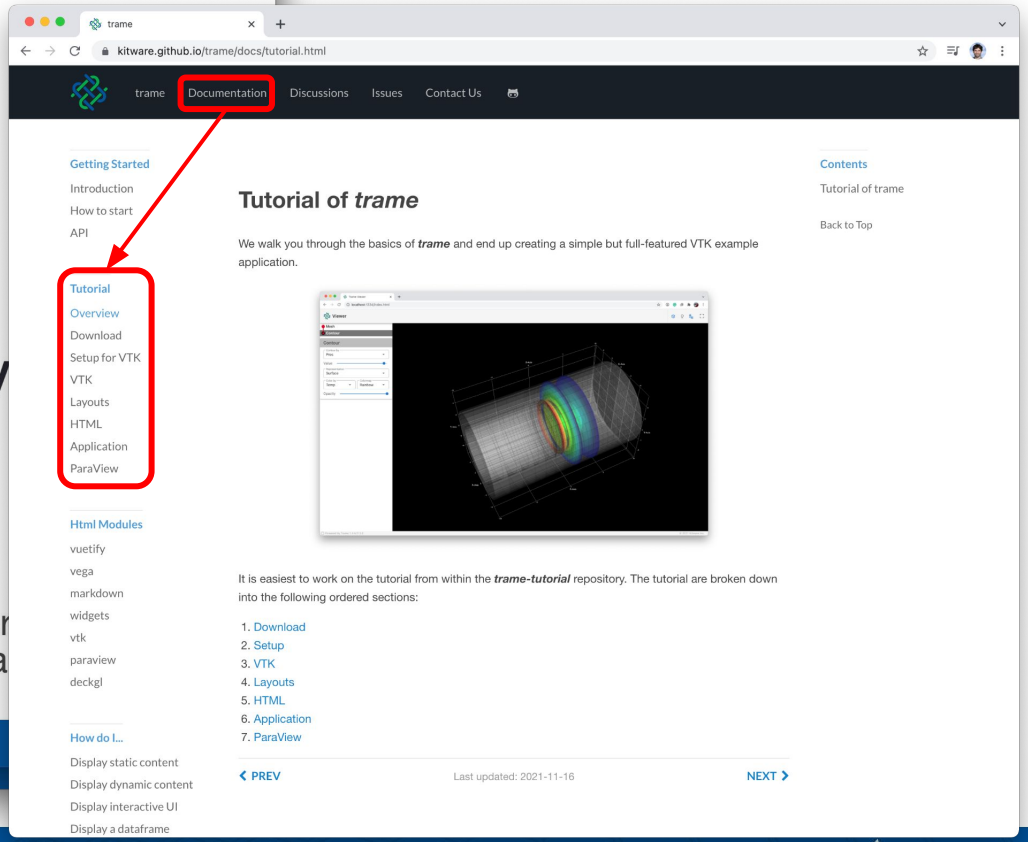




Tutorial - trame

https://kitware.github.io/trame/docs/tutorial.html

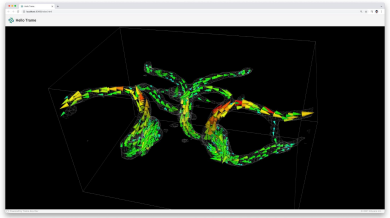


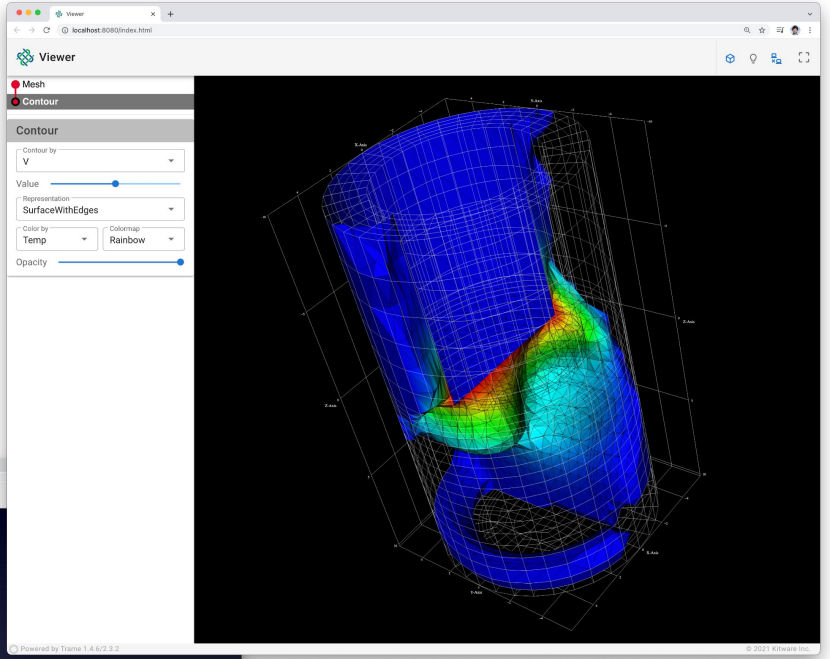


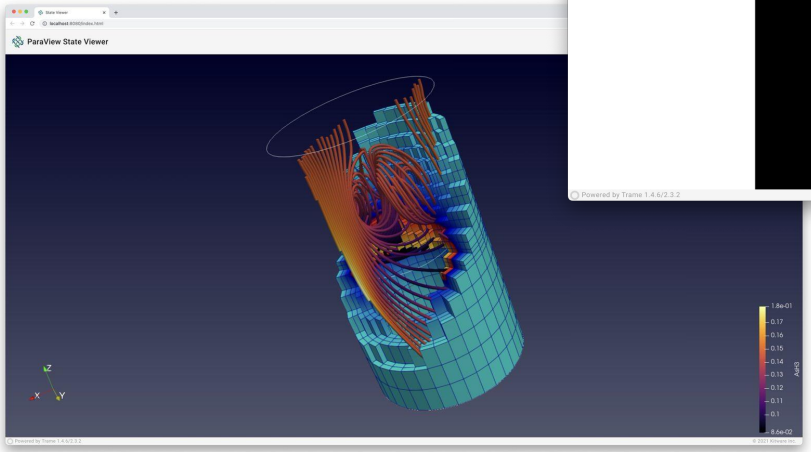
https://kitware.github.io/trame/

Kitware

Tutorial







- Download code with exercises
- Setup your Python environment
- VTK for doing 3D visualization
- Trame UI
  - Layouts
  - HTML
- Application
- ParaView

Download exercises

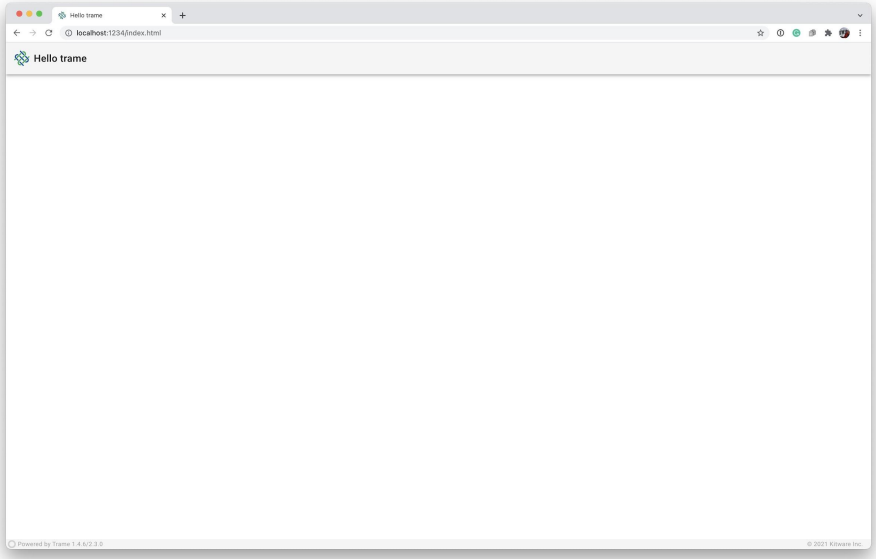
git clone https://github.com/Kitware/trame-tutorial.git

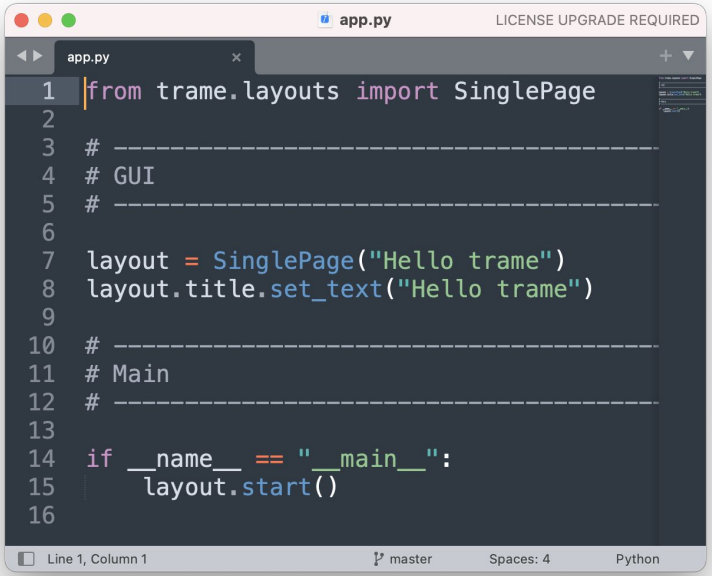
Setup a Python virtual environment

```
cd trame-tutorial
python3.9 -m venv .venv
source ./.venv/bin/activate
python -m pip install --upgrade pip
pip install "trame"
pip install "vtk>=9.1.0"
```

Run your first trame application

```
$ python ./00_setup/app.py --port 1234
```

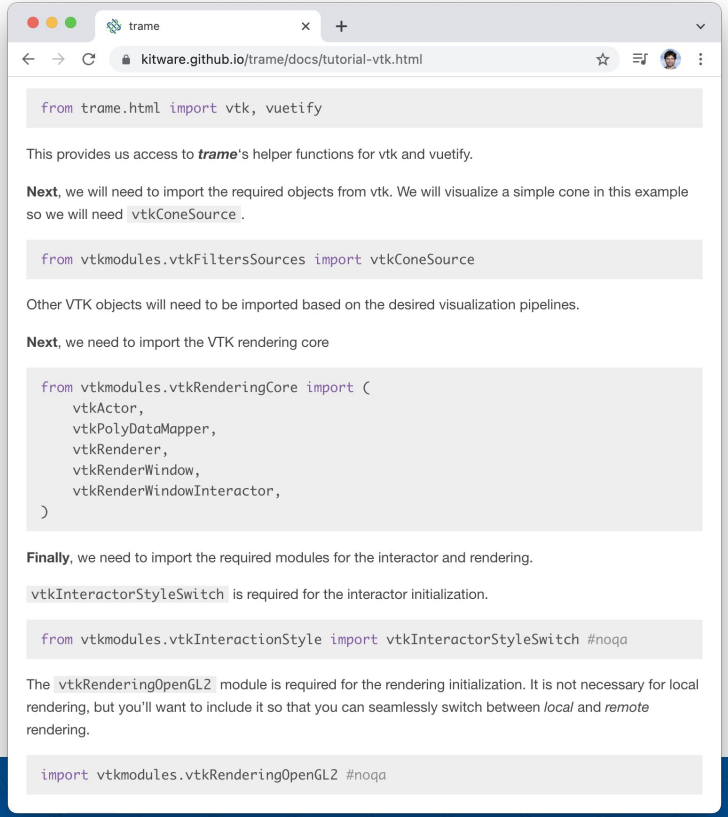


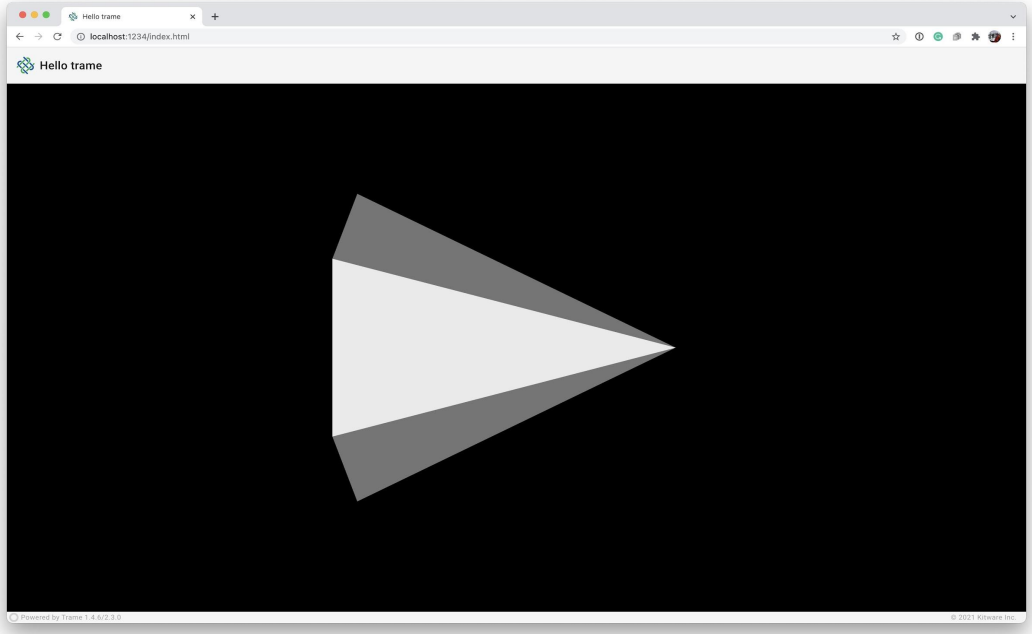


Kitware

VTK inside trame

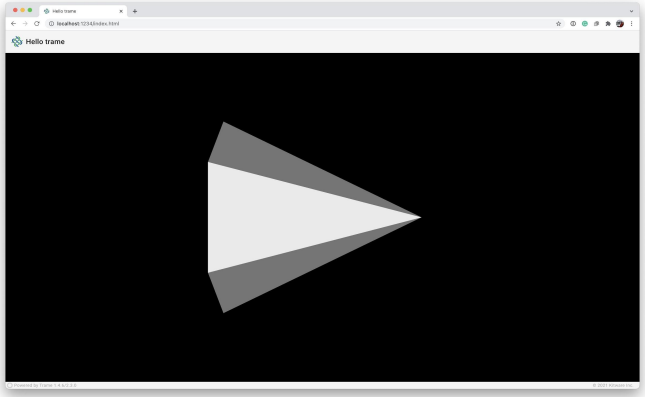
Start editing the file in `01_vtk/app_cone.py` which has the same content as `00_setup/app.py`.

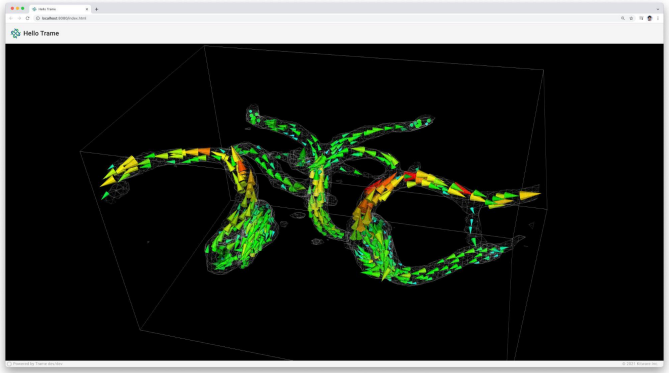




Kitware

Coding time...





solution_cone.py

LICENSE UPGRADE REQUIRED

solution_cone.py

```python
from vtkmodules.vtkFiltersSources import vtkConeSource
from vtkmodules.vtkRenderingCore import (
    vtkActor,
    vtkPolyDataMapper,
    vtkRenderer,
    vtkRenderWindow,
    vtkRenderWindowInteractor,
)

# Required for interactor initialization
from vtkmodules.vtkInteractionStyle import vtkInteractorStyleSwitch  # noqa

# Required for rendering initialization, not necessary for
# local rendering, but doesn't hurt to include it
import vtkmodules.vtkRenderingOpenGL2  # noqa


# -----------------------------------------------------------------------------
# VTK pipeline
# -----------------------------------------------------------------------------

renderer = vtkRenderer()
renderWindow = vtkRenderWindow()
renderWindow.AddRenderer(renderer)

renderWindowInteractor = vtkRenderWindowInteractor()
renderWindowInteractor.SetRenderWindow(renderWindow)
renderWindowInteractor.GetInteractorStyle().SetCurrentStyleToTrackballCamera()

cone_source = vtkConeSource()
mapper = vtkPolyDataMapper()
mapper.SetInputConnection(cone_source.GetOutputPort())
actor = vtkActor()
actor.SetMapper(mapper)

renderer.AddActor(actor)
renderer.ResetCamera()
```

Line 58, Column 1 master Spaces: 4 Python

Kitware

Local and Remote rendering

Local Rendering

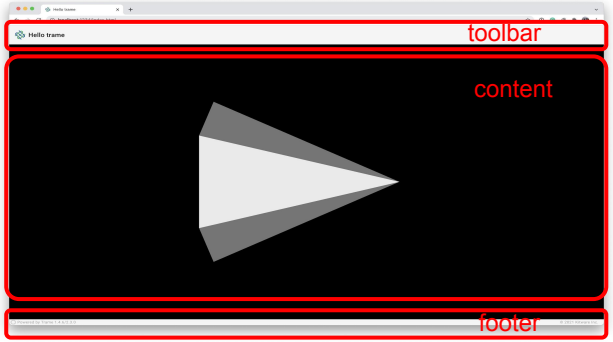
- Advantages
  - The server doesn't need a GPU
  - Interacting with the scene is as fast as it can get
- Disadvantages
  - The geometry needs to be send to the client. This can be big and therefore slow.
  - The client rendering capabilities is not known

Remote Rendering

- Advantages
  - No data movement except the rendered image
  - Rendering can use parallel and distributed processing
  - Application can serve a more diverse set of client (phone)
- Disadvantages
  - The framerate will be capped by the speed and latency of the network (~20-30 fps).
  - The server needs a GPU for faster rendering

Layouts

SinglePage

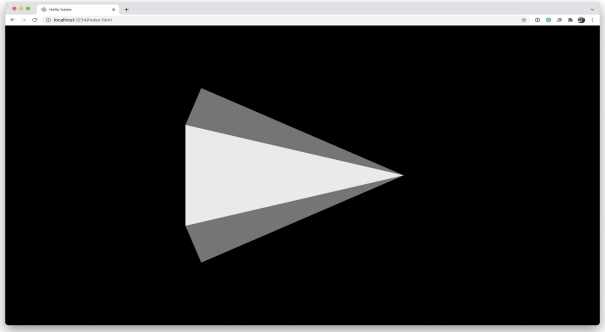


- toolbar
- logo
- title

Inside toolbar

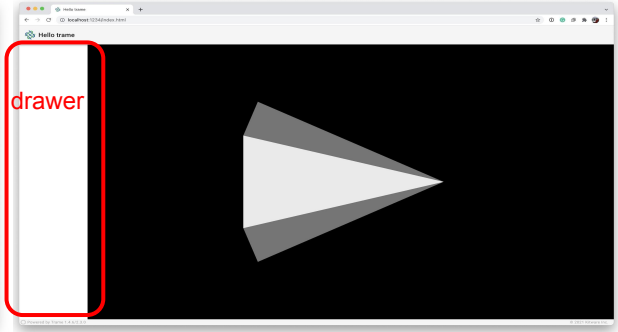
- content
- footer

FullScreenPage



- children

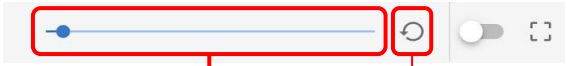
SinglePageWithDrawer



- toolbar
- logo
- title
- content
- footer
- drawer

HTML

- Vuetify (-:@) => (_)
- Shared state / actions

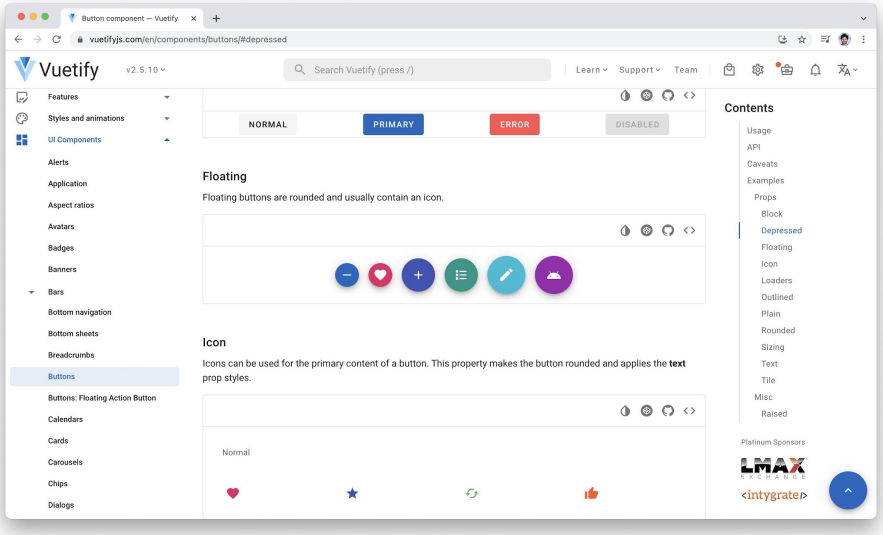


```
vuetify.VSlider(
    v_model=("resolution", DEFAULT_RESOLUTION),
    min=3,
    max=60,
    step=1,
    hide_details=True,
    dense=True,
    style="max-width: 300px",
)
```

```
@change("resolution")
def update_resolution(resolution, **kwargs):
    cone_source.SetResolution(resolution)
    html_view.update()
```

```
vuetify.VBtn(icon=True, click=reset_resolution)
```

```
def reset_resolution():
    update_state("resolution", DEFAULT_RESOLUTION)
```

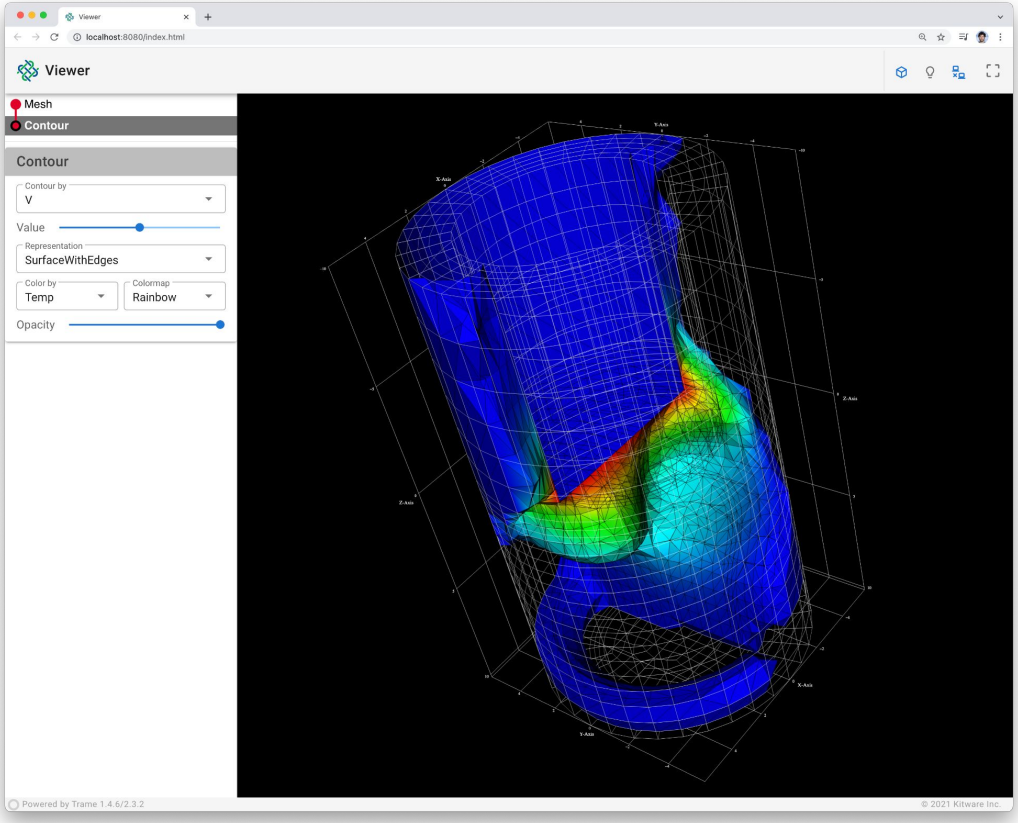


Kitware

Coding time...

Application

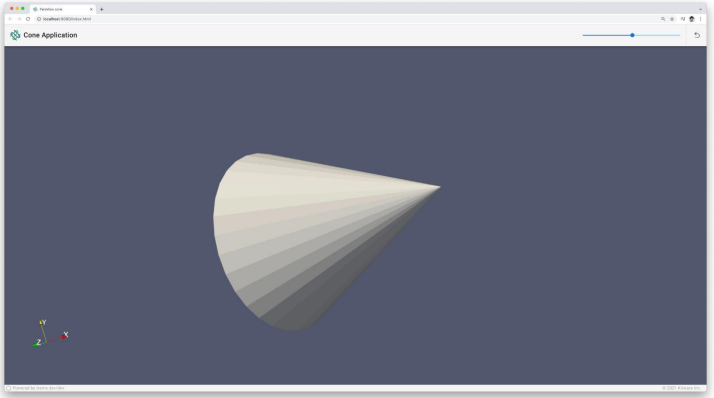
- VTK pipeline
  - CubeAxes
  - Mesh (unstructured grid)
  - Contour (filter)
- UI to control
  - Visibility
  - Representation
  - Color By
  - Color maps
  - Opacity
  - Contour By (field+value)
  - Local / Remote rendering

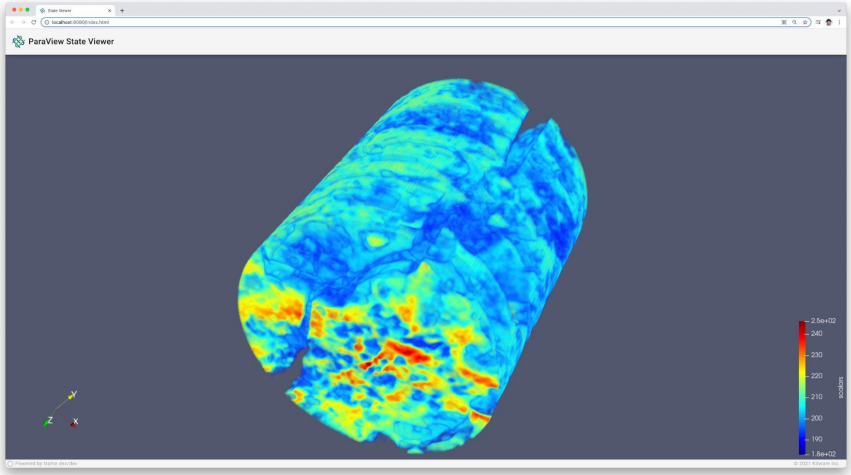


Coding time...

ParaView

- --venv path/to/venv
- Cone example
- StateLoader





Demo time...

Thank you!

sebastien.jourdain@kitware.com