Web-based Visualization: Overview of Client and Server-side Techniques and their Use in GeoJS and CDATWeb

Aashish Chaudhary
Technical Leader, Kitware Inc
with
Matthew Harris, Jonathan Beezley, Chris Harris,
Thomas Maxwell, Charles Doutriaux, Samuel Fries, John Harney

Overview

Rendering Modes

- Client side
- Server side
- Mixed

Tools

- ParaViewWeb / vtkWeb
- Cinema
- GeoJS

CDATWeb

Conclusion

Client Side Visualization

SVG: D3, snap.svg, Raphaël

Supports basic shapes, paths, and polygons

Native support for mouse events and CSS

Too slow for large numbers of elements

Canvas: Paper, Fabric, Cinema, Bokeh

Fast pixel level access for 2D graphics

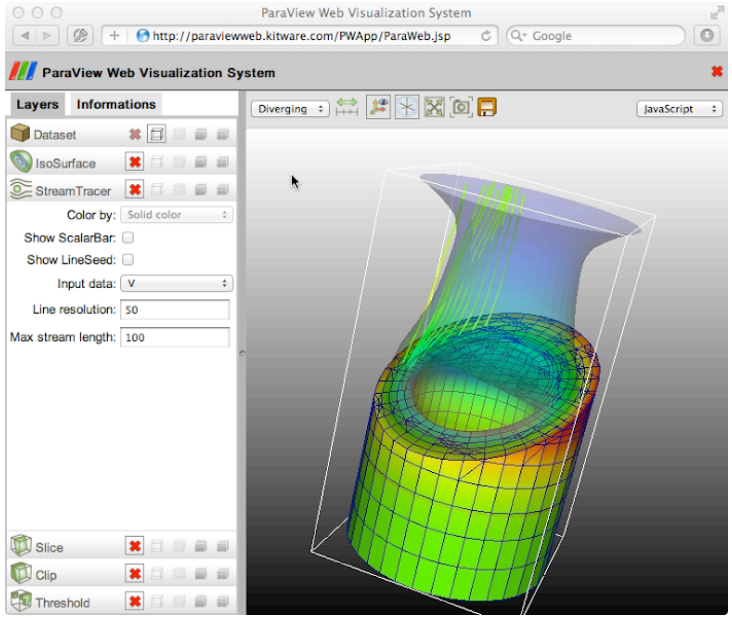
No native 3D support

WebGL: VGL, Three

Fast, GPU accelerated 3D graphics

Not as widely supported among browsers/devices

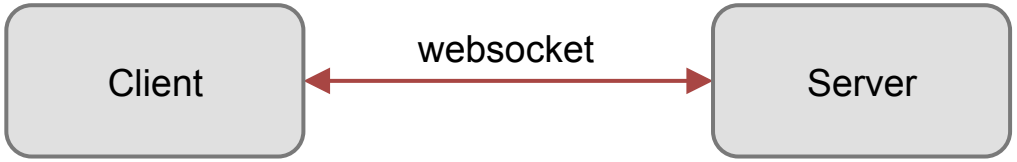
Server Side Visualization



- Renders images on the server and streams them to the client.
- Requires large server infrastructure and may not scale well for large numbers of users on a typical web-server (memory or processing limitations).
- Ideal for serving the visualization using existing code base
- Not ideal for pre-computed analysis or visualization

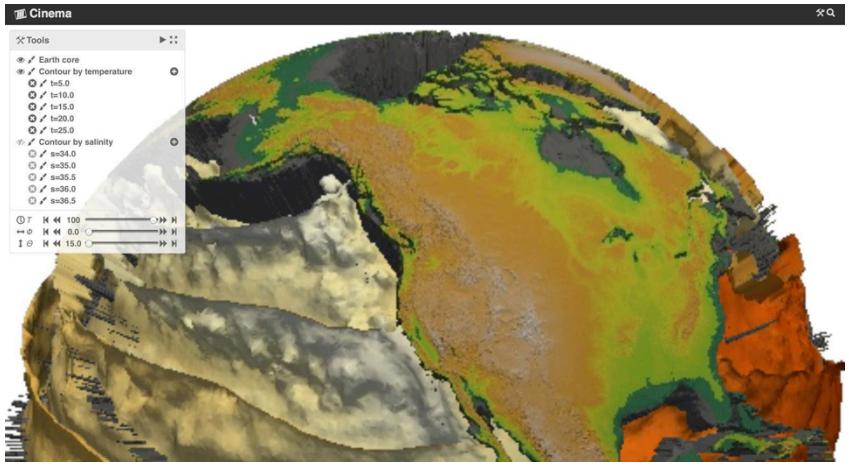
ParaViewWeb / vtkWeb

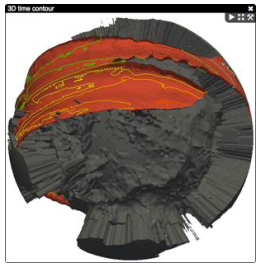
Uses Autobahn to stream data between the client browser and the server ParaView instance. Communication is transmitted via Remote Procedure Calls (RPC) over websockets to minimize latency.

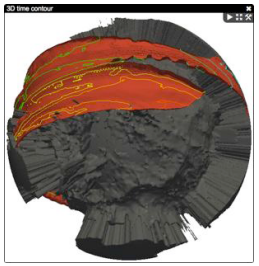


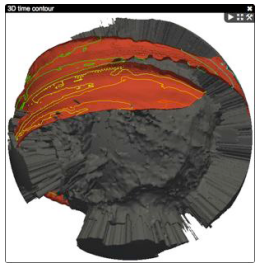
- Clients can remotely call ParaView python methods on the server.
- The server renders the scene and transmits data back to the client.
- User interactions are seamlessly passed from client to server.
- The ParaViewWeb server can stream results as images or GL primitives to be rendered with VGL.

Mixed Mode

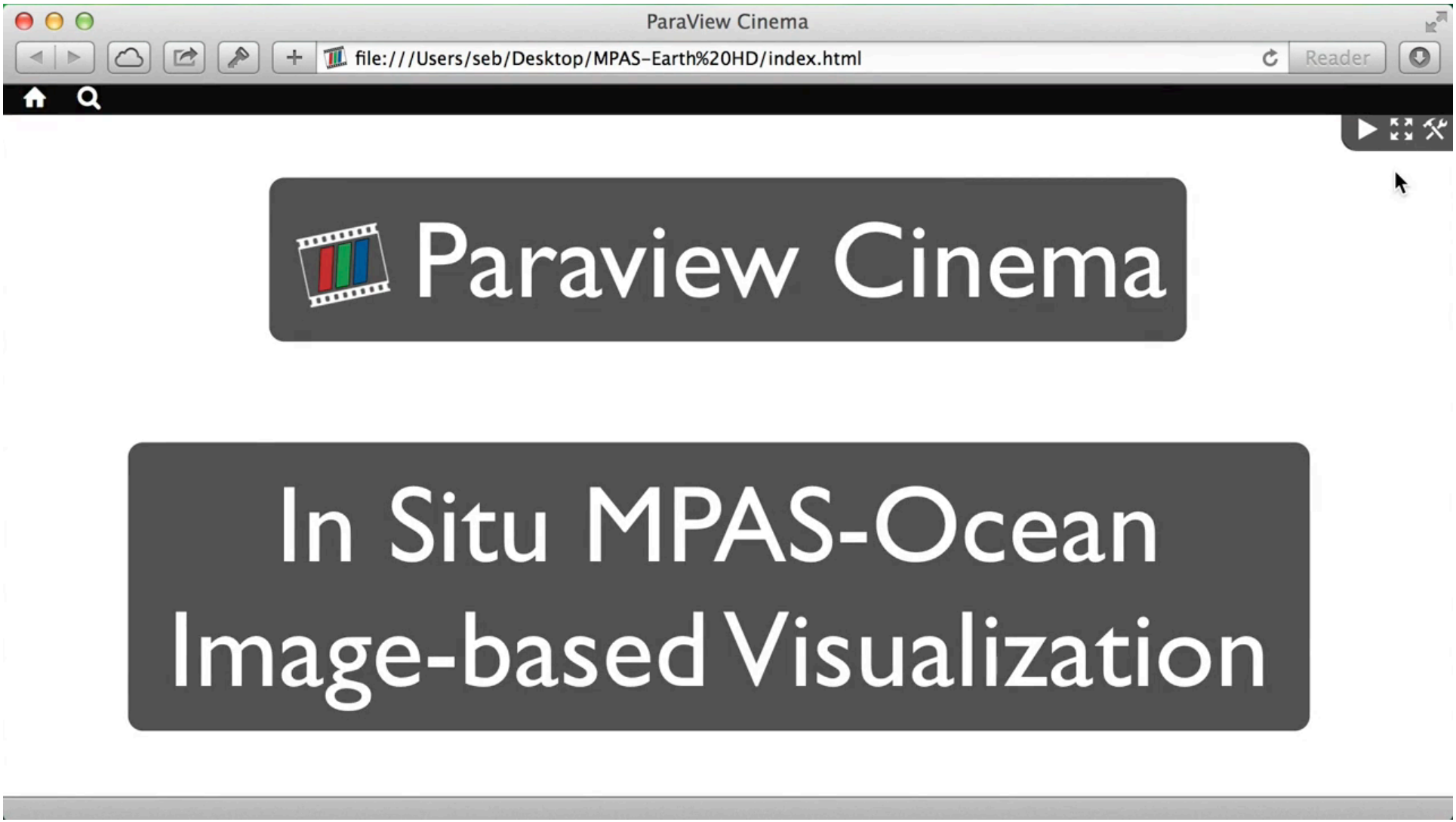


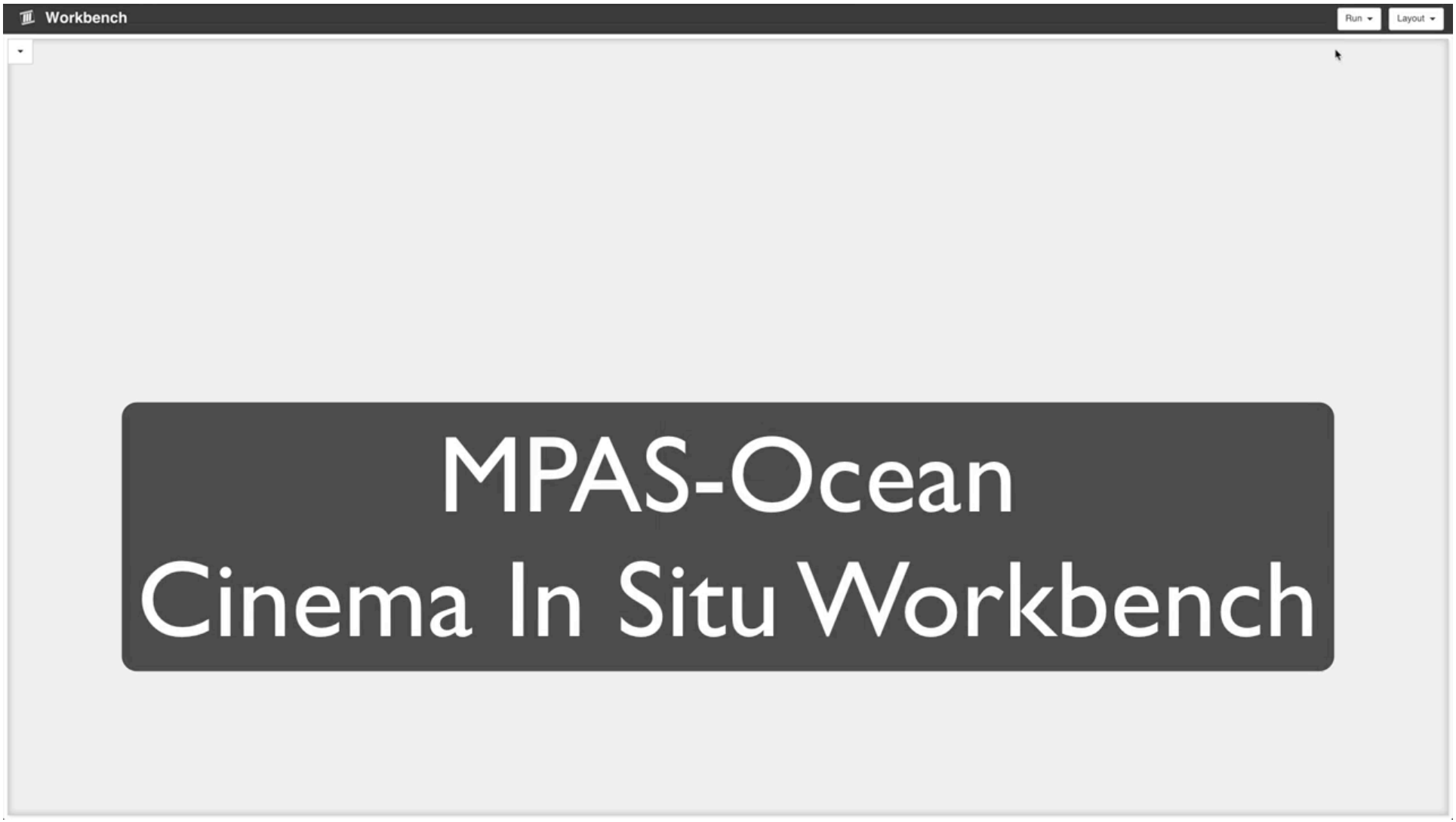






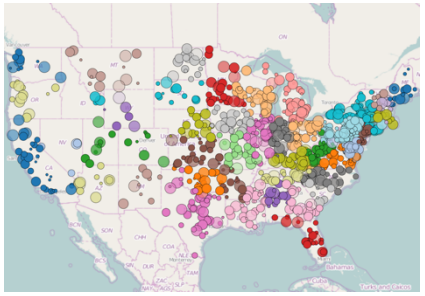
- Images are renderers on the server and stored as data arrays
- Fixed viewpoints
- Composition is done on the client side using WebGL or Canvas2D
- Visualization using image database

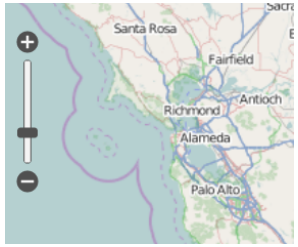

ParaView Cinema
file:///Users/seb/Desktop/MPAS-Earth%20HD/index.html
Reader
Paraview Cinema
In Situ MPAS-Ocean
Image-based Visualization

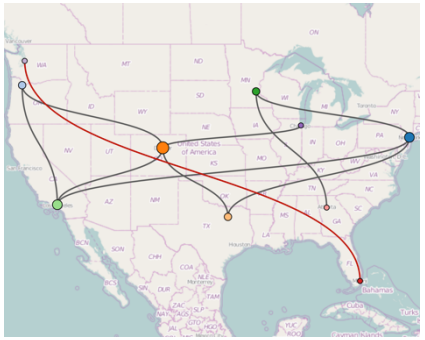

Workbench
Run
Layout
MPAS-Ocean
Cinema In Situ Workbench

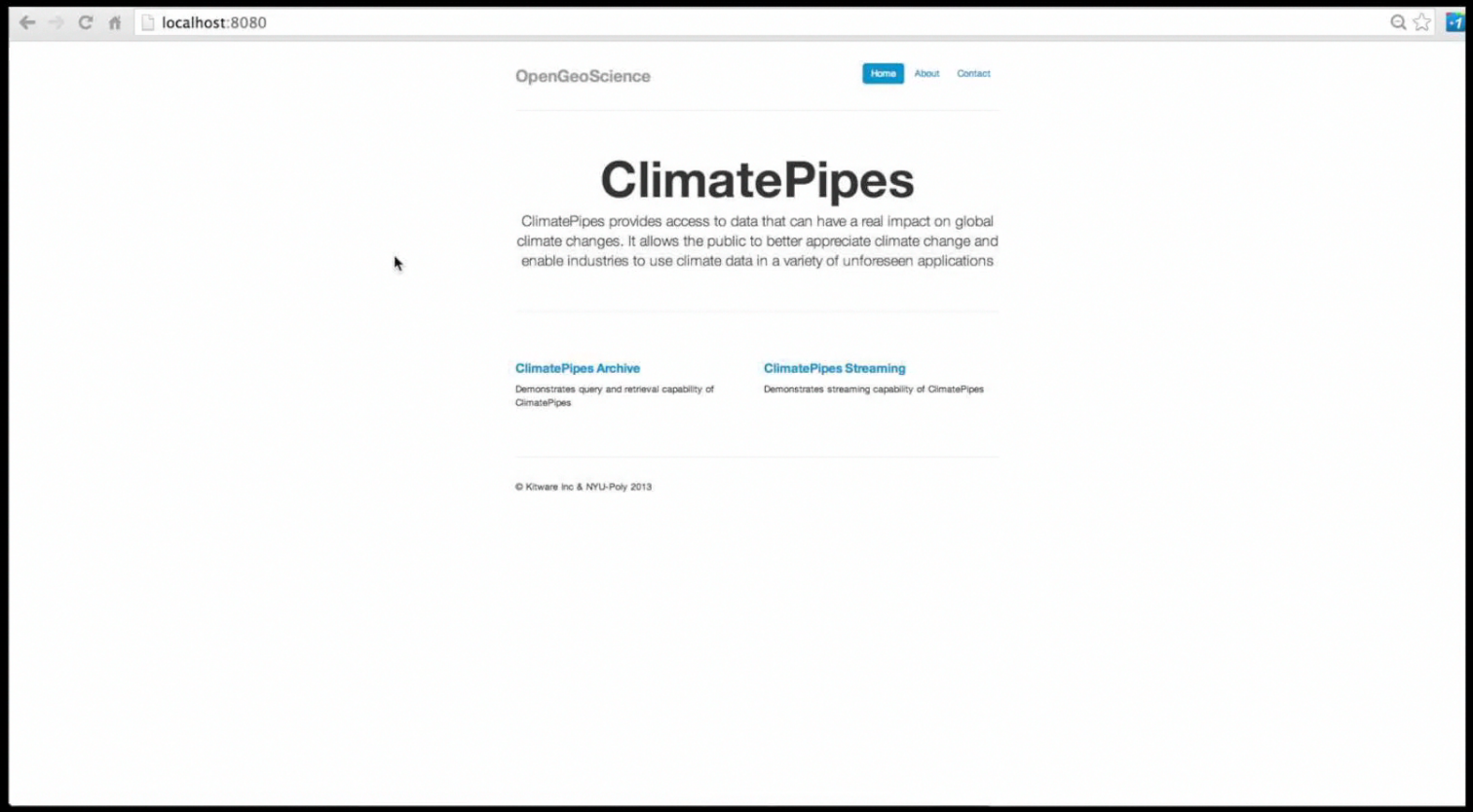
GeoJS: A geospatial visualization and mapping framework

- Combine GL, Canvas, and SVG graphics into a single visualization
- Unified API for generating features in SVG or GL
- Designed to perform well with large datasets
- Features provide DOM-like mouse events
- Provides low level access to underlying APIs, i.e. SVG layers expose D3 selections that are synced to the map
- Actively developed and maintained with new features added almost daily
- Supports scientific visualization, geo-visualization, GIS, and infovis








localhost:8080
OpenGeoScience
Home
About
Contact
ClimatePipes
ClimatePipes provides access to data that can have a real impact on global climate changes. It allows the public to better appreciate climate change and enable industries to use climate data in a variety of unforeseen applications
ClimatePipes Archive
Demonstrates query and retrieval capability of ClimatePipes
ClimatePipes Streaming
Demonstrates streaming capability of ClimatePipes
© Kitware Inc & NYU-Poly 2013

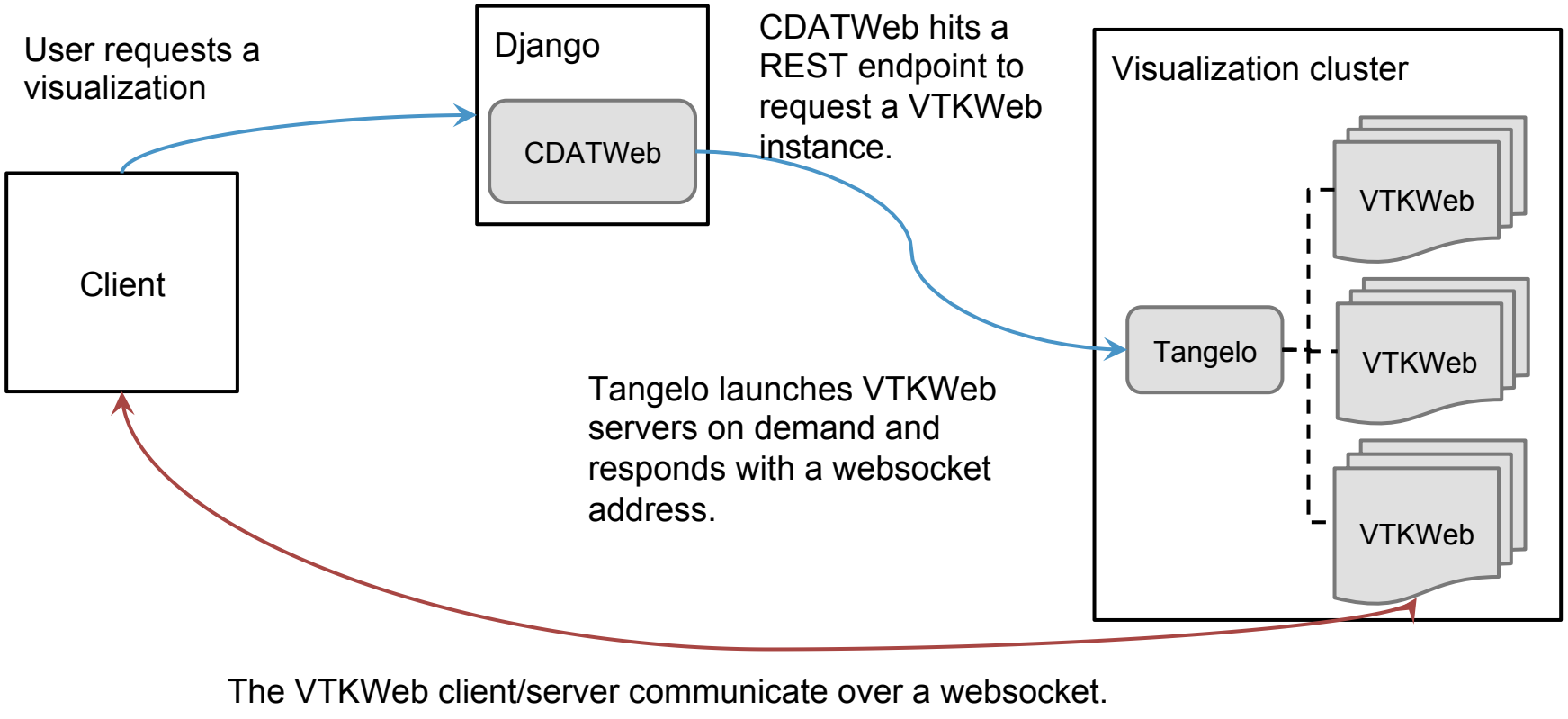
GeoJS: Feature API

```
layer.createFeature('point', options)
  .data(data)
    .position(function (d) { return {x: d.x, y: d.y}; })
    .style('fillColor', function (d) { return d.color; })
    .style('fillOpacity', function (d) { return d.opacity; })
    .style('strokeColor', 'black')
    .geoOn(geo.event.feature.mouseover, function (evt) {
      evt.data.opacity = 0.75;
      evt.data.color = 'red';
      this.modified();
      this.draw();
    })
    .geoOn(geo.event.feature.mouseout, function (evt) {
      evt.data.opacity = 0.25;
      evt.data.color = 'steelblue';
      this.modified();
      this.draw();
    })
    .draw();
```

All feature types use a flexible data binding API making it easy to customize appearance and behavior per marker

- Similar to D3
- Works for both SVG and GL
- Mouse events can propagate to features in all layers (not just the top)

CDATWeb Visualization Components



Final Thoughts

- Web visualization is now ready for production
- Use of web visualization for exploratory data analysis (EDA) or scientific purposes is emerging
- Flexibility is good: Ability to switch between client, server or mixed modes
- Web visualization tools for scientific datasets are in scarcity