

DevSummit DC

February 11, 2015 | Washington, DC



ArcGIS API for JavaScript: Advanced Topics

Andy Gup, John Gravois, Ben Ramseth



Agenda

Working with data

Security

Templates

Andy Gup

agup@esri.com

@agup

John Gravios

jgravois@esri.com

@geoganstster

Ben Ramseth

bramseth@esri.com

@esriMapNinja

Working with Data

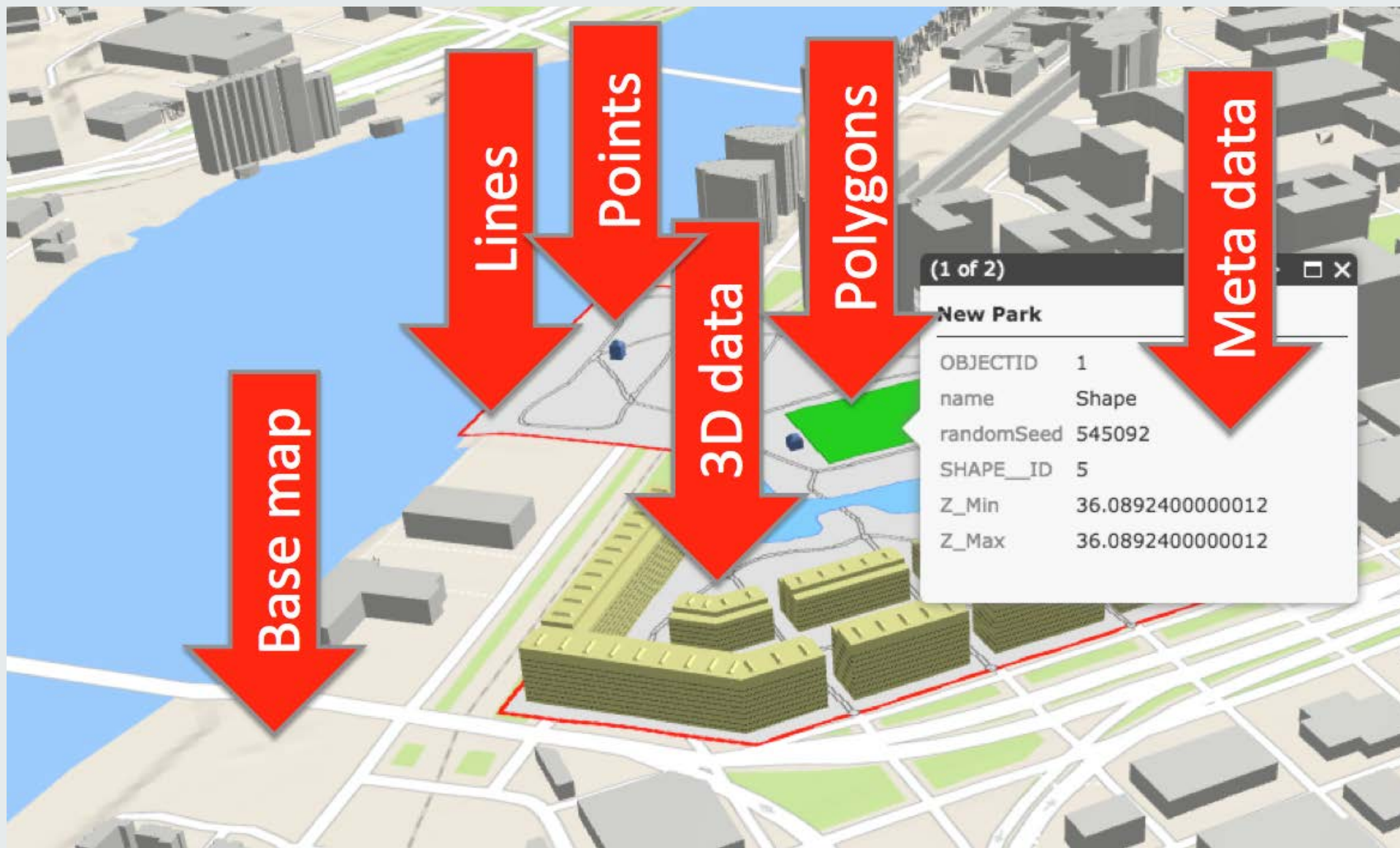
Data Visualization

Andy



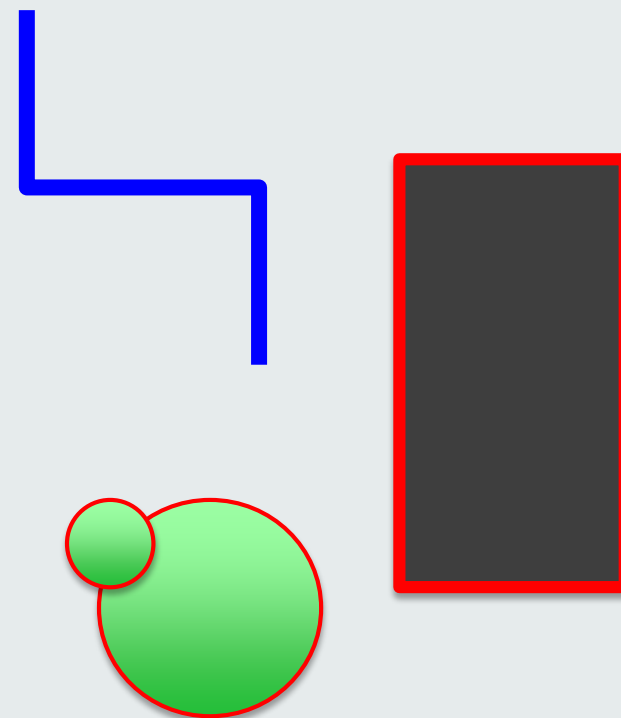
Data
Data
Data
Data
Data
Data

Lots of spatial data...



Geographic Features + Symbology

```
var feature = {  
  "type": "Feature",  
  "id": 3,  
  "geometry": {  
    "type": "Point",  
    "coordinates": [  
      -122.68287749,  
      45.526496967  
    ]  
  },  
  "properties": {  
    "FID": 3,  
    "PlaceName": "Starbucks",  
    "Place_addr": "1134 NW Glisan St Portland Oregon",  
    "Score": 100,  
    "lat": 45.526496967,  
    "lon": -122.68287749  
  }  
}
```



Bye-bye old style visualization!



Renderers to the rescue!

SimpleRenderer

HeatMapRenderer

TemporalRenderer

ClassBreaksRenderer

DotDensityRenderer

ScaleDependentRenderer



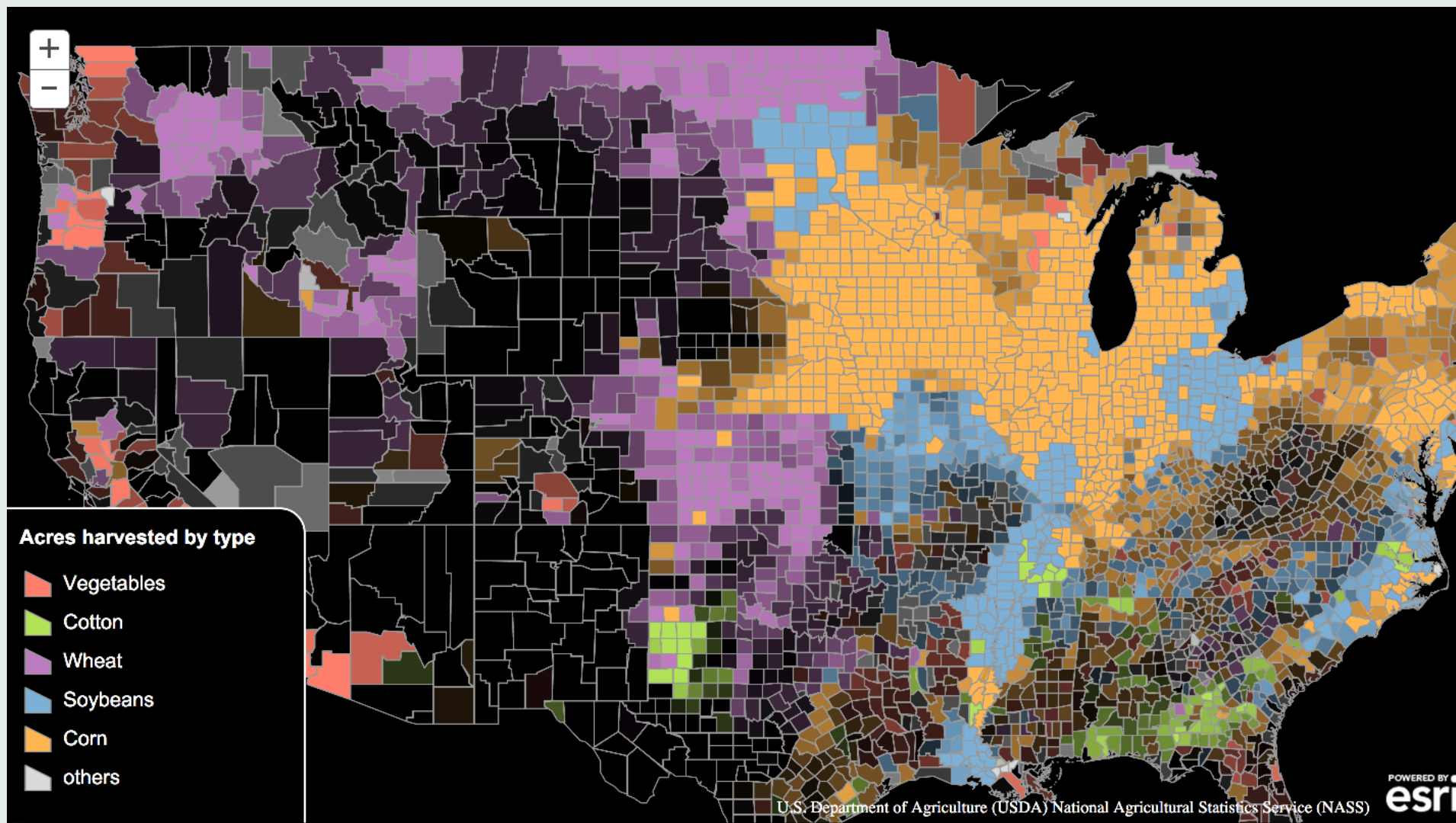
Why Renderers?

Applies symbology to an entire Feature Layer

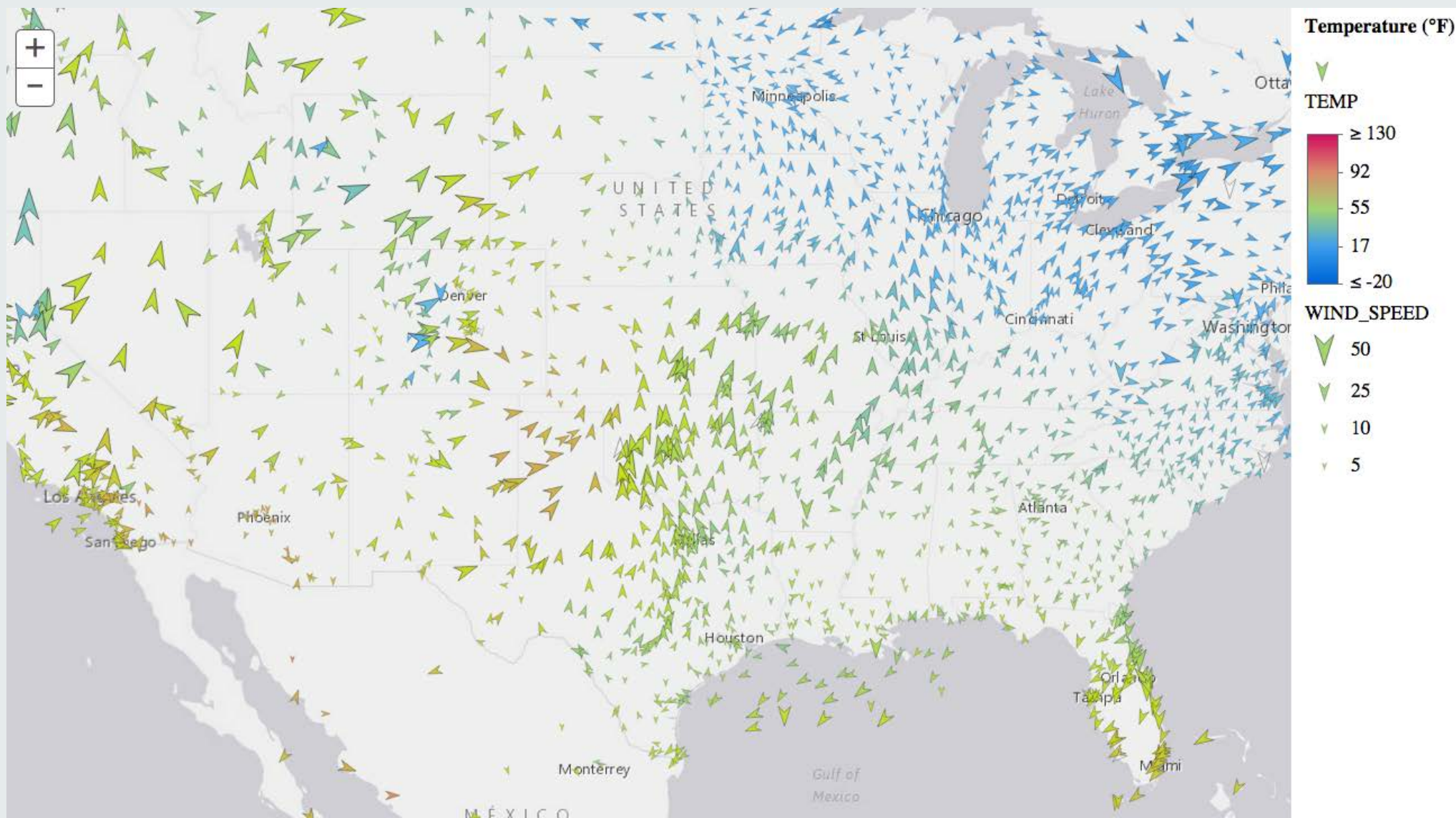
No “for” loops

Write the code once

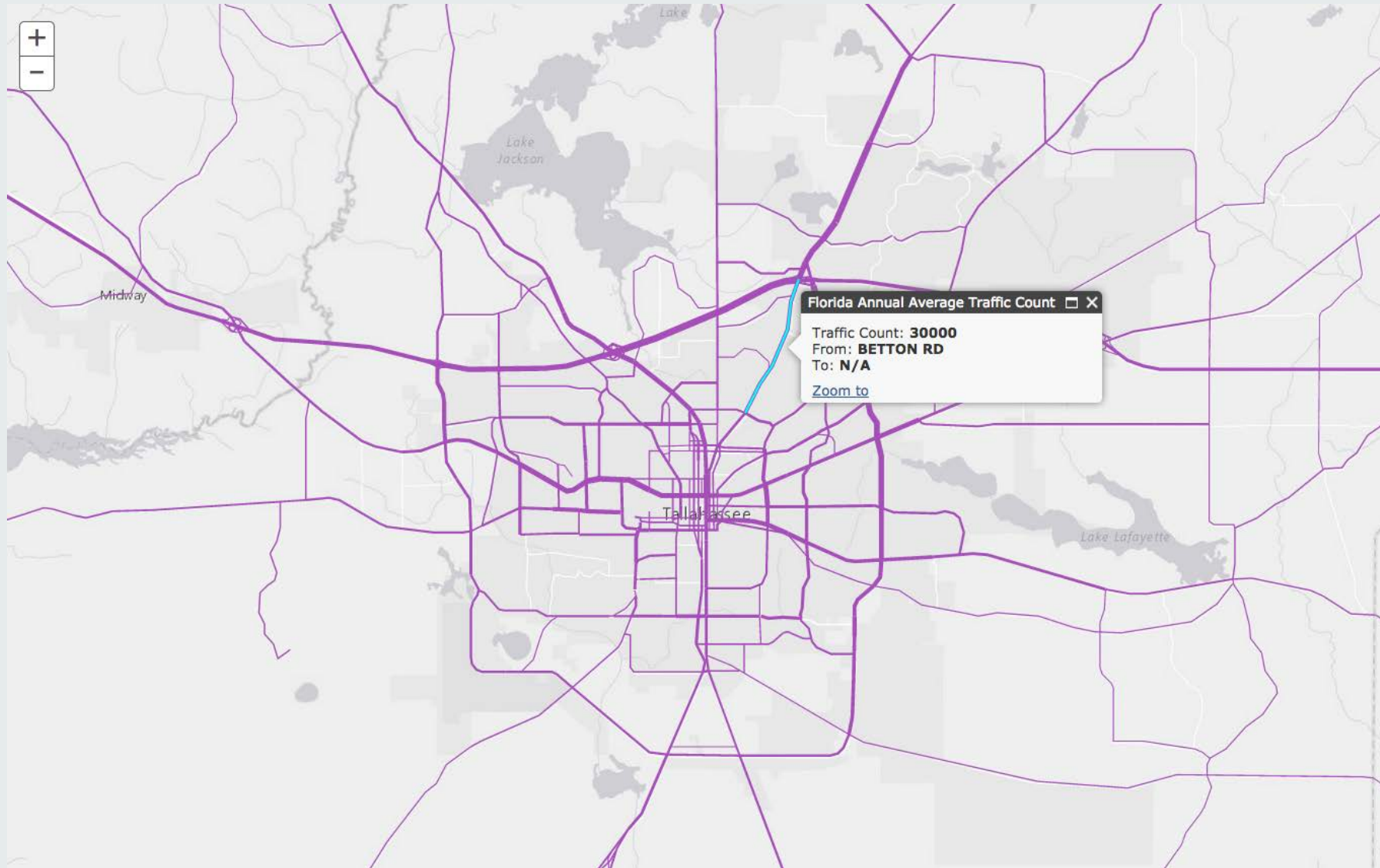
Opacity Renderer – Show data by color



Class Breaks Renderer – show data by color/size



Proportional Line Render – show data by size



Example: esri/renderers/SimpleRenderer

```
var symbol = new SimpleMarkerSymbol();  
symbol.setColor(new Color("#ffa500"));  
  
var renderer = new SimpleRenderer(symbol);  
  
featureLayer.setRenderer(renderer);  
map.addLayer(featureLayer);
```


Example: esri/renderers/SimpleRenderer

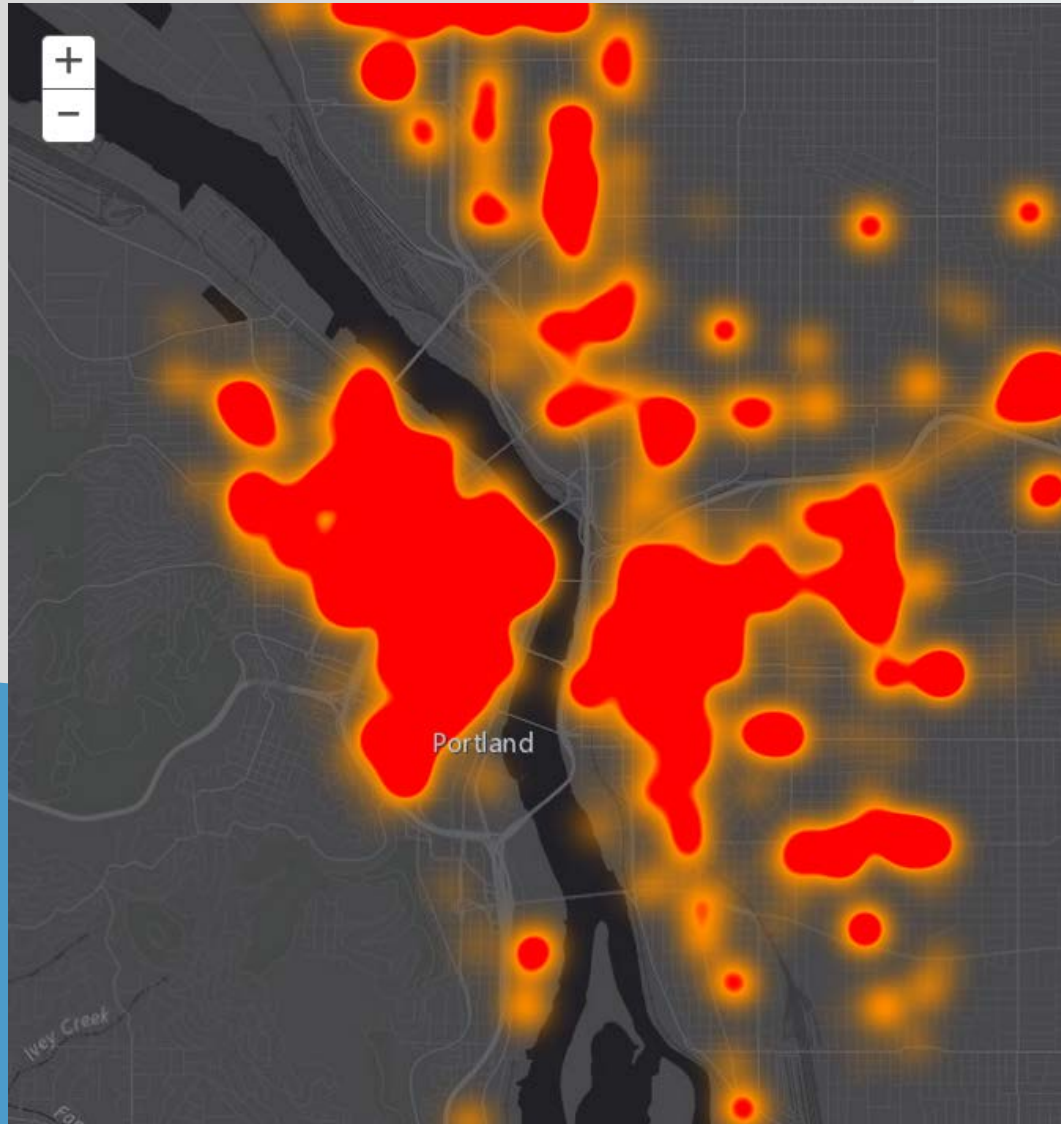
```
featureLayer.on("load",function(evt){  
    var symbol = new SimpleMarkerSymbol();  
    symbol.setColor(new Color("#ffa500"));  
  
    var renderer = new SimpleRenderer(symbol);  
  
    featureLayer.setRenderer(renderer);  
    map.addLayer(featureLayer);  
});
```

Example: esri/renderers/SimpleRenderer

```
var featureLayer = new FeatureLayer(featureLayerUrl,  
  {  
    mode: FeatureLayer.MODE_SNAPSHOT,  
    infoTemplate: infoTemplate,  
    outFields: ["ATTRIBUTE1", "ATTRIBUTE2"]  
  }  
);
```

Example: esri/renderers/HeatmapRenderer

```
var heatmapRenderer = new HeatmapRenderer({  
    field: "NUMSPACES",  
    blurRadius: 10,  
    maxPixelIntensity: 100,  
    minPixelIntensity: 0  
});  
  
featureLayer.setRenderer(heatmapRenderer);  
map.addLayer(featureLayer);
```



Renderer Demo

Real-time Streaming

Andy

Why Stream with Web Sockets?

Get real-time updates

Only need to open a connection once

Examples

- Any type of tracking
- News feeds
- Payment systems

Web Sockets – The tech specs...

TCP/IP

Low latency

Full-duplex

Works with SSL

Uses same ports as HTTP (80) and HTTPS (443)

Can traverse firewalls and proxies

Web Sockets browser support

Web Sockets  - CR

Bidirectional communication technology for web apps


Global 84.54% + 1.47% = 86.01%

unprefixed: 84.54% + 1.32% = 85.86%

Current aligned Usage relative Show all

| IE | Firefox | Chrome | Safari | Opera | iOS Safari * | Opera Mini * | Android Browser * | Chrome for Android |
|----|---------|--------|--------|-------|--------------|--------------|-------------------|--------------------|
| | | 31 | | | | | | |
| | | 35 | | | | | | |
| | | 36 | | | | | | |
| 8 | | 37 | | | | | 4.1 | |
| 9 | | 38 | | | | | 4.3 | |
| 10 | 34 | 39 | 7.1 | 26 | 7.1 | | 4.4 | |
| 11 | 35 | 40 | 8 | 27 | 8.1 | 8 | 4.4.4 | 40 |
| TP | 36 | 41 | | 28 | | | | |
| | 37 | 42 | | 29 | | | | |
| | 38 | 43 | | | | | | |

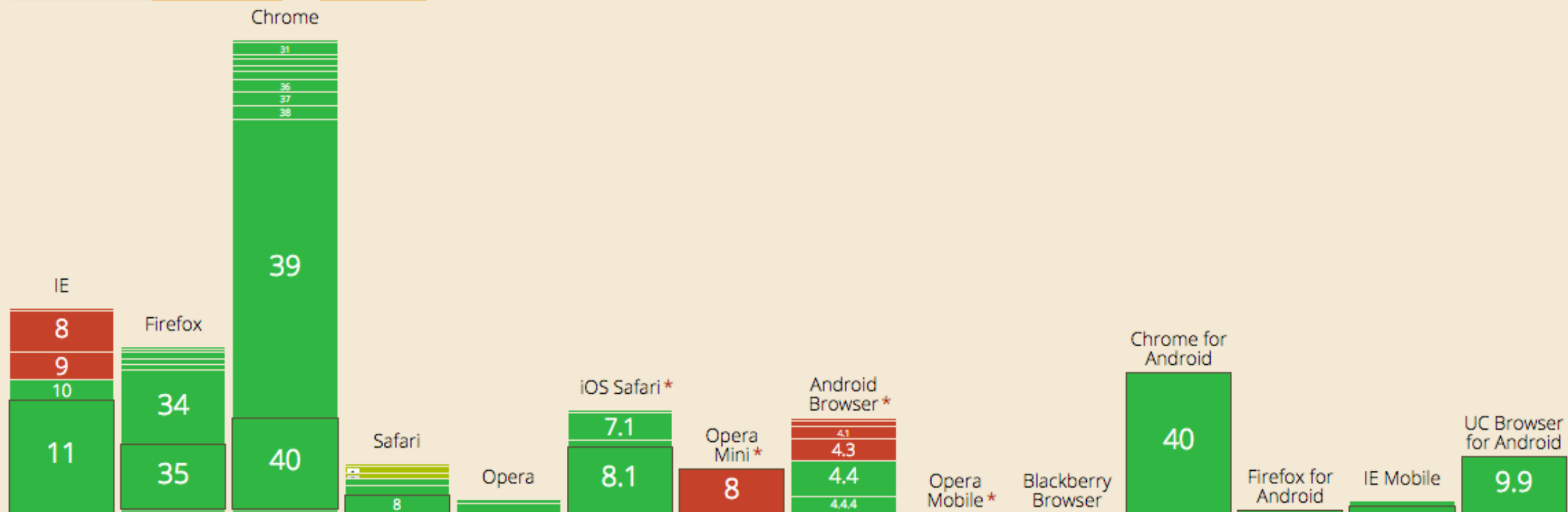
Web Sockets usage by browser

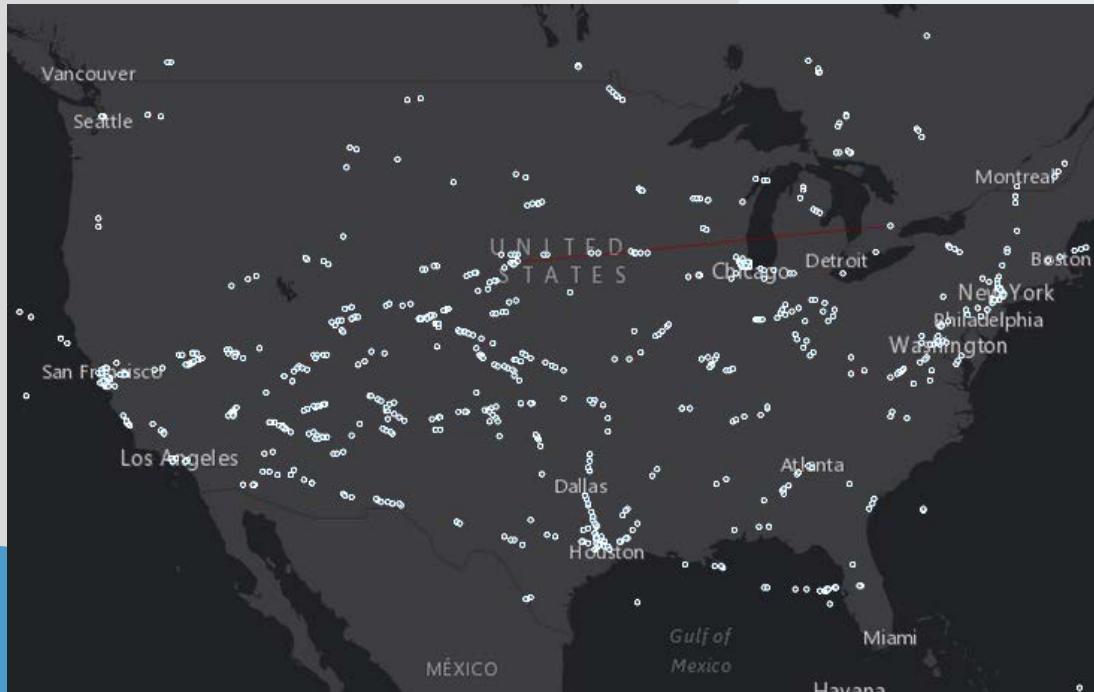
Web Sockets  - CR

Bidirectional communication technology for web apps

Global 84.54% + 1.47% = 86.01%
unprefixed: 84.54% + 1.32% = 85.86%

Current aligned Usage relative Showing all





StreamLayer demo

esri/layers/StreamLayer

Feature Layer

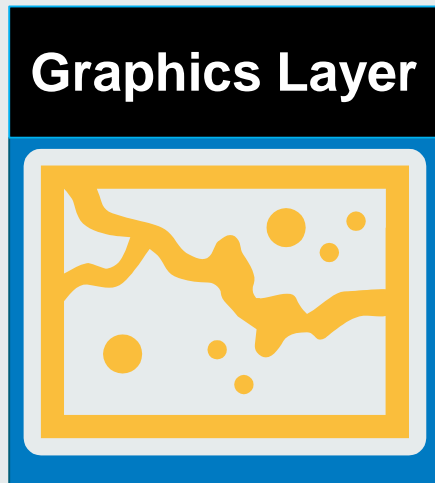
- Relatively static
- Pull

Stream Layer

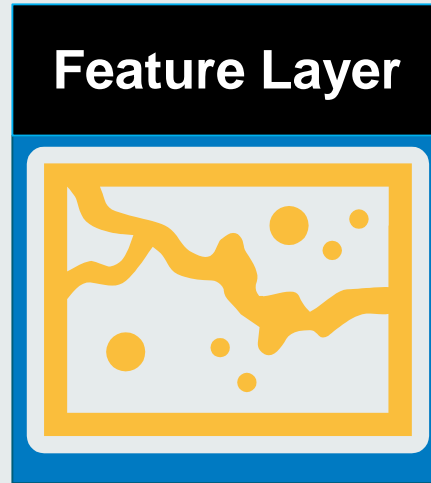
- Dynamic streams
- Continuous
- Push



StreamLayer



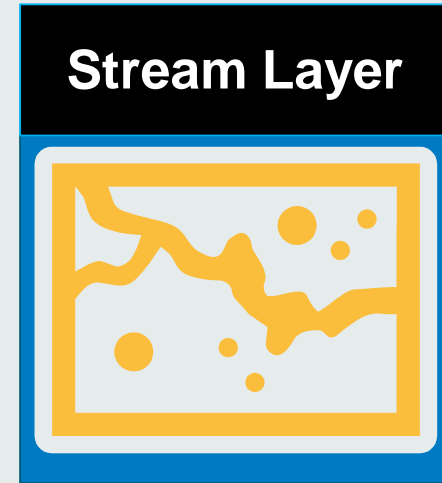
LOCAL ONLY



**HTTP
GET/POST**



ArcGIS Server



**Web Socket
Push**



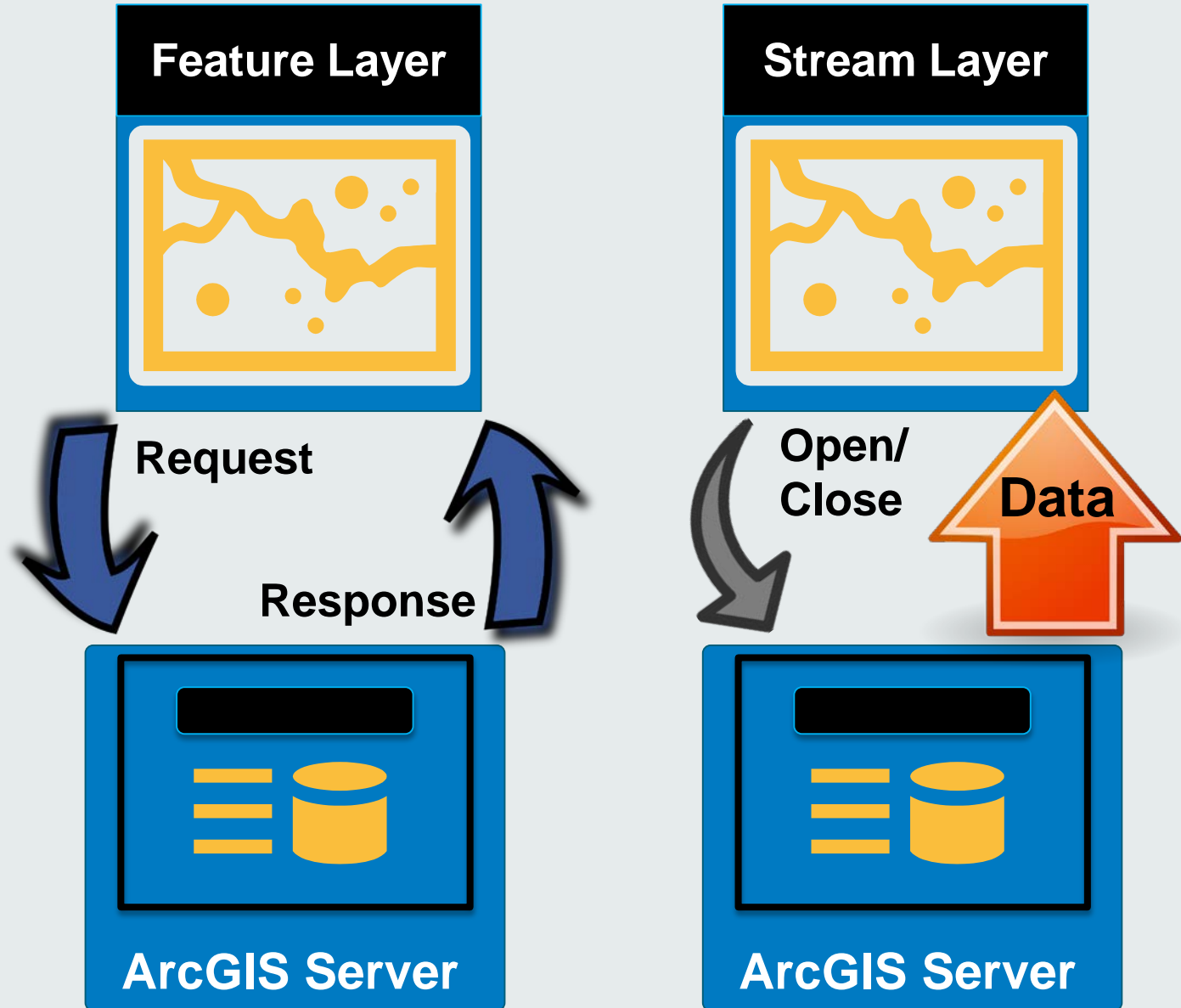
ArcGIS Server

StreamLayer

More efficient transfer of data.

Avoids need to poll for data updates

Auto-reconnect



Stream Layer

`esri/layer/Layer`

└─ `esri/layers/GraphicsLayer`

└─ `esri/layers/FeatureLayer`

└─ `esri/layers/StreamLayer`

StreamLayer

```
new StreamLayer(url, options?);
```

```
new StreamLayer(featureCollectionObject,  
options?);
```

StreamLayer

```
StreamLayer = new StreamLayer(svcUrl, {  
    purgeOptions: { displayCount: 10000 },  
    filter: filterObj,  
    infoTemplate: new InfoTemplate("Attributes", "${*}")  
});
```

StreamLayer – managing performance

```
purgeOptions: { displayCount: 10000 }
```

Set max number of features allowed

Purge based on FIFO

StreamLayer – shut down the connection #1

```
streamLayer.disconnect(function(evt) {  
    //Let user know  
});
```

Also triggers the `onDisconnect` event

StreamLayer – shut down the connection #2

```
map.removeLayer(streamLayer);
```

Completely removes layer

Also calls `disconnect()`

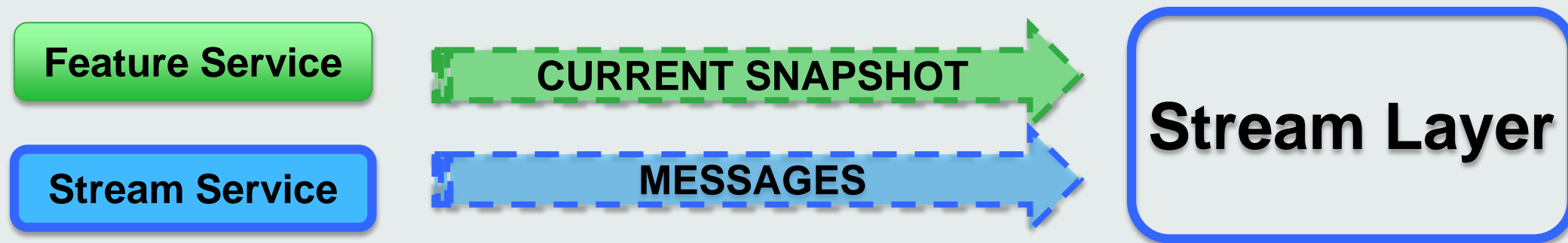
StreamLayer – re-establish connection

```
streamLayer.connect(function(evt) {  
    //Let user know  
});
```

Also triggers the `onConnect` event

Stream Service with buddied Feature Service

- Stream Service published with Feature Service
- Queries Feature Service before establishing web socket connection
- Client starts up with data snapshot



Stream Layer - What is needed?

ArcGIS GeoEvent Processor

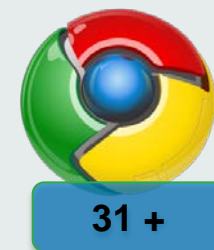
- Or, your own Web Socket Server (Esri JSON)

Browser that supports Web Sockets

- <http://caniuse.com/websockets>

Web Socket protocol allowed on network

`ws://, wss://`



www.websocket.org

Security

Templates and App Structure







Understanding our world.