Trame

An Open Source Framework for Efficiently Building Interactive Visualization and Analysis Applications

Patrick Avery

kitware

Who am I - Patrick Avery

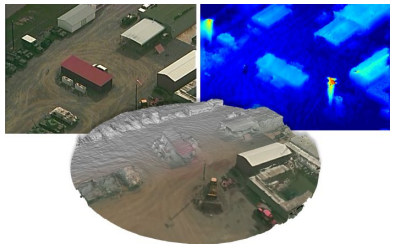
- **Ph. D. in Computational Chemistry from University at Buffalo**
  - 2019 - Research on Crystal Structure Prediction
- **Staff R&D Engineer at Kitware**
  - 2019 - Present
  - Lead a variety of materials science and chemistry projects in the scientific computing team
  - Helped Sebastien Jourdain develop trame

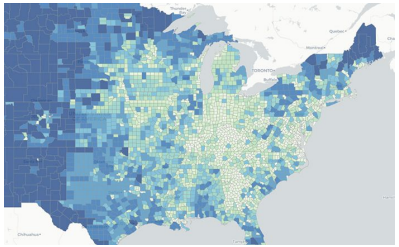
Kitware

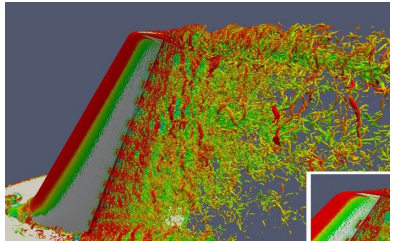
What we do?

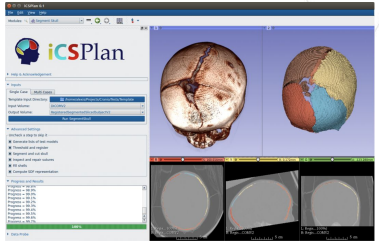
kitware

Areas of expertise / Built on open source









Computer Vision

Data and Analytics

Scientific Computing

Medical Computing

Software Solutions

kitware

Making the world a better place through custom software solutions

Customers / Various fields of application

Academics

70+ academic institutions worldwide

Government agencies

50+ government agencies and national laboratories

Commercial companies

Over 500 commercial customers

Medical

Image processing, multimodal visualization, image registration & segmentation, assisted surgery, custom software…

Energy

HPC, in-situ simulation, scientific visualisation, particle flow, fluid mechanics, ground exploration…

Intelligence

Scene analysis, big data analysis, scientific visualization, flow analysis…

kitware

Used by 536

Contributors 28

+ 14 contributors

trame

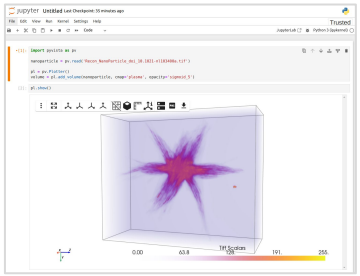
Getting more with less

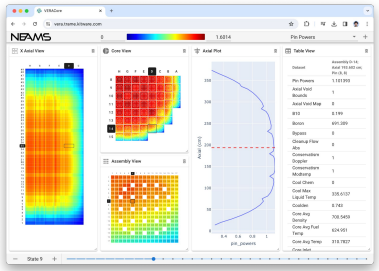
kitware

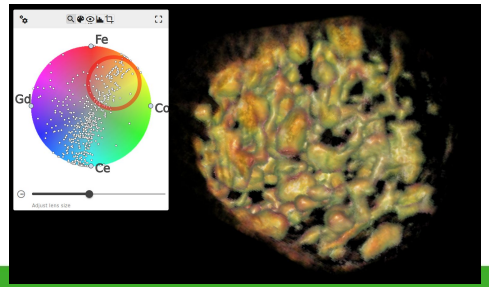
https://kitware.github.io/trame/

What is trame? (1/2)

- Open Source Python framework to create Web UI for applications
- Runs locally, in the cloud, in Jupyter or on HPC
- Seamless integration with VTK and ParaView
- Seamless integration with any Python and Vue.js libraries
- A French word - the weft or framework of a woven fabric

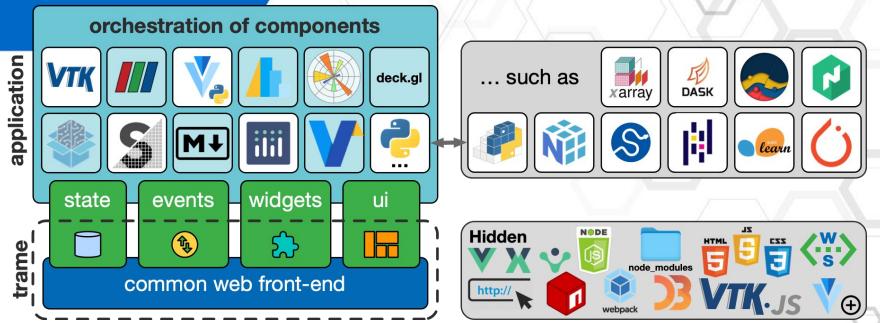






kitware

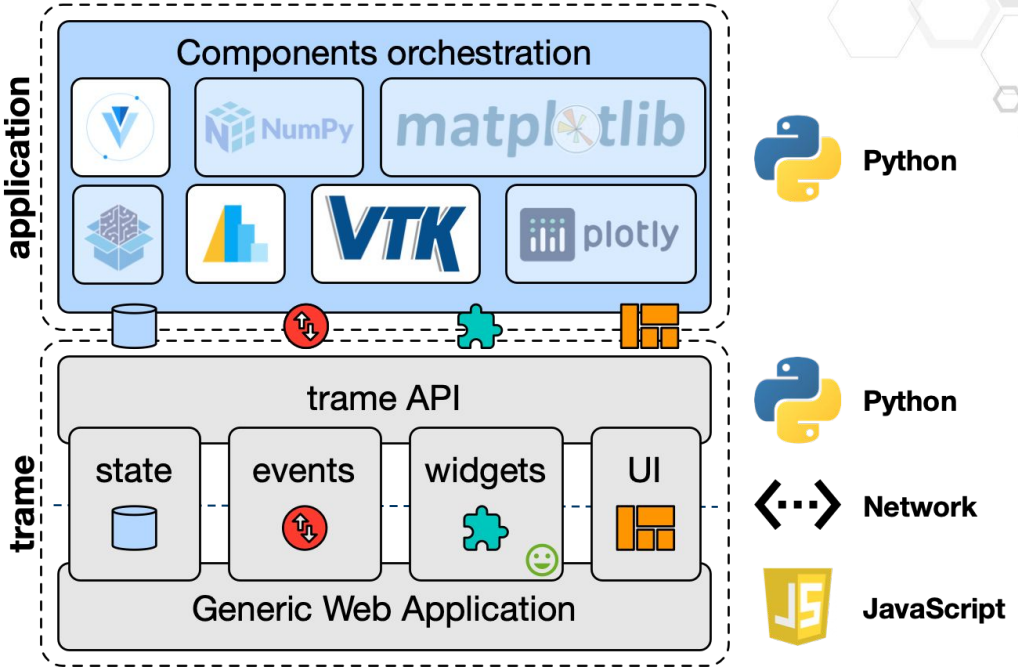
What is trame? (2/2)



- **Simple**
  All logic and UI definitions can be done in plain Python
- **Powerful**
  Python offers scientific and information data visualization with capable data processing (numpy, scipy, PyTorch, Matplotlib, VTK, ParaView…)
- **Ubiquitous**
  Runs on laptops, desktops, clusters, and the cloud while displaying everywhere (phone, tablet, laptop, workstation)

Architecture

- **Server**
  - Application logic
  - Data & processing
  - Work like everything is local
- **Client**
  - Presentation layer and user input



kitware

How to set up UI?

Look up API on Vuetify website

Convert names to trame/Python syntax (“-” becomes “_”)

https://vuetifyjs.com/en/api/v-slider/#props

You have access to all of Vuetify in Python!!

Make a slider

Bind widget value to this state variable

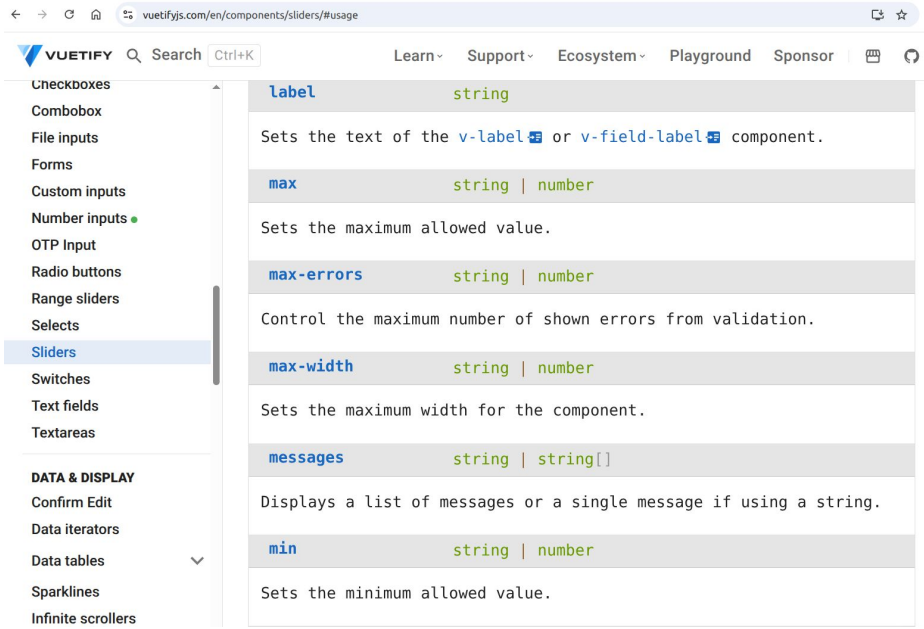
Default value for state variable

Set other widget properties

```
vuetify.VSlider(
        v_model=('sigma', 0),
        min=0,
        max=10,
        step=0.05,
        hide_details=True,
        dense=True,
        style='max-width: 300px',
)
```

Define callback function for when state variable is modified

```
@state.change('sigma')
def sigma_changed(sigma, **kwargs):
    ...
```



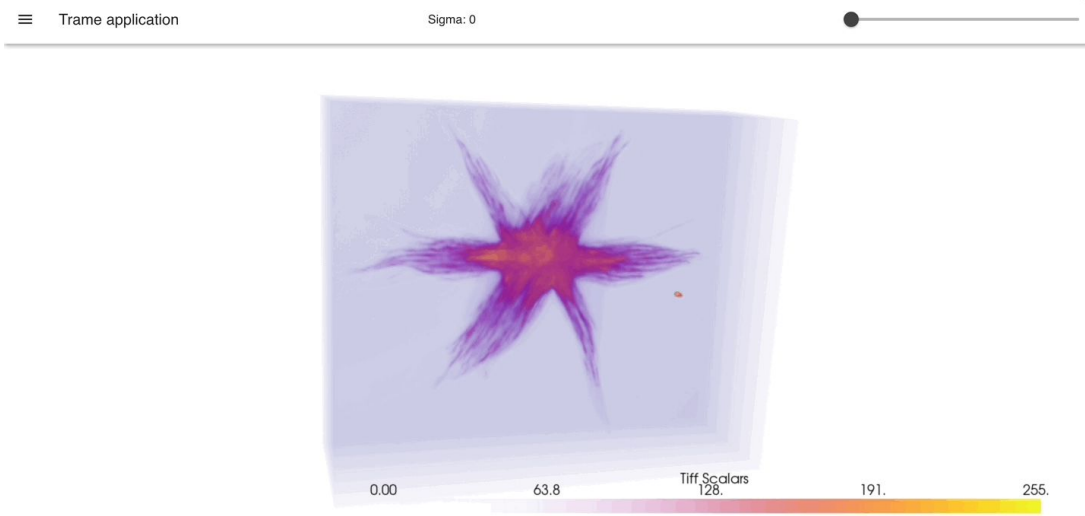
Lots of trame examples available: github.com/Kitware/trame/tree/master/examples

Alternative UI frameworks like “Quasar” and “Tweakpane” also available

kitware

M&M 2024 Interactive Demos (Toolbar UI)

```python
with SinglePageLayout(server) as layout:
    with layout.toolbar:
        html.Div("Sigma: {{ sigma }}")
        vuetify.VSpacer()
        vuetify.VSlider(
            v_model=('sigma', 0),
            min=0,
            max=10,
            step=0.05,
            hide_details=True,
            dense=True,
            style='max-width: 300px',
        )
        vuetify.VDivider(vertical=True, classes='mx-2')
```



github.com/psavery/mm2024_trame_demo

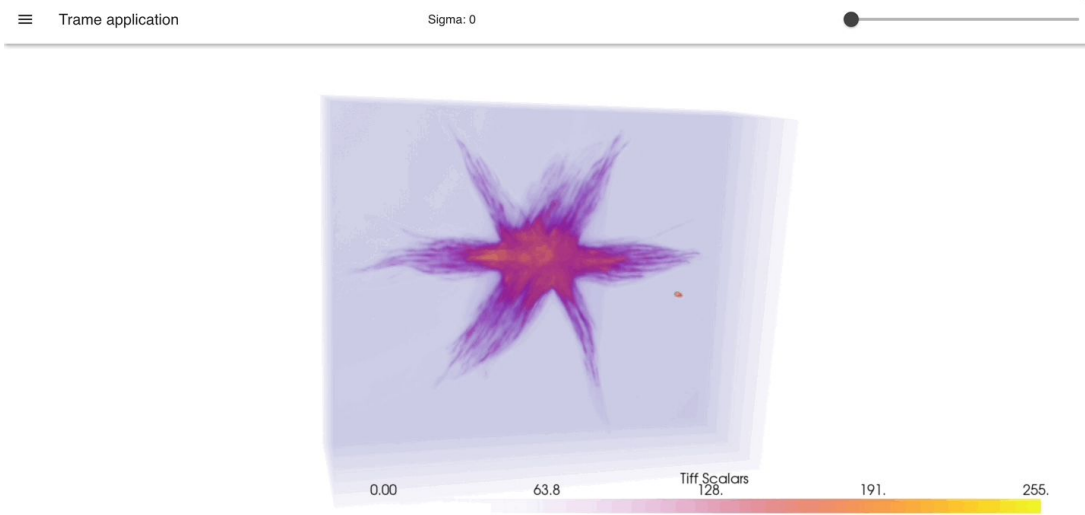
kitware

Try it out: *uv run https://raw.githubusercontent.com/psavery/mm2024_trame_demo/refs/heads/main/volume.py*

M&M 2024 Interactive Demos (Slider Callback)

```python
@state.change('sigma')
def sigma_changed(sigma, **kwargs):
    shape = np_data.shape
    np_data[:] = scipy.ndimage.gaussian_filter(
        original_data.reshape(shape[::-1]),
        sigma,
    ).reshape(shape)

    # Update the view
    data.Modified()
    ctrl.view_update()
```



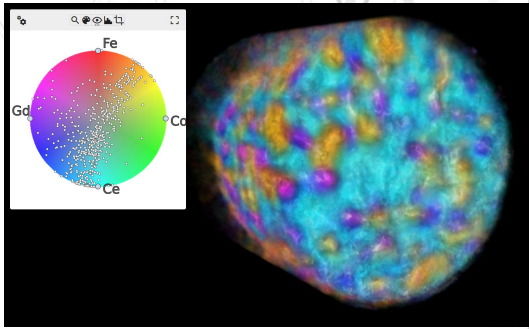
github.com/psavery/mm2024_trame_demo

kitware

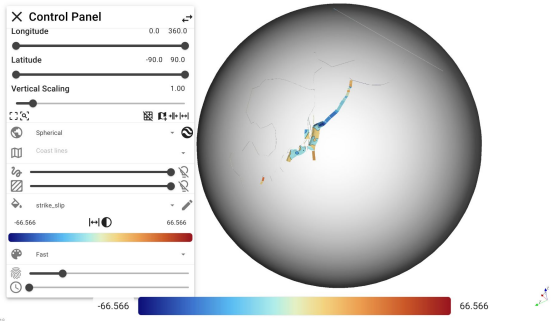
Try it out: ***uv run https://raw.githubusercontent.com/psavery/mm2024_trame_demo/refs/heads/main/volume.py***

Try it out?

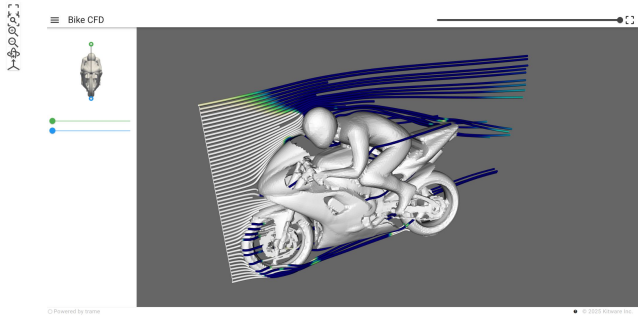
- **Easiest way to try out a trame app is through uv**
  - Install uv: docs.astral.sh/uv/getting-started/installation
    - Easiest way might be `pip install uv`
- **Run the command below the app you want to try**



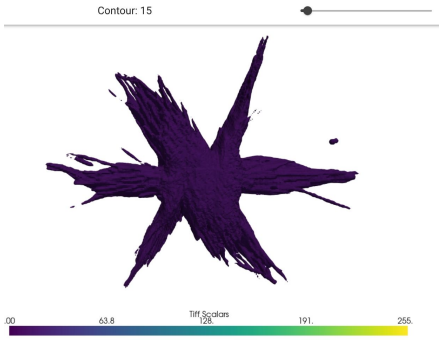
uvx multivariate-view



uvx parsli



uv run https://raw.githubusercontent.com/Kitware/trame/refs/heads/master/examples/06_vtk/04_wasm/app.py

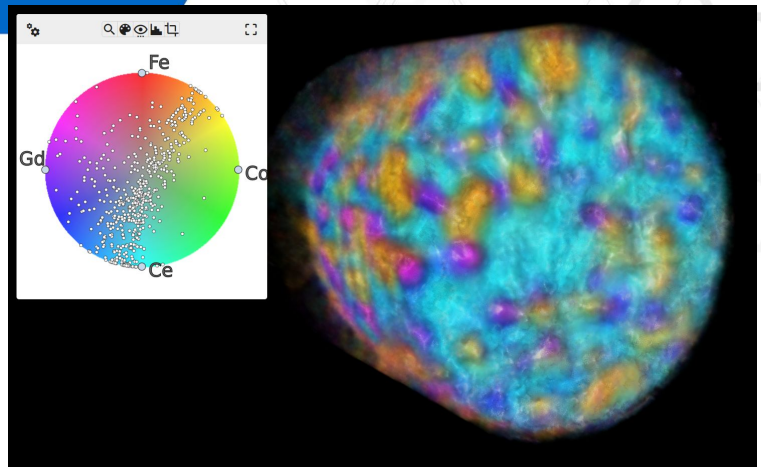


uv run https://raw.githubusercontent.com/psavery/mm2024_trame_demo/refs/heads/main/contour.py

kitware

Multivariate View Example

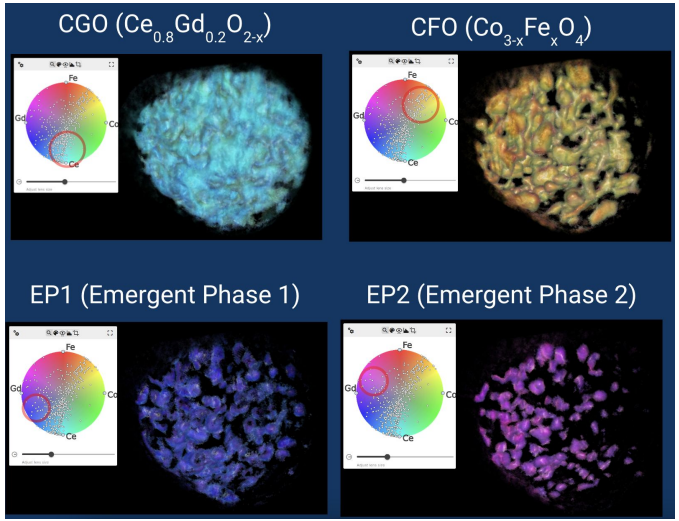
- **Multivariate Volume Viz**
  - XRF Tomography from BNL



- **~1500 lines of Python**
- **~300 lines of JavaScript**

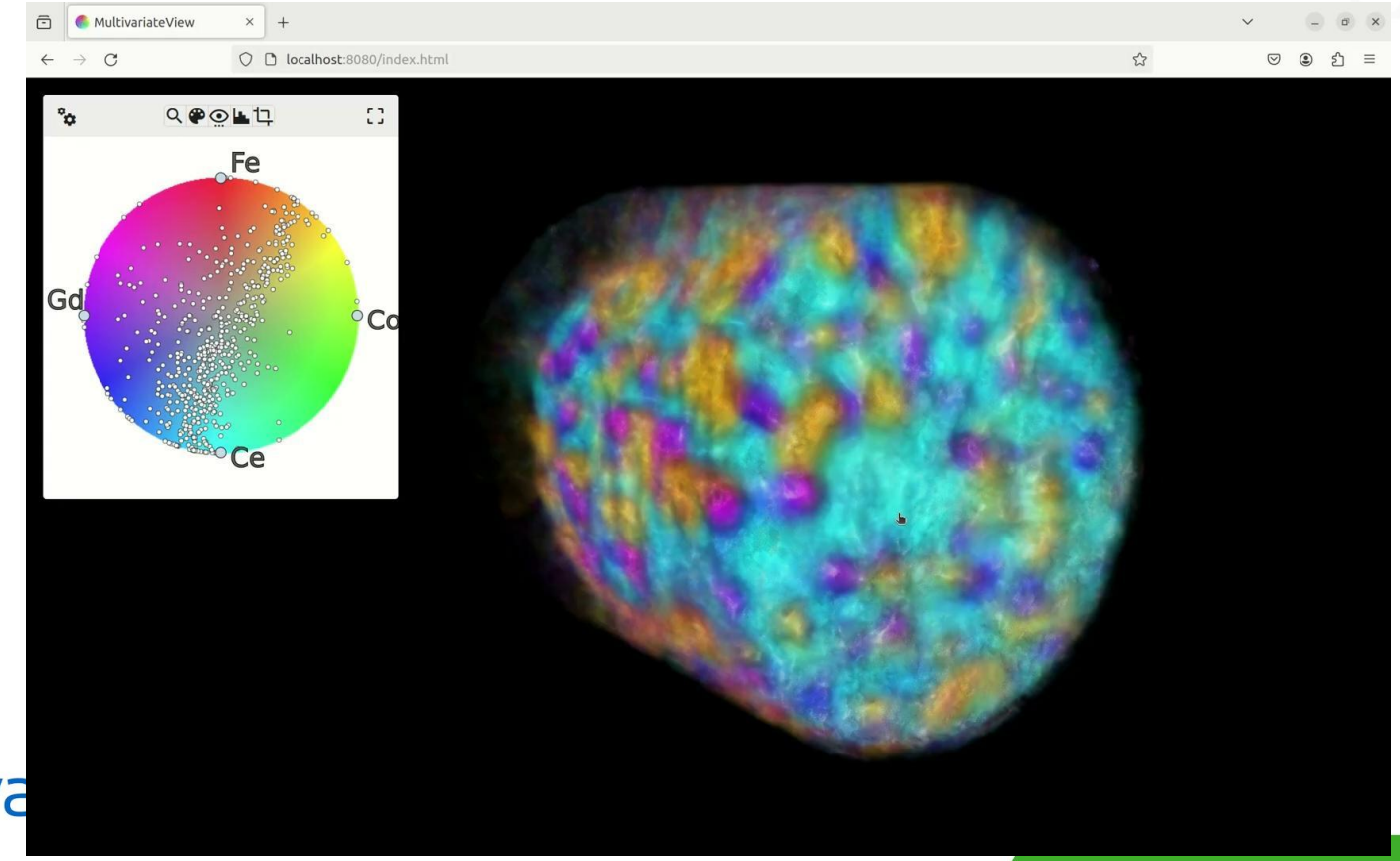
github.com/Kitware/multivariate-view

Run: ***uvx multivariate-view***

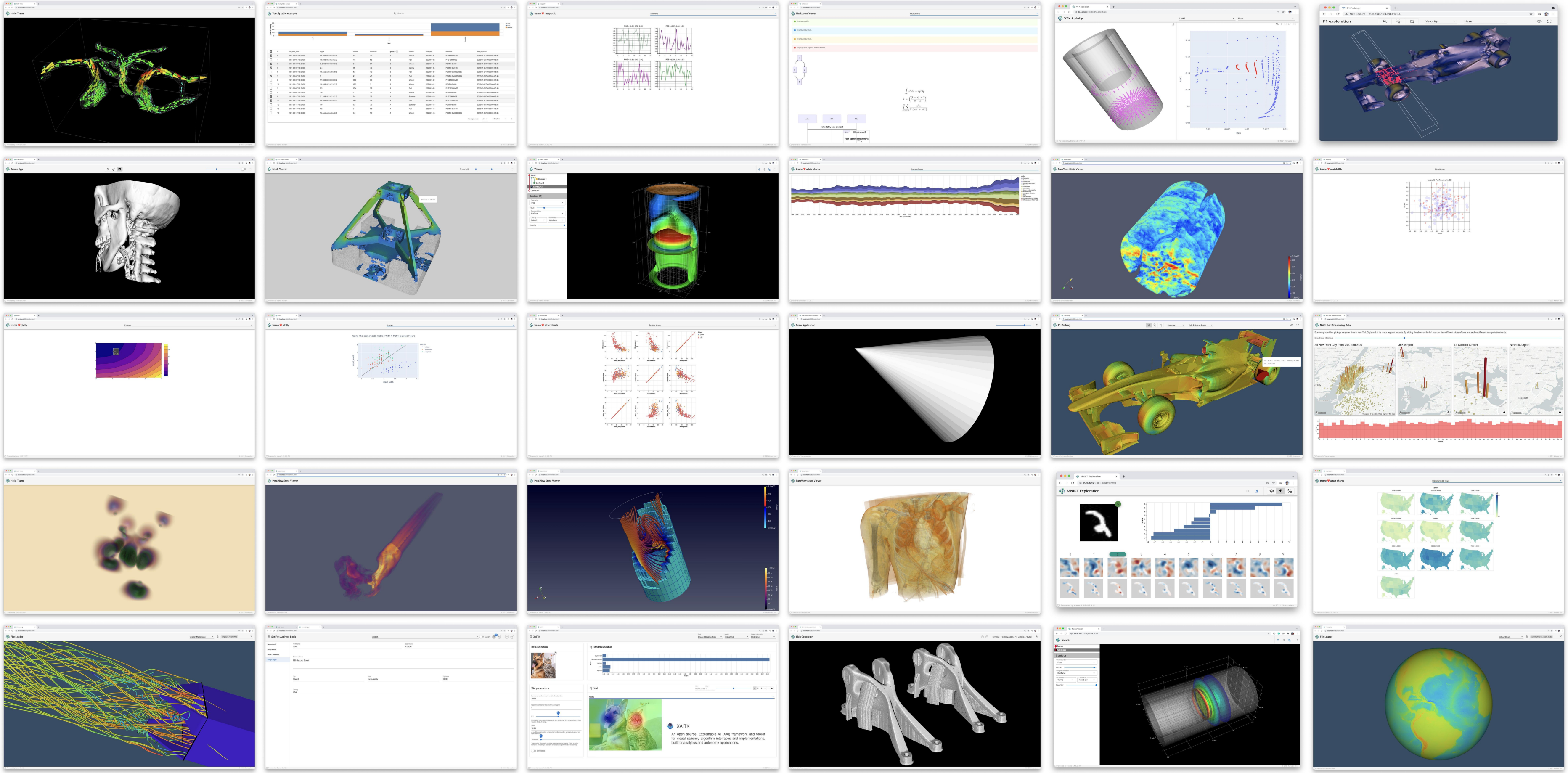


kitware

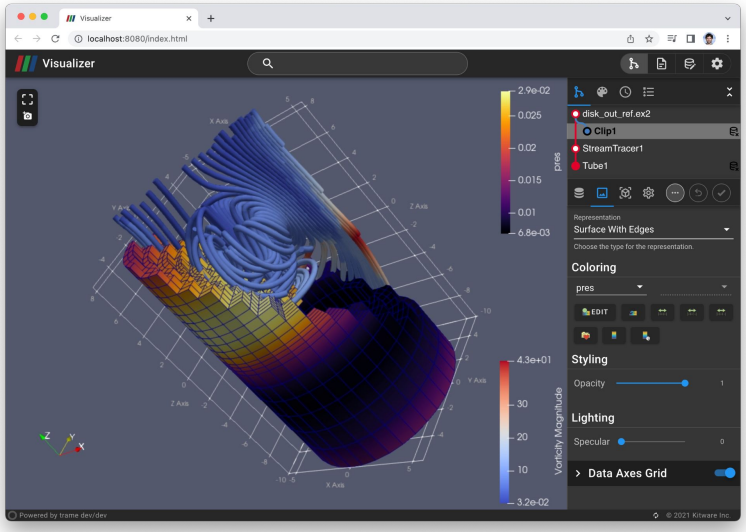
Multivariate View

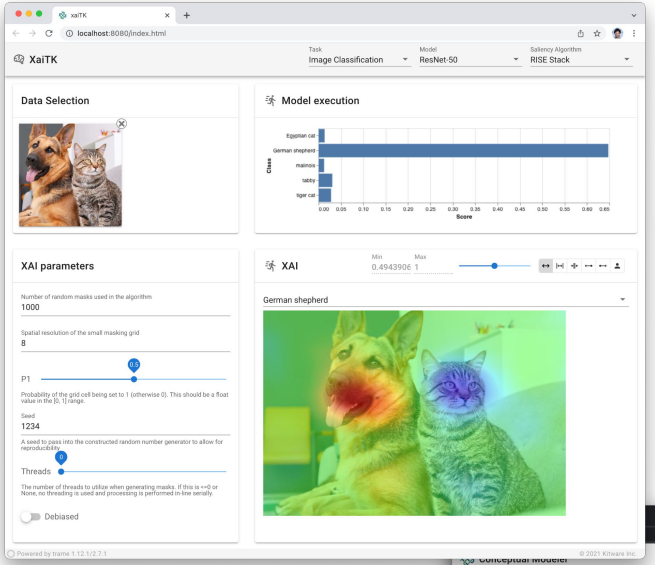


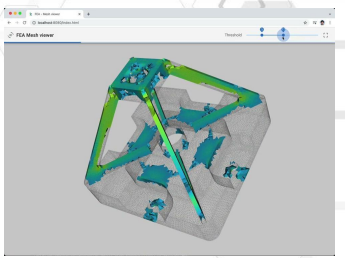
Many Examples

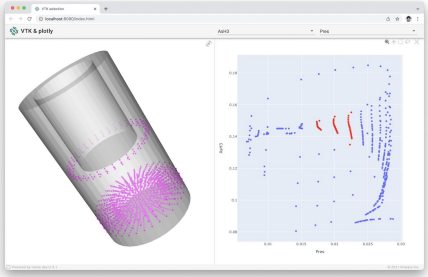


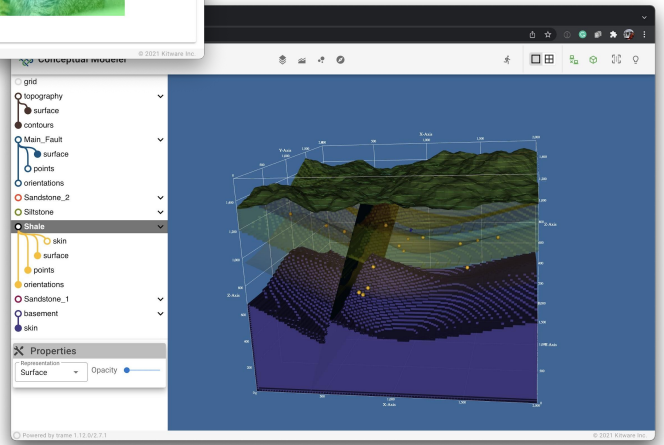
Many Applications



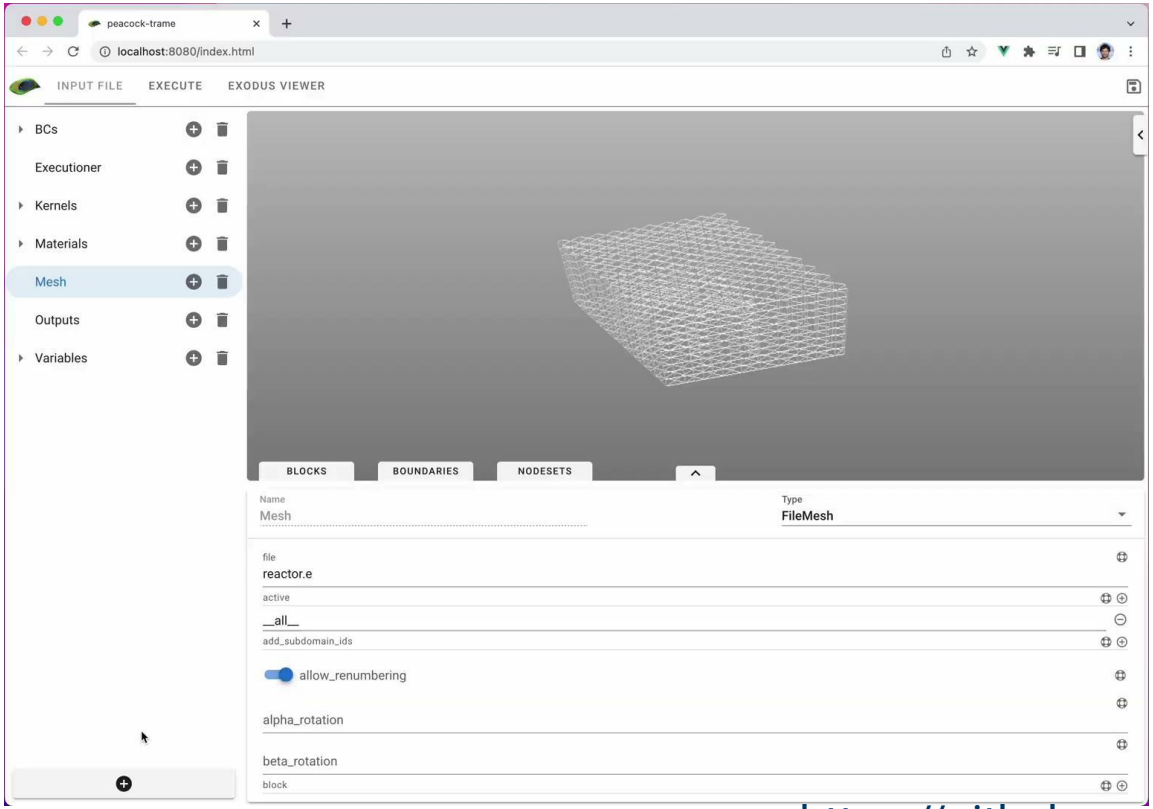








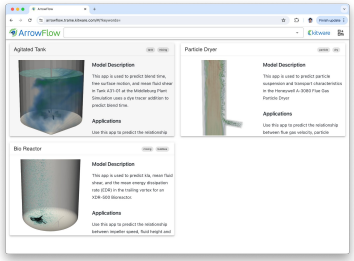
Peacock - Idaho National Laboratory

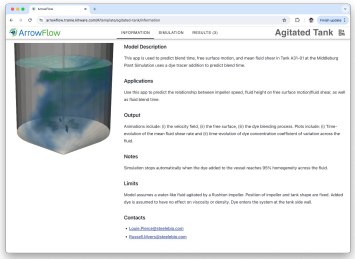


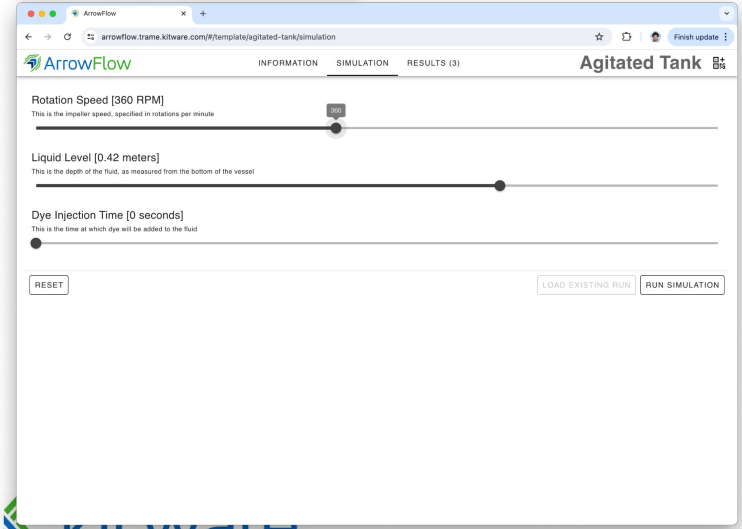
kitware

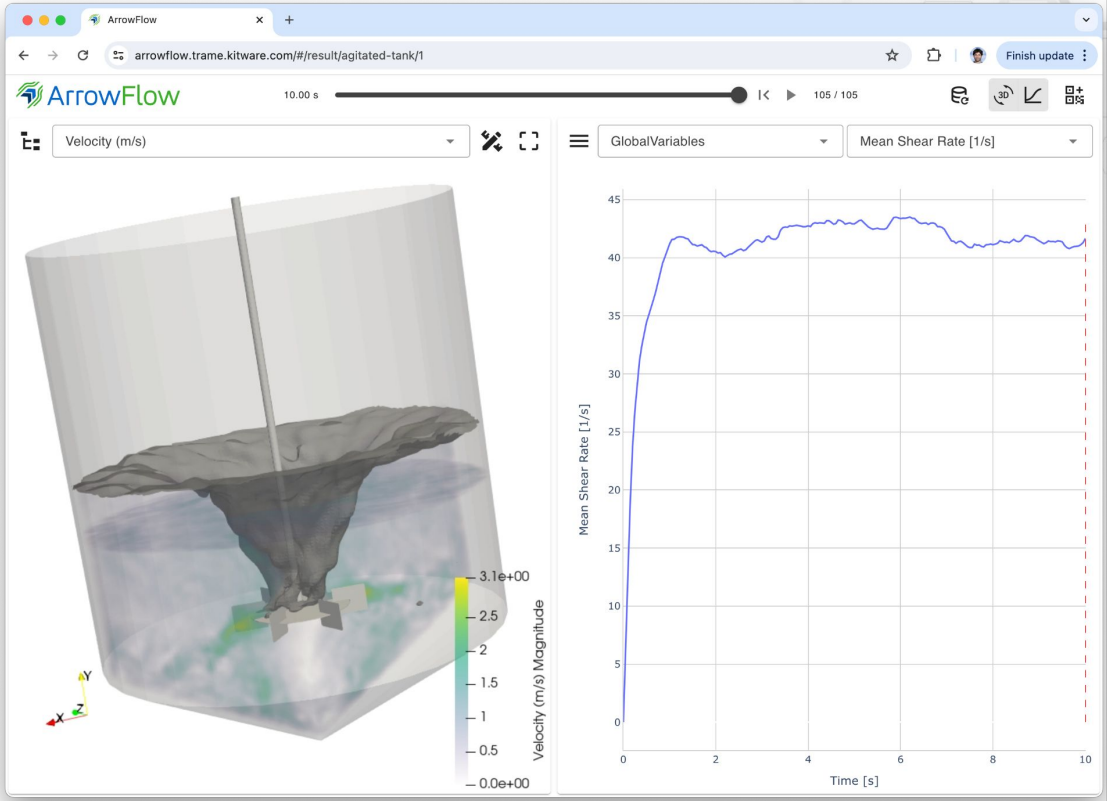
https://github.com/Kitware/peacock

ArrowFlow - Live and templated CFD workflow



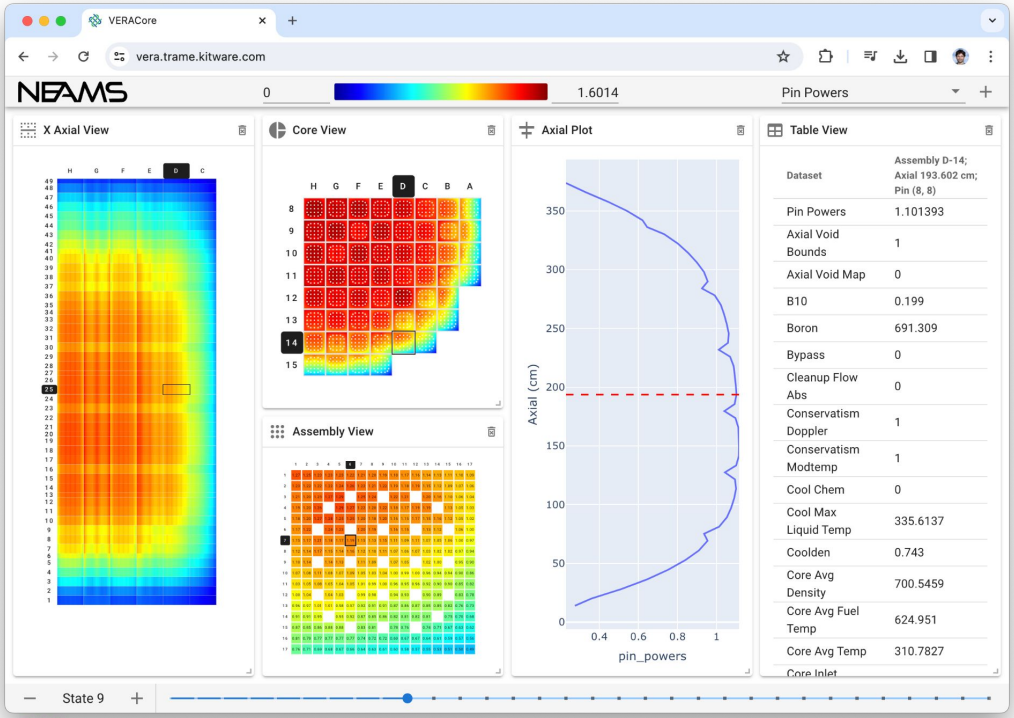


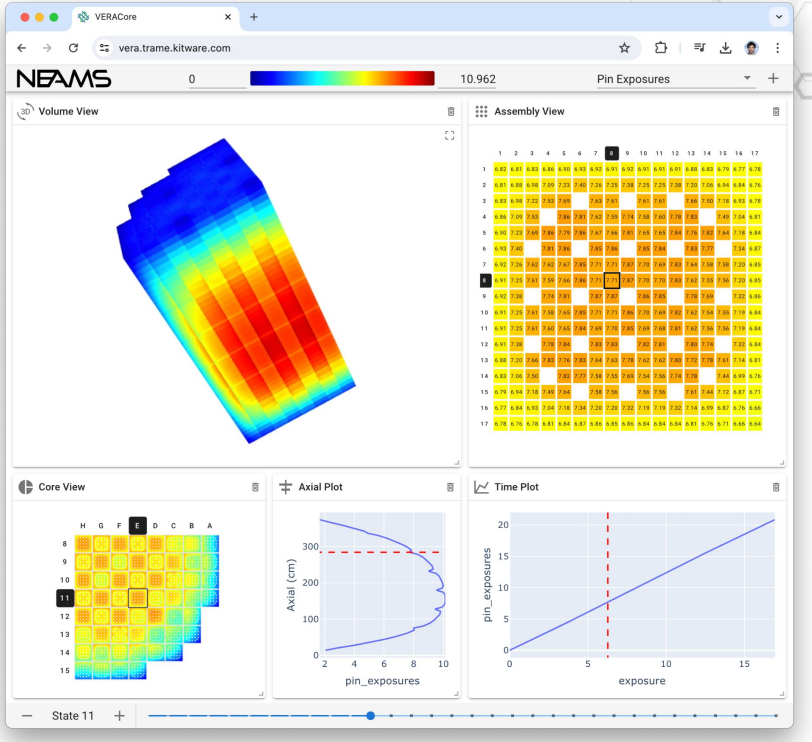




https://www.kitware.com/arrowflow/

Example: VeraCore - Simulation result exploration

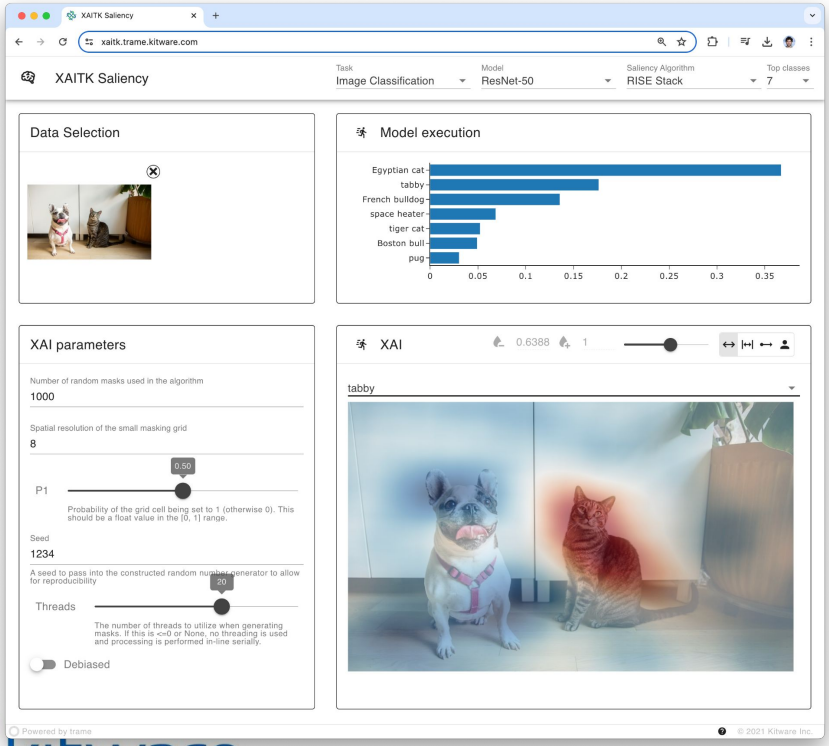


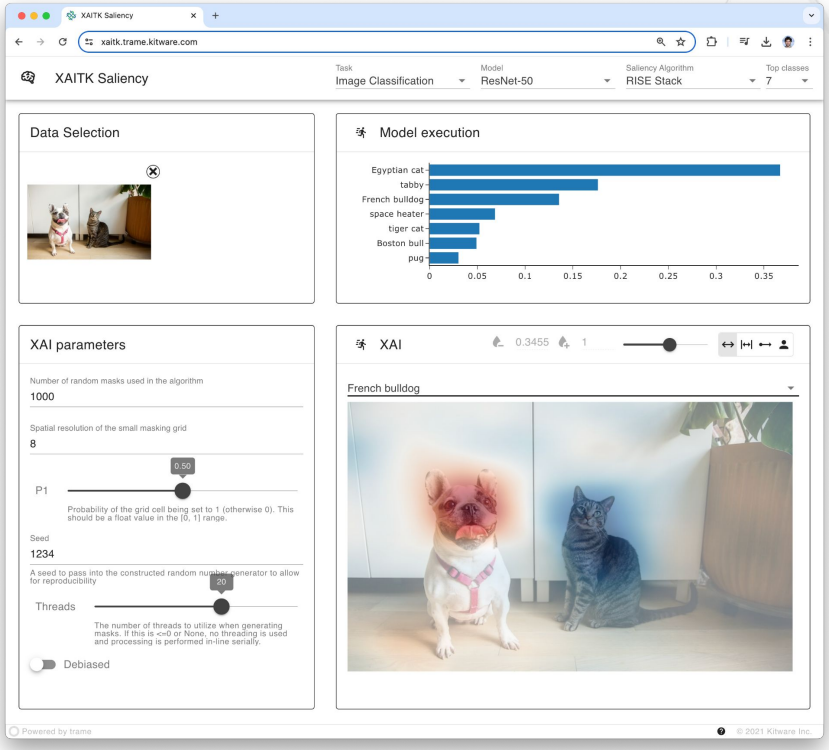


kitware

https://github.com/Kitware/veracore

Example: XAITK - Understanding complex machine learning models

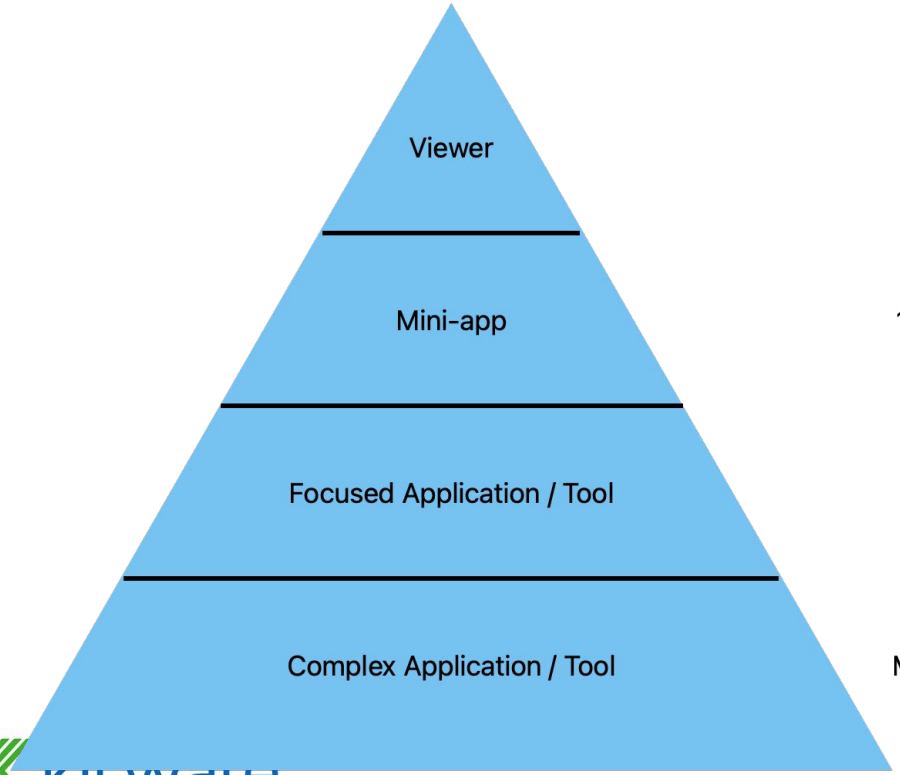




kitware

https://xaitk.org/

Usage examples (1/5)



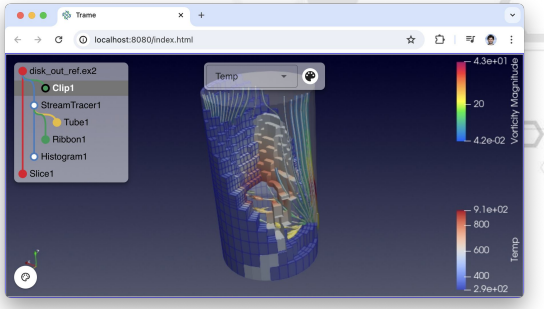
1 Python file ~ 10 lines

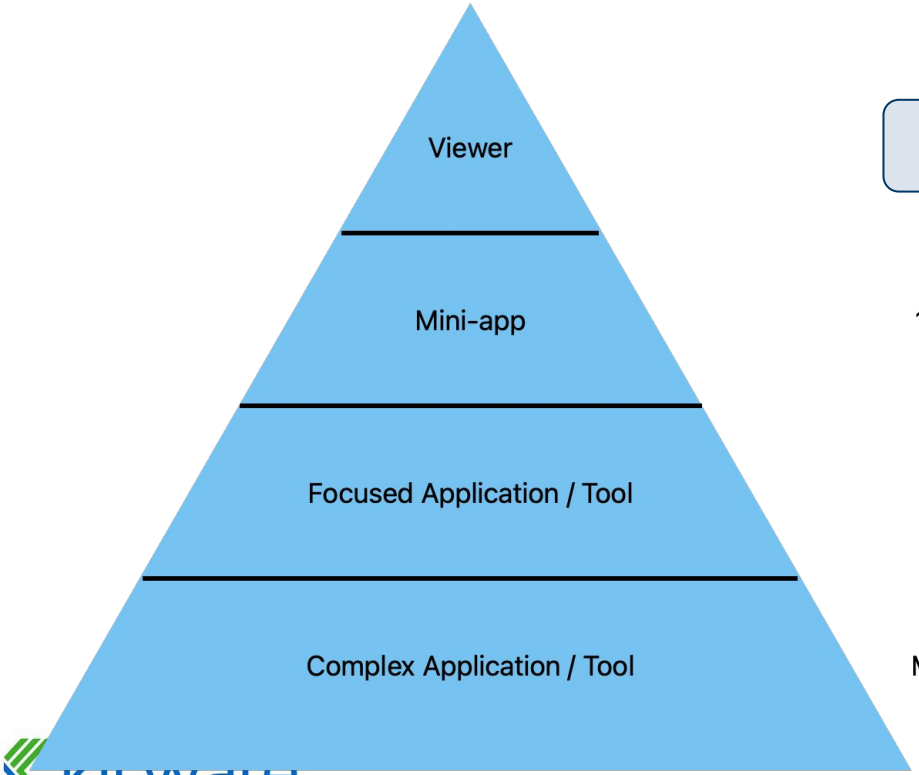
1 Python file ~ 500 lines

3-6 files ~ 1500 lines

Many files ~ +2000 lines

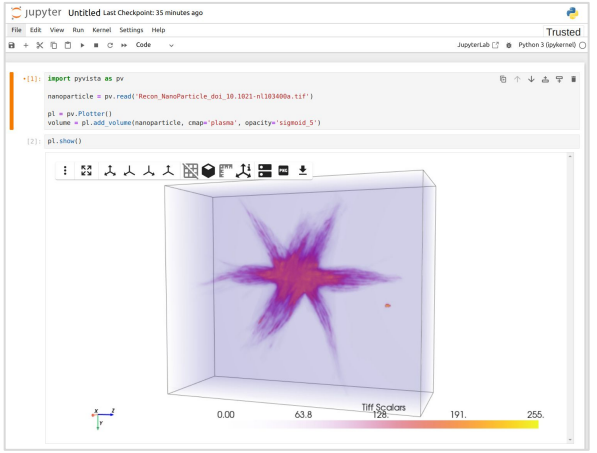
Usage examples (2/5)





1 Python file ~ 10 lines

1 Python file ~ 500 lines

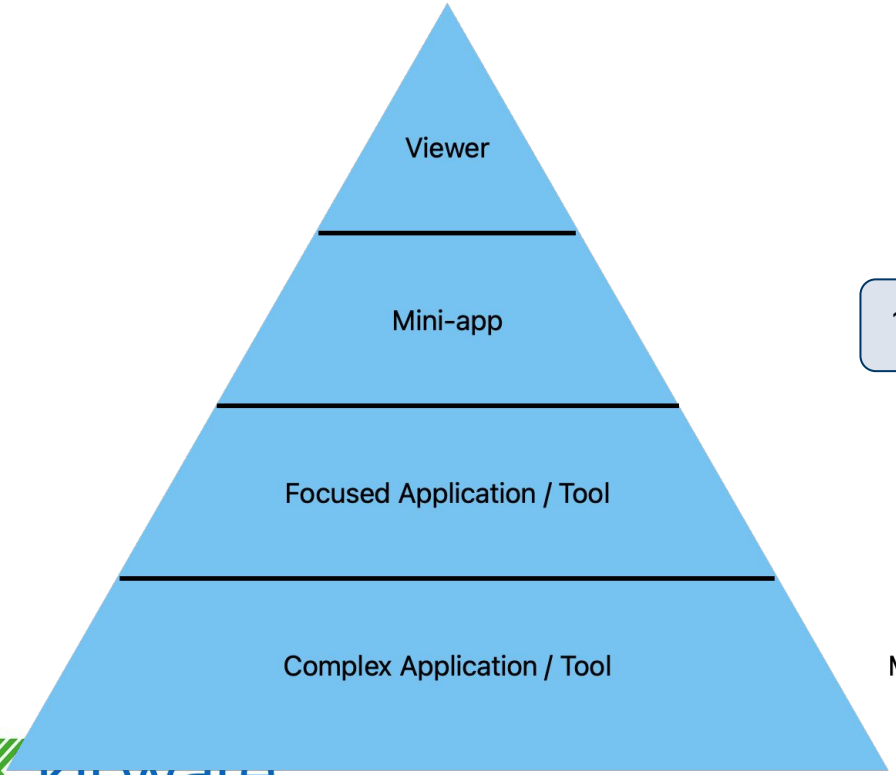


3-6 files ~ 1500 lines

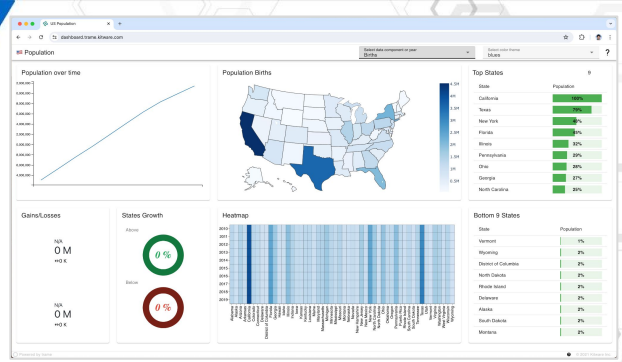
Many files ~ +2000 lines

kitware

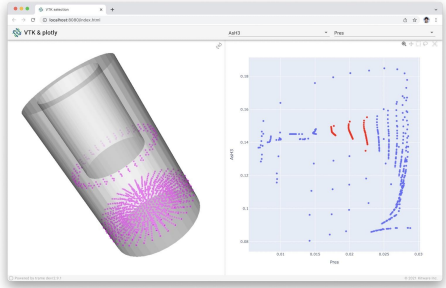
Usage examples (3/5)



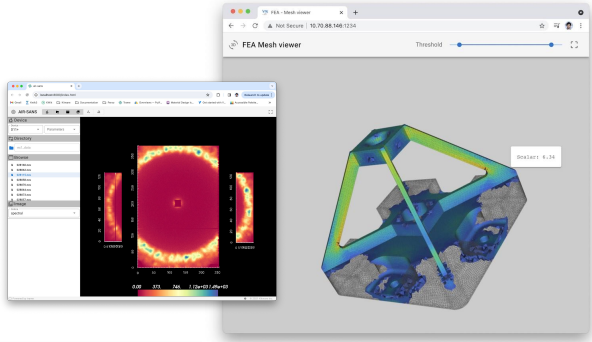
1 Python file ~ 10 lines



1 Python file ~ 500 lines



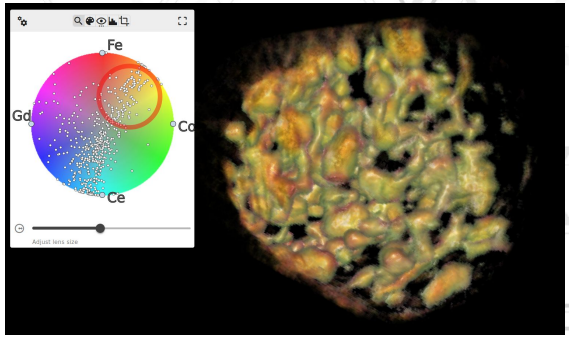
3-6 files ~ 1500 lines

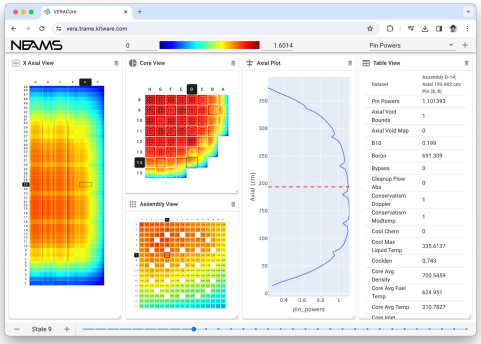


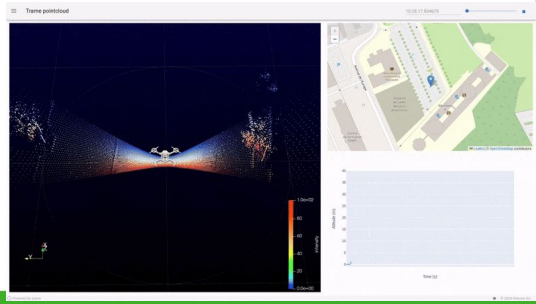
Many files ~ +2000 lines

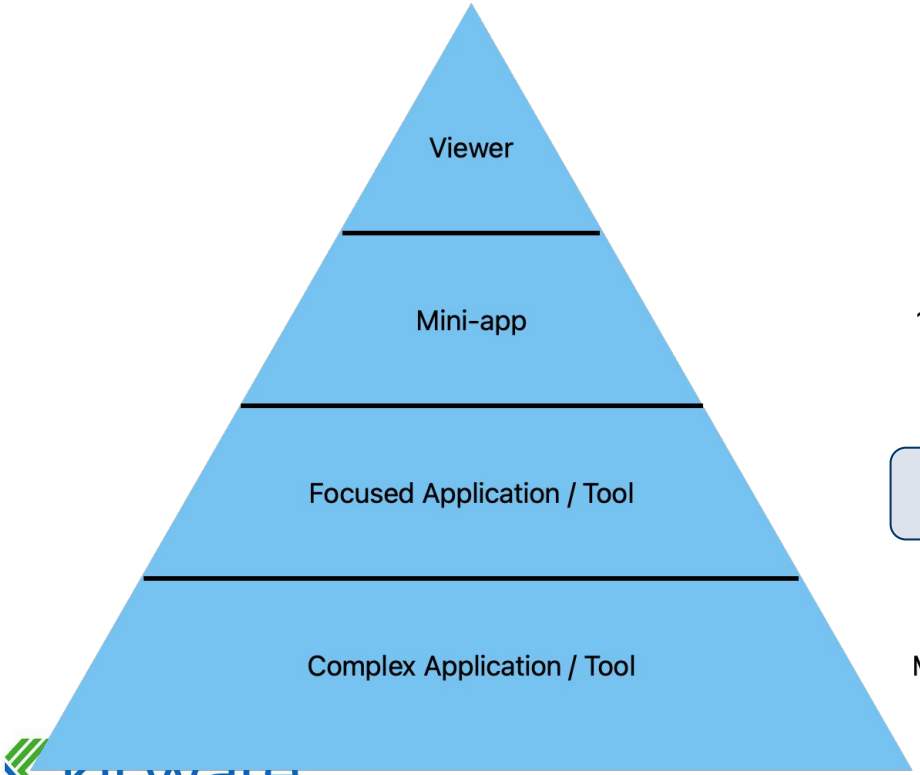
kitware

Usage examples (4/5)









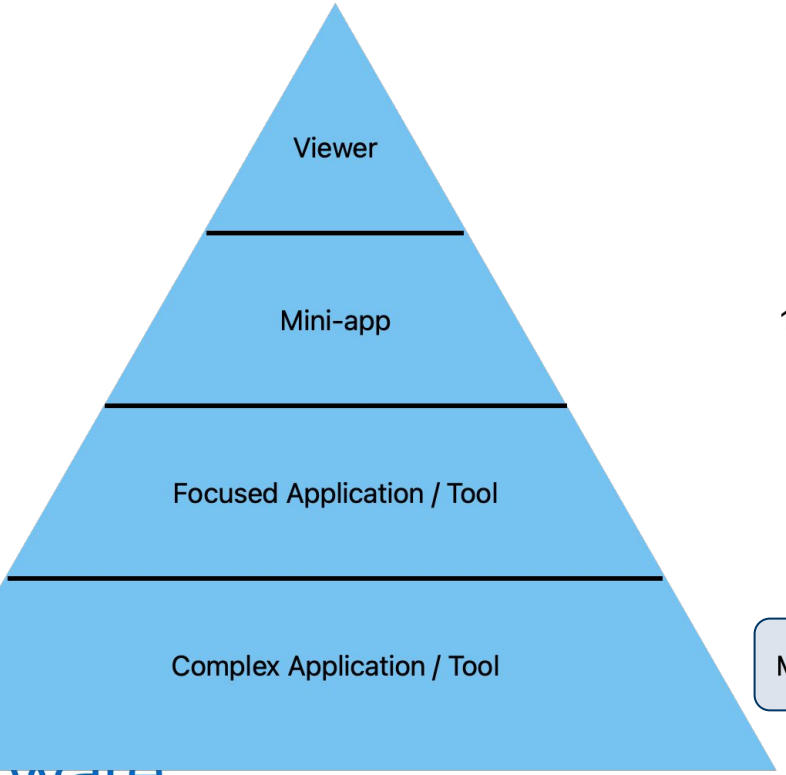
1 Python file ~ 10 lines

1 Python file ~ 500 lines

3-6 files ~ 1500 lines

Many files ~ +2000 lines

Usage examples (5/5)

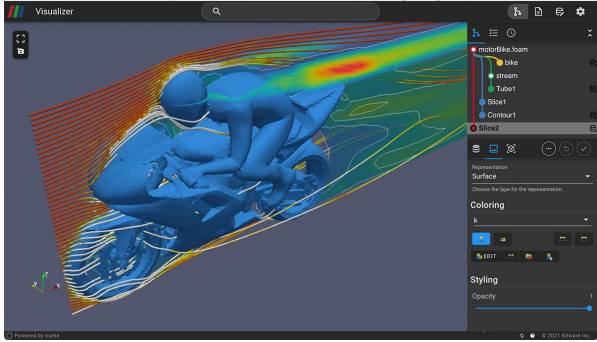


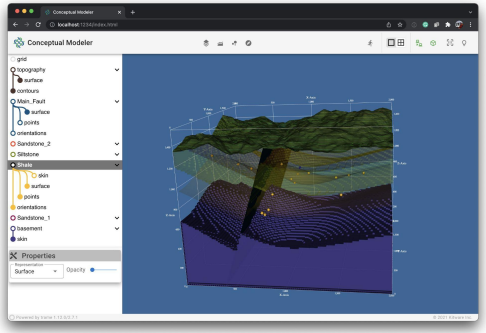
1 Python file ~ 10 lines

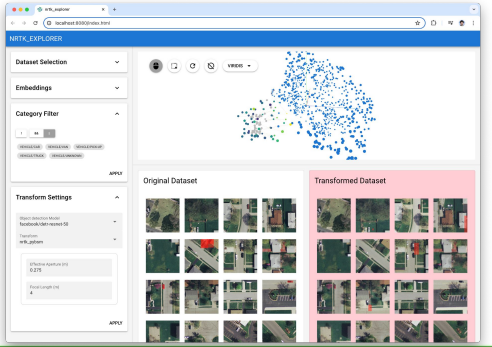
1 Python file ~ 500 lines

3-6 files ~ 1500 lines

Many files ~ +2000 lines







Nothing is too big or too small for trame

- From visualization to interactive data processing
- Running locally or on HPC
- Within Jupyter or in the cloud
- Trame has you covered

What you do with it is up to you…

kitware

It is like playing with legos…

kitware

Demo Time - Trame with ParaView

- **Create Python 3.10.13 conda environment**
  - Python 3.10.13 is needed for ParaView later

```
conda create -n trame-examples -y
conda activate trame-examples
conda install -y -c conda-forge python=3.10.13 \
  paraview-trame-components paraview=5.13.3
```

Alternatively, these can be installed with pip. These include dependencies needed for the ParaView example later.

kitware

Demo Time - Trame with ParaView (Pip only, external ParaView)

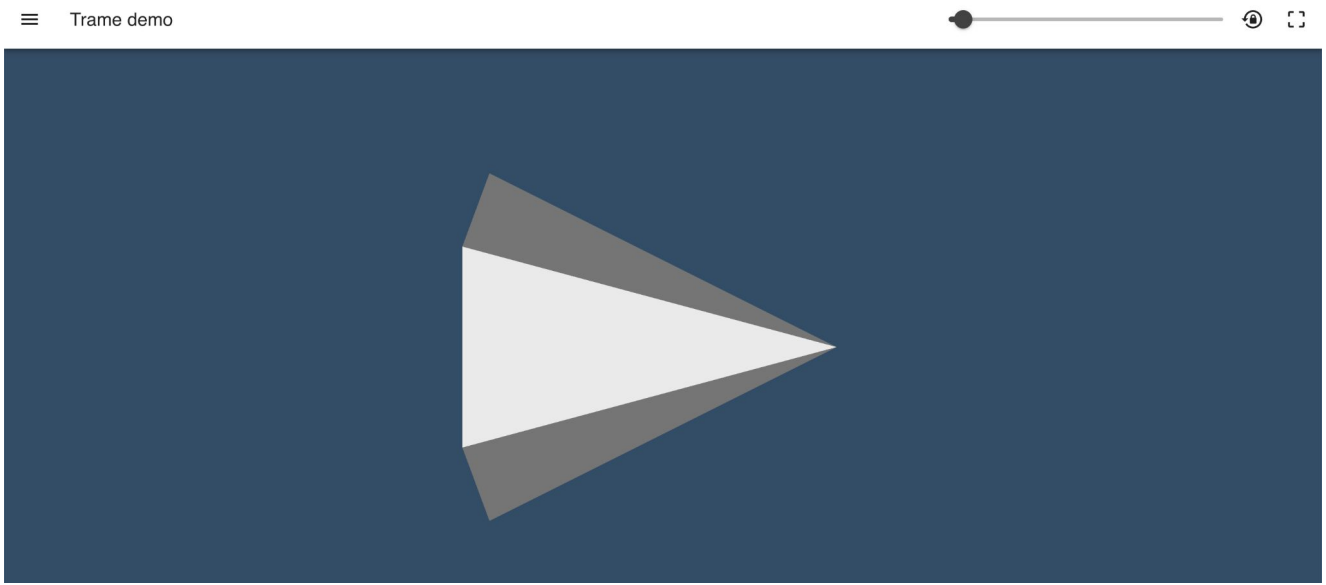
- **Create Python 3.10.13 virtual environment**
  - Python 3.10.13 is needed for ParaView later

```
pip install \
  trame trame-vtk trame-client trame-components trame-vuetify
```

These include dependencies needed for the ParaView example later.

Validate it is working

```
python -m trame.app.demo
```



kitware

Important: differences between remote and local rendering

- It’s very easy to switch between remote and local rendering in trame
- **Remote**: all rendering done on the server, images are streamed to the client. Mouse interactions are streamed from the client to the server.
- **Local**: geometry/data is sent from the server to the client. All rendering and interactions are done on the client.
  - Local rendering is usually done with VTK.js. It may not contain all components.
  - When VTK-WASM is ready (getting close), that will be used for local rendering, and all components will be available.

Tricks Needed for ParaView

- **Using ParaView from conda-forge**
  - Everything works out of the box.

```
# conda install -c conda-forge paraview trame ...
# Then run your trame apps that use ParaView
```

- **Using pre-installed ParaView**
  - ParaView's pvpython does not come with network capabilities (pip and OpenSSL), making it difficult to install new Python packages
    - This is partly so ParaView can be easily integrated in systems with security concerns
  - Often need to merge our venv with ParaView's venv
    - Requires **identical** version of Python in our venv

ParaView Demo Time

- Ensure your virtual environment is activated
- Download these ParaView state file examples
- Unzip them somewhere and remember location
- Clone trame. We will run an example script in it.

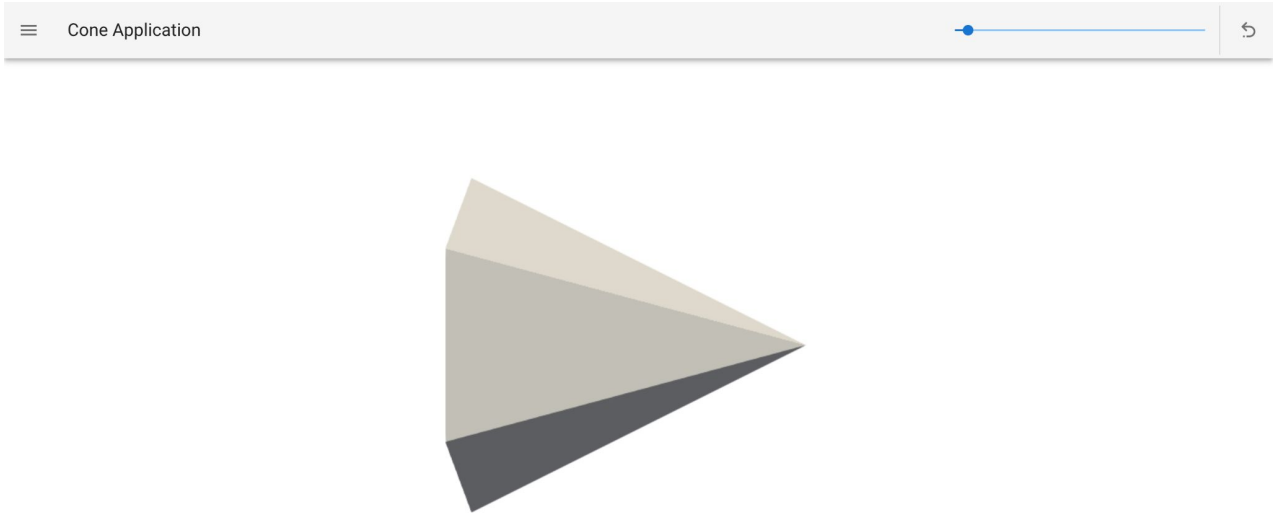
```
git clone https://github.com/kitware/trame
cd trame
```

kitware

ParaView Examples - Cone

```
python ./examples/07_paraview/SimpleCone/RemoteRendering.py
```

RemoteRendering.py Link

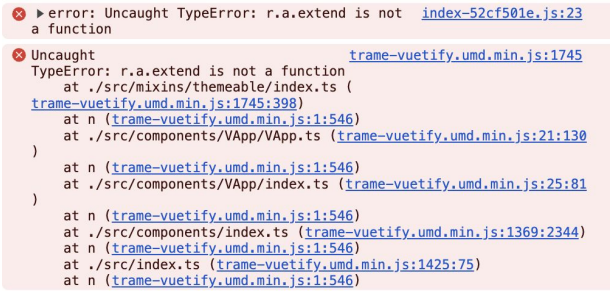


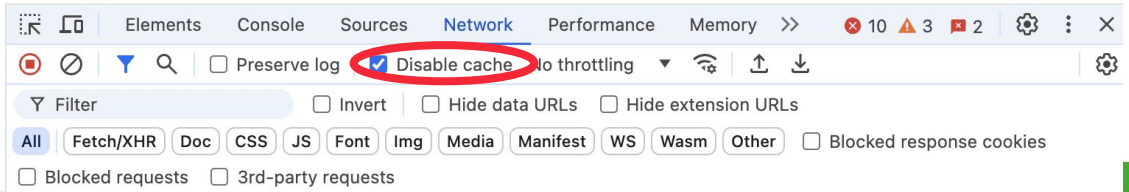
kitware

Caching Issues?

- **Modern web browsers do a lot of caching (for improved performance)**
- **You may occasionally get pages with issues, such as failure to load, due to caching issues, especially when switching between Vue2 and Vue3 trame apps.**
- **To fix: open dev tools, go to “Network”, check “Disable cache”, and reload web page**
  - Note that “Disable cache” only works while dev tools are open. If you close the dev tools, the cache will not be disabled. You must open dev tools and reload again.

Loading...



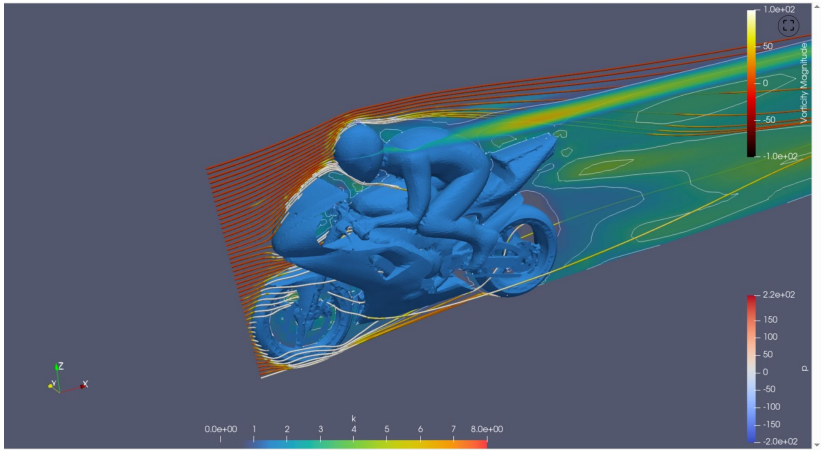


kitware

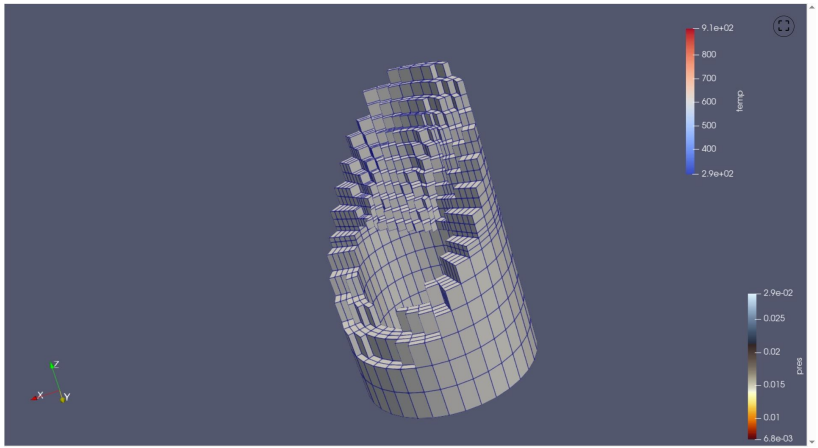
ParaView Examples - State Viewer

```
ptc-state --state $HOME/Downloads/pv-state/<file>.pvsm
```

motobike2.pvsm



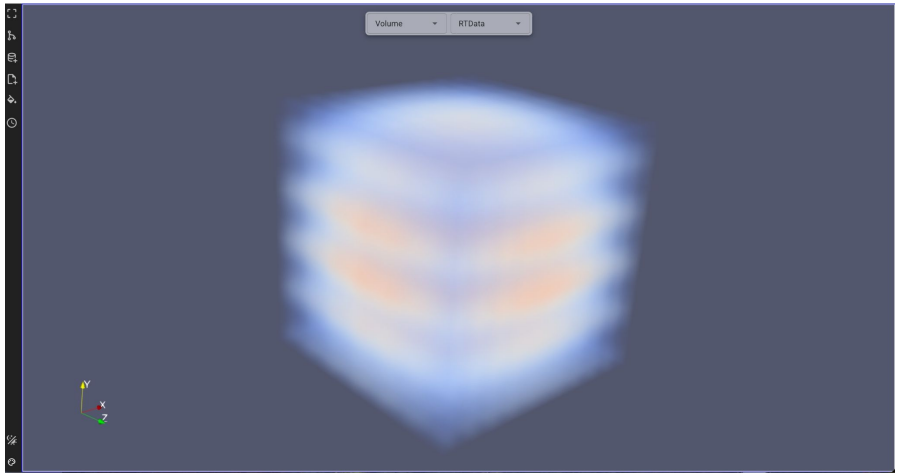
diskout/diskout2.pvsm

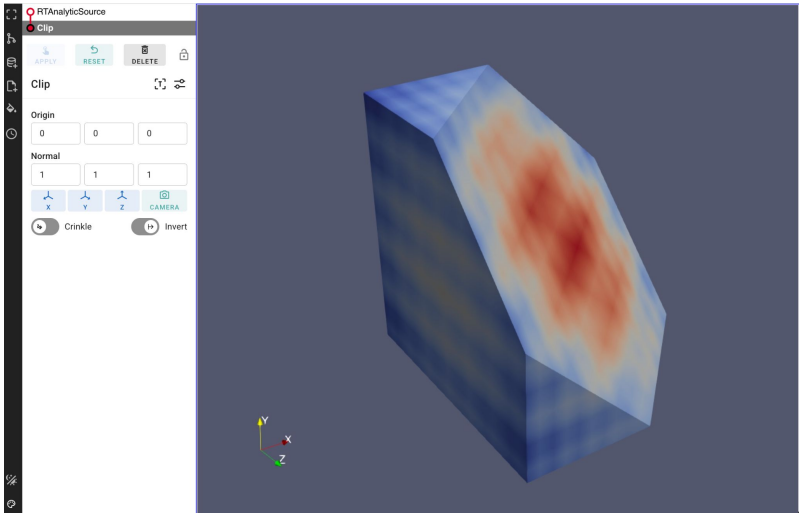


kitware

ParaView Trame Components - control ParaView with trame!

```
ptc-lite
```





kitware https://github.com/kitware/paraview-trame-components

Using external ParaView

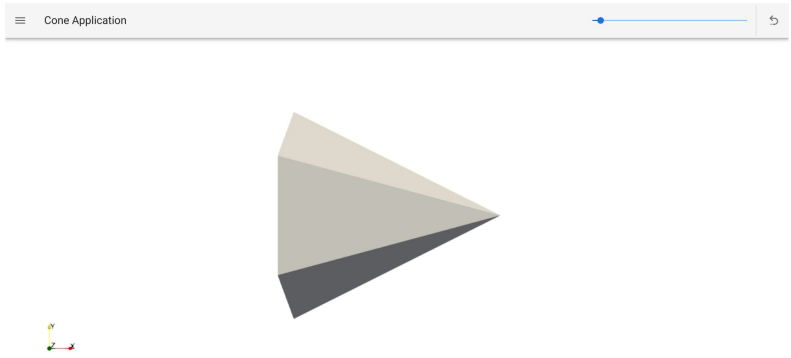
This was for my Mac

```
# First, be sure to remove ParaView from the conda environment
conda remove paraview --force

/Applications/ParaView-5.13.3.app/Contents/bin/pvpython \
  ./examples/07_paraview/SimpleCone/RemoteRendering.py --venv $CONDA_PREFIX
```

RemoteRendering.py Link

- Use the path to pvpython on your system.
- On Windows, use %CONDA_PREFIX%



kitware

Running trame on HPC

- **ParaView handles the parallelism (i. e., pvbatch)**
- **Need to connect to the root node from web browser**
- **Might require a little network configuration/tricks for each HPC cluster**
  - For example, a reversed connection
  - Sebastien can handle it…
    - Infrastructure is available to set it up on any cluster
- **Already being done at Sandia**

New Developments

- **WASM for client-side rendering**
  - More robust local rendering (uses the same VTK code locally)
  - Access to more VTK components, including 3D widgets (whatever is serializable in VTK)
  - More user-friendly API for interacting with objects
  - Better performance through access to WebGPU (coming soon)

Check out the website and the many examples available

- **Website with Great Examples: kitware.github.io/trame**

Live Trame Examples
(might be slow if everyone runs them at once)

- **VERACore: vera.trame.kitware.com**
- **Visualizer: visualizer.trame.kitware.com**
- **ArrowFlow: arrowflow.trame.kitware.com**
- **XAITK: xaitk.trame.kitware.com**
  - Example File

kitware

Kitware Services

Training

Learn how to confidently use trame from the expert developers at Kitware.

GET STARTED

Support

Our experts can assist your team as you build your Web application and establish in-house expertise.

GET SUPPORT

Custom Development

Leverage Kitware's 25+ years of experience to quickly build your Web application.

REQUEST A QUOTE

kitware

Questions?

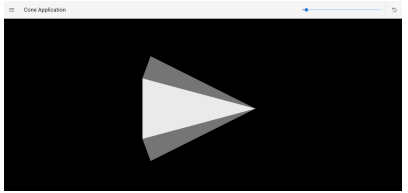
Thank you

kitware

VTK Demo

- **Do this in a separate conda environment than ParaView**
  - The conda VTK may collide with ParaView's VTK
- **First, install VTK**

```
conda install -c conda-forge 'vtk>=9.3'
```



- **Second, run VTK example scripts in trame repo**

```
python ./examples/06_vtk/01_SimpleCone/RemoteRendering.py
```