The Dashboarding Problem
Analytics team at Wikimedia Foundation
Goal:

Easily Visualize Editor Engagement for all Wikimedia projects
Side Goal:

Explore Dashboarding Technologies
<table>
<thead>
<tr>
<th>Date</th>
<th>project</th>
<th>pages created</th>
</tr>
</thead>
<tbody>
<tr>
<td>20140101</td>
<td>arwiki</td>
<td>50</td>
</tr>
<tr>
<td>20140102</td>
<td>arwiki</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>20140101</td>
<td>dewiki</td>
<td>100</td>
</tr>
<tr>
<td>20140102</td>
<td>dewiki</td>
<td>200</td>
</tr>
</tbody>
</table>
Problem #1
Problem #1

Access to Data
Easily Visualize Editor Engagement for all Wikimedia projects

Everyone should have access!
Problem #1

Public Access to Data
Problem #2

Visualization
New post - Visualizing Algorithms - using vision to study processes & systems. bost.ocks.org/mike/algorithm... #eyeo2014
Visualization is an easier problem than data access.
This JavaScript...

```javascript
new Dygraph(div, "ny-vs-sf.txt", {
  legend: 'always',
  title: 'NYC vs. SF',
  showRoller: true,
  rollPeriod: 14,
  customBars: true,
  yLabel: 'Temperature (F)',
});

...makes this chart!
```
Rickshaw is a JavaScript toolkit for creating interactive time series graphs.

Rickshaw on GitHub
Vega is a visualization grammar, a declarative format for creating, saving and sharing visualization designs.

With Vega you can describe data visualizations in a JSON format, and generate interactive views using either HTML5 Canvas or SVG.

Read the tutorial, browse the documentation, join the discussion, and explore visualizations using the web-based Vega Editor.
Problem #3
Problem #3
Information Architecture
How do you explore data for more than 800 projects?
#1 Access to Data
#2 Visualization
#3 Information Architecture
Design knows best

#3 Information Architecture
#1 Access to Data
Tech Decision: Make all data available via HTTP
Index of /static/public/datafiles

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NamespaceEdits/</td>
<td>04-Sep-2014 19:24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NewlyRegistered/</td>
<td>21-Aug-2014 18:04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PagesCreated/</td>
<td>04-Sep-2014 19:29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RollingActiveEditor/</td>
<td>21-Aug-2014 18:07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RollingNewActiveEditor/</td>
<td>12-Sep-2014 17:08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RollingSurvivingNewActiveEditor/</td>
<td>12-Sep-2014 17:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>available-metrics.json</td>
<td>16-Sep-2014 13:58</td>
<td>1.2K</td>
<td></td>
</tr>
<tr>
<td>available-projects.json</td>
<td>12-Sep-2014 20:38</td>
<td>229K</td>
<td></td>
</tr>
<tr>
<td>defaultDashboard.json</td>
<td>12-Sep-2014 20:38</td>
<td>192</td>
<td></td>
</tr>
</tbody>
</table>

Apache/2.2.22 (Ubuntu) Server at metrics.wmflabs.org Port 80
Data is available in JSON format
```json
{
    "result": {
        "Sum": {
            "newly_registered": {
                "2014-08-26 00:00:00": 364.0,
                "2014-08-24 00:00:00": 331.0,
                "2014-08-21 00:00:00": 382.0,
                "2014-08-30 00:00:00": 383.0,
                "2014-09-25 00:00:00": 372.0,
                "2014-08-22 00:00:00": 367.0,
                "2014-09-24 00:00:00": 410.0,
                "2014-09-06 00:00:00": 370.0,
                "2014-09-09 00:00:00": 370.0,
                "2014-09-27 00:00:00": 321.0,
                "2014-09-05 00:00:00": 317.0
            }
        }
    },
    "parameters": {
        "Cohort": "arwiki",
        "Created On": "2014-09-13 00:00:00",
        "Metric_end_date": "2014-09-13 00:00:00",
        "Metric_start_date": "2014-09-12 00:00:00",
        "Cohort Size": 0,
        "Metric": "NewlyRegistered"
    }
}
```
Data is available to everyone.
Why not?...

Server-less
Why not?...
Server-less
What is the middle tier for anyways?
No authentication needed

Decouples visualization piece from data gathering piece
We are server-less, we need to store the bootstrap state of the dashboard somewhere.
https://metrics.wmflabs.org/static/public/dash/

```json
{
   "defaultProjects": [
      "enwiki", "eswiki", "dewiki", "frwiki", "ruwiki", "jawiki", "itwiki"
   ],
   "defaultMetrics": [
      "RollingActiveEditor", "NewlyRegistered"
   ]
}
```
Request

Web Server

Dashboard Panels and Visualization
Only HTML and JS

Data Request

Data as JSON

Any other data store. Yours?

Meta Data Request

Meta Data as JSON

MediaWiki Storage
(MetaData and Graph description Store)
Technology Stack
Must Haves:

Package Manager
Dependency Loading
Data Binding Library
DOM manipulation and ajax
Testing
Technology Stack

Micro libraries: “each library does 1 thing well”
> Bower install
<table>
<thead>
<tr>
<th>Name Path</th>
<th>Method</th>
<th>Status Text</th>
<th>Type</th>
<th>Initiator</th>
<th>Size Content</th>
<th>Time Latency</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>dash/</td>
<td>GET</td>
<td>200 OK</td>
<td>text/html</td>
<td>Other</td>
<td>631 B</td>
<td>108 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>351 B</td>
<td>107 ms</td>
<td></td>
</tr>
<tr>
<td>style-1a1858fd.css</td>
<td>GET</td>
<td>200 OK</td>
<td>text/css</td>
<td>metrics.wmflabs.org</td>
<td>15.7 KB</td>
<td>296 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Parser</td>
<td>92.0 KB</td>
<td>294 ms</td>
<td></td>
</tr>
<tr>
<td>scripts-0f0bca6b.js</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>metrics.wmflabs.org</td>
<td>84.5 KB</td>
<td>775 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Parser</td>
<td>262 KB</td>
<td>326 ms</td>
<td></td>
</tr>
<tr>
<td>available-projects.json</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>21.4 KB</td>
<td>724 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>229 KB</td>
<td>696 ms</td>
<td></td>
</tr>
<tr>
<td>defaultDashboard.json</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>564 B</td>
<td>142 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>192 B</td>
<td>142 ms</td>
<td></td>
</tr>
<tr>
<td>available-metrics.json</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>691 B</td>
<td>147 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>1.2 KB</td>
<td>146 ms</td>
<td></td>
</tr>
<tr>
<td>project-selector-aa5c2175.js</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>11.3 KB</td>
<td>210 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>34.9 KB</td>
<td>209 ms</td>
<td></td>
</tr>
<tr>
<td>icons.svg</td>
<td>GET</td>
<td>200 OK</td>
<td>image/svg</td>
<td>scripts-0f0bca6b.js</td>
<td>194 KB</td>
<td>2.09 s</td>
<td></td>
</tr>
<tr>
<td>/static/public/badfonts/</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>193 KB</td>
<td>436 ms</td>
<td></td>
</tr>
<tr>
<td>vega-timeseries-95ddf252.js</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>80.9 KB</td>
<td>1.52 s</td>
<td></td>
</tr>
<tr>
<td>/static/public</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>243 KB</td>
<td>590 ms</td>
<td></td>
</tr>
<tr>
<td>jawiki.json</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>856 B</td>
<td>393 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public/datafiles/RollingActiveEditor</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>2.3 KB</td>
<td>392 ms</td>
<td></td>
</tr>
<tr>
<td>enwiki.json</td>
<td>GET</td>
<td>200 OK</td>
<td>application</td>
<td>scripts-0f0bca6b.js</td>
<td>895 B</td>
<td>397 ms</td>
<td></td>
</tr>
<tr>
<td>/static/public/datafiles/RollingActiveEditor</td>
<td></td>
<td></td>
<td></td>
<td>Script</td>
<td>2.3 KB</td>
<td>396 ms</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- **most**
- **Dashboard**
- **2nd load, visual**
- **Fetching**
The Power
Of Knockout Components
Knockout is not Polymer (It's a LOT better)
Recollectable pieces

Web Server

Dashboard Panels and Visualization
Only HTML and JS

Data Request

Meta Data Request

Wikimetrics
(Data Store)

Data as JSON

Meta Data as JSON

MediaWiki Storage
(MetaData and Graph description Store)
Show me the code....
<section class="ui six column celled page grid">
  <div class="row">
    <section class="column projects-container">
      <project-selector params="projectOptions: projectOptions,
        languageOptions: languageOptions,
        defaultProjects: defaultProjects,
        reverseLookup: reverseLookup,
        selectedProjects: selectedProjects">
      </project-selector>
    </section>
  </section>
  <section class="five wide column main-container">
    <div class="ui segment">
      <metric-selector params="selectedMetric: selectedMetric,
        metrics: metrics,
        defaultSelection: defaultMetrics">
      </metric-selector>
      <time-selector></time-selector>
    </div>
    <div class="ui segment graph">
      <wikimetrics-visualizer params="projects: selectedProjects,
        metric: selectedMetric">
      </wikimetrics-visualizer>
    </div>
  </section>
</div>
</section>
On the AngularJS team, we rely on testing and we always seek better tools to make our life easier. That’s why we created Karma - a test runner that fits all our needs.
Dashiki: a thin wrapper on top of OS technology + information architecture. We can use dashiki to build other dashboards.
https://github.com/wikimedia/analytics-dashiki
Lessons:

Server-less = Reduced complexity
Easy(est) to deploy
CORS
No error log for now
Bootstrap default state
Keep things simple. No authentication.
Questions?