Python and JavaScript

Visualizing Data in the Browser

Chris Mueller

PyData New York 2012

Lab7 Systems, Inc. ▪ Austin, TX

www.lab7.io

How We Share Results

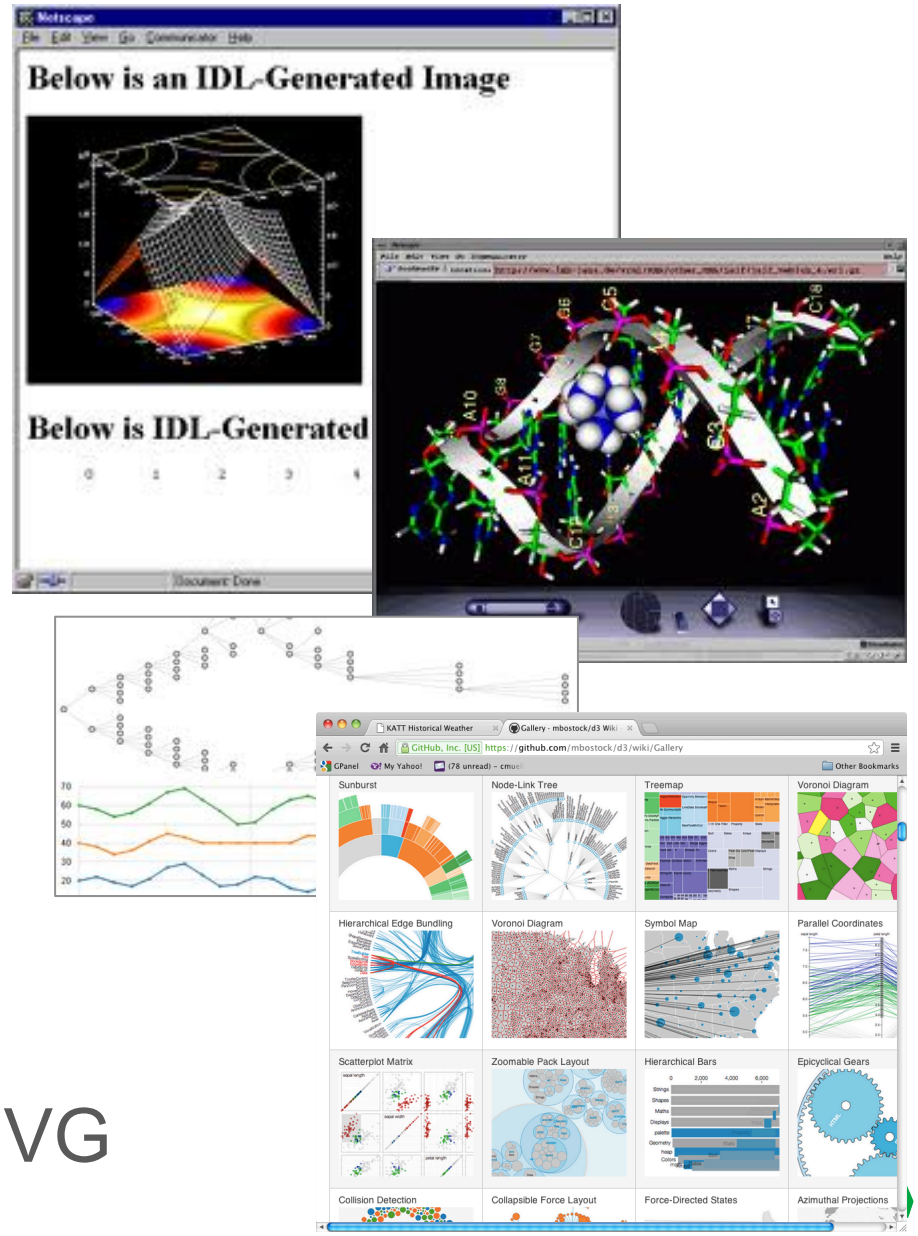
"Eventually, we need to share our work with others."
- Dan Williams, Data Scientist at Life Technologies

- Analysis does not happen in a vacuum
- Best practices for sharing data
  - Excel
  - Word
  - Power Point
  - Images
  - "Run this script…"

Visualization in the Browser

Interactive graphics have always been an afterthought in Web browsers.

- Back in the day…
  - Java
  - VRML
  - Plug-ins
  - GIF, JPG, PNG
- The Aughties: Flash
- Today
  - HTML5: Canvas + SVG



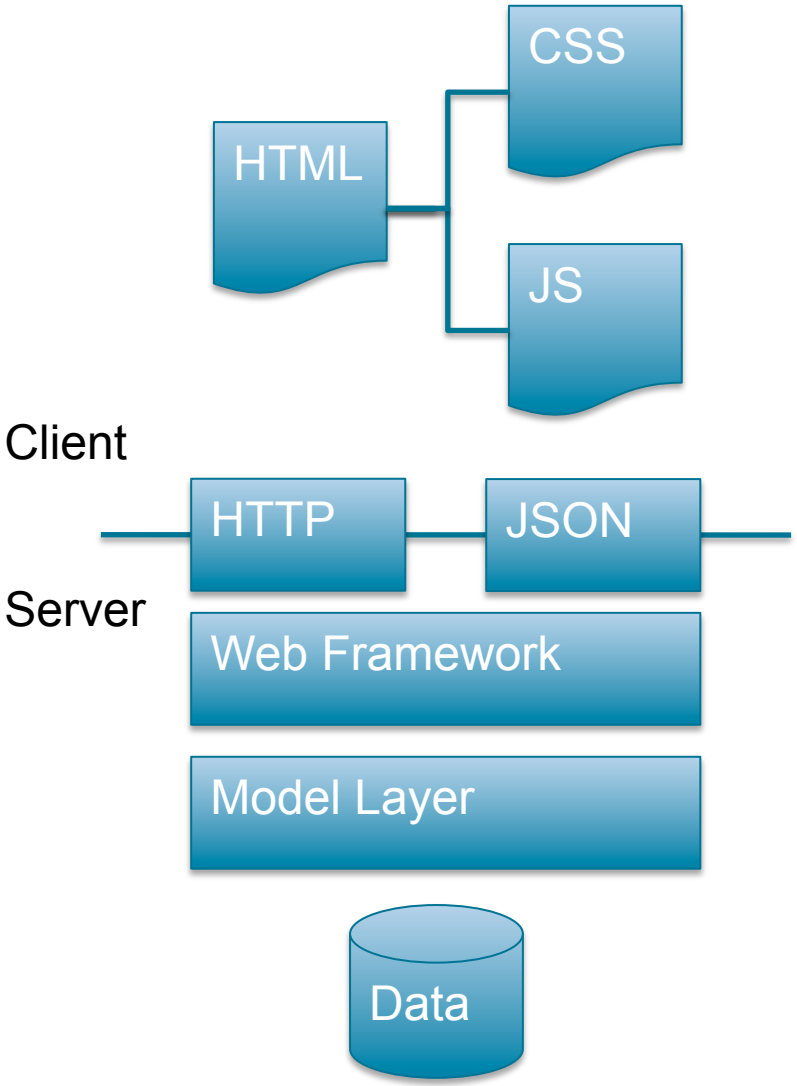
Enabling Technologies for a Native Vis Stack

Web-enabled visualizations are made possible by the convergence of many trends.

- AJAX
  - Web pages don't need to reload on every action!
- Canvas/SVG
  - Images can be rendered directly in the browser!
- jQuery/Prototype
  - JavaScript doesn't have to be poorly written!
- REST
  - URLs can be data accessors!
- Google, Microsoft, Apple, Mozilla
  - JavaScript can be very fast!

Anatomy of a Modern Web Application

Web applications are implemented as a combination of client and server side code that work together to provide a seamless user experience.



- HTML documents define the static structure of the current page
- CSS allows elements to be styled dynamically
- JavaScript is the client side language that enables interactive pages
- HTTP is the transport protocol
- JSON is a data format used to marshal data between processes and languages
- Web frameworks are lightweight and easy to use
- ORM, ODM, and user defined APIs provide application logic and data access

Modern JavaScript

At some point in the recent past, good programmers started using JavaScript… the results have been very interesting.

- JavaScript is primarily used to manipulate the DOM
  - ("The DOM" is just the HTML on the page)
- JavaScript lacks standard libraries, but the community has filled in the gaps
  - jQuery (used almost everywhere)
  - Backbone (decouple data from the DOM)
  - D3 (data transformation, primarily for vis)

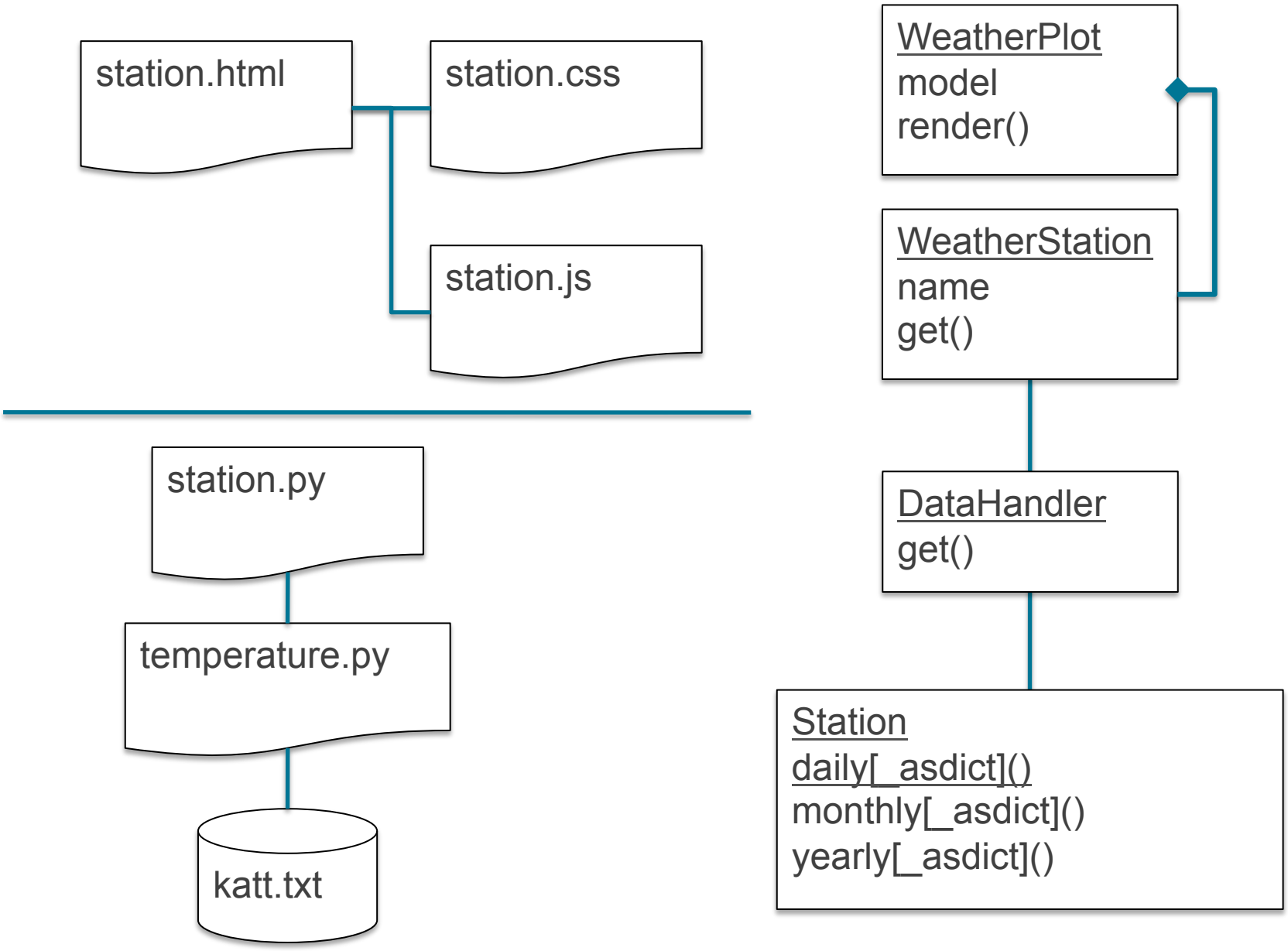
A Simple Application: Temperature Data Viewer

Let's build a simple application that takes historical temperature data from a weather station and displays it on a Web page.

Steps:

1. Acquire/clean data (Amazon/InfoChimps)
2. Design Python Analytics and API (Numpy)
3. Wrap Python API in RESTish Web API (Tornado)
4. Develop client side models (Backbone)
5. Develop client side views (D3)
6. Tie it all together (jQuery, CSS)

Application Architecture



Links

Libraries

- http://d3js.org/
- http://backbonejs.org/
- http://www.tornadoweb.org/

Tutorials

- http://www.w3schools.com/
- http://alignedleft.com/tutorials/d3/

Thank You!

Chris Mueller, Ph.D.
President and CTO
chris.mueller@lab7.io

Enjoy developing scientific applications?
We are hiring!

Lab7 Systems, Inc. ▪ Austin, TX
www.lab7.io