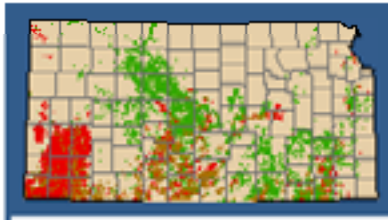
Patterns and Best Practices for Building Applications with the ArcGIS API for JavaScript

Jeremy Bartley

A personal history of building web maps

- Mo IMS
  - Don’t close the map window on the server!
- Arcview IMS
- Arcims
  - Html viewer
  - User driven custom apps
  - Server side blending of multiple arcims services



State of Kansas Geographic Information Systems Policy Board, Data Access and Support Center, and the Kansas Geological Survey
Please email questions and comments to Jeremy Bartley

Kansas Well Top Stratigraphy Viewer

help reset

Beginning Top: --------Viola Limestone Formation (19907)
Ending Top: --------Viola Limestone Formation (19907)
Well Top Go

The number in () is the number of well tops we have for that member. View Current Viewing Extent Stats Download Tops View Tops 3D

Overview Map

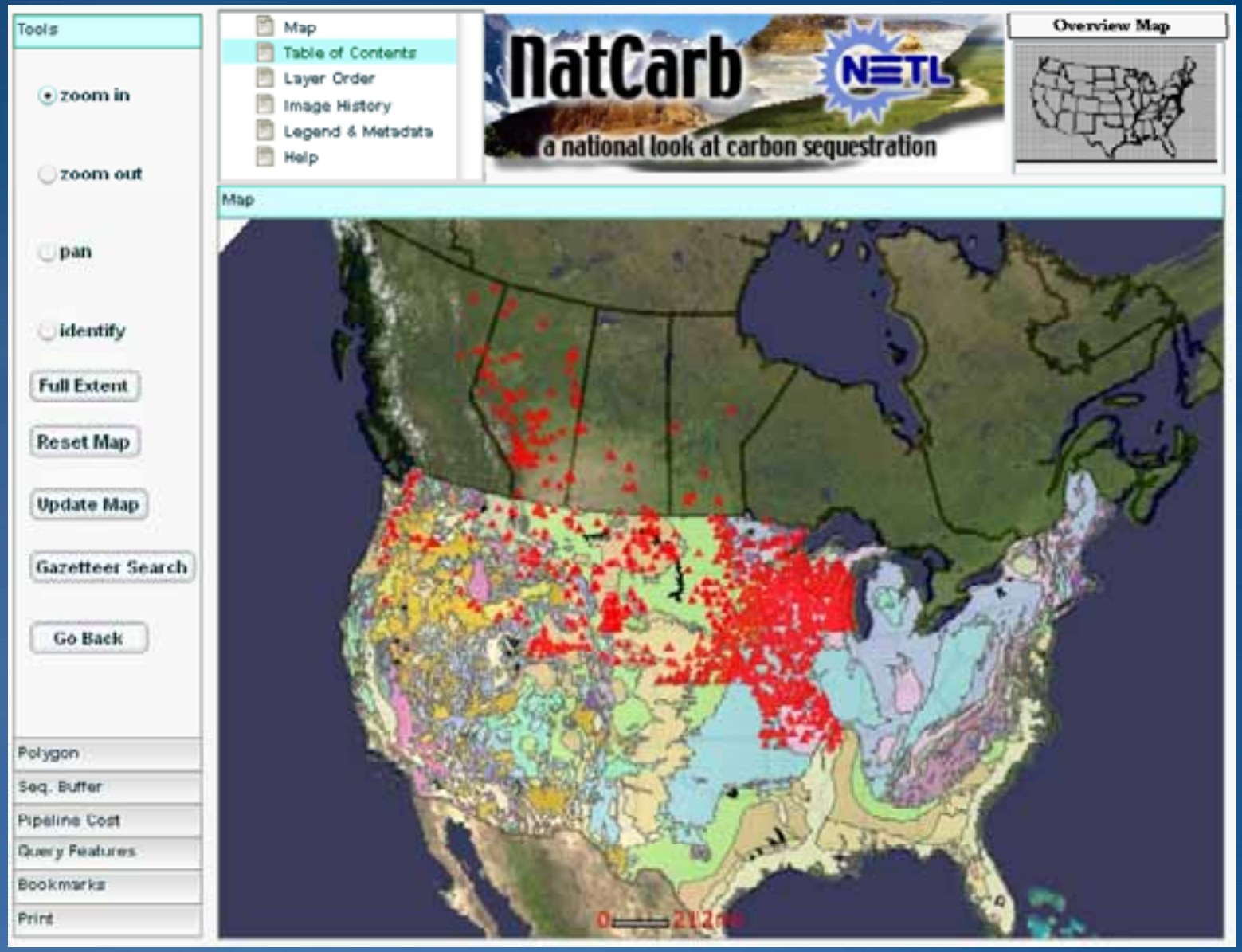
Legend
Depth
446 TO 2788
2788 TO 3083
3083 TO 3377
3377 TO 3672
3672 TO 3966
3966 TO 4261
4261 TO 4555
4555 TO 4849
4849 TO 5144
5144 TO 8189
County Boundary

Zoom Level: 1.5 Zoom In Zoom Out Pan Identify Section Profile
Full Extent

Table of Contents
Visible
Dynamic Stratigraphic Map
County Boundary
update map

A personal history of building web maps

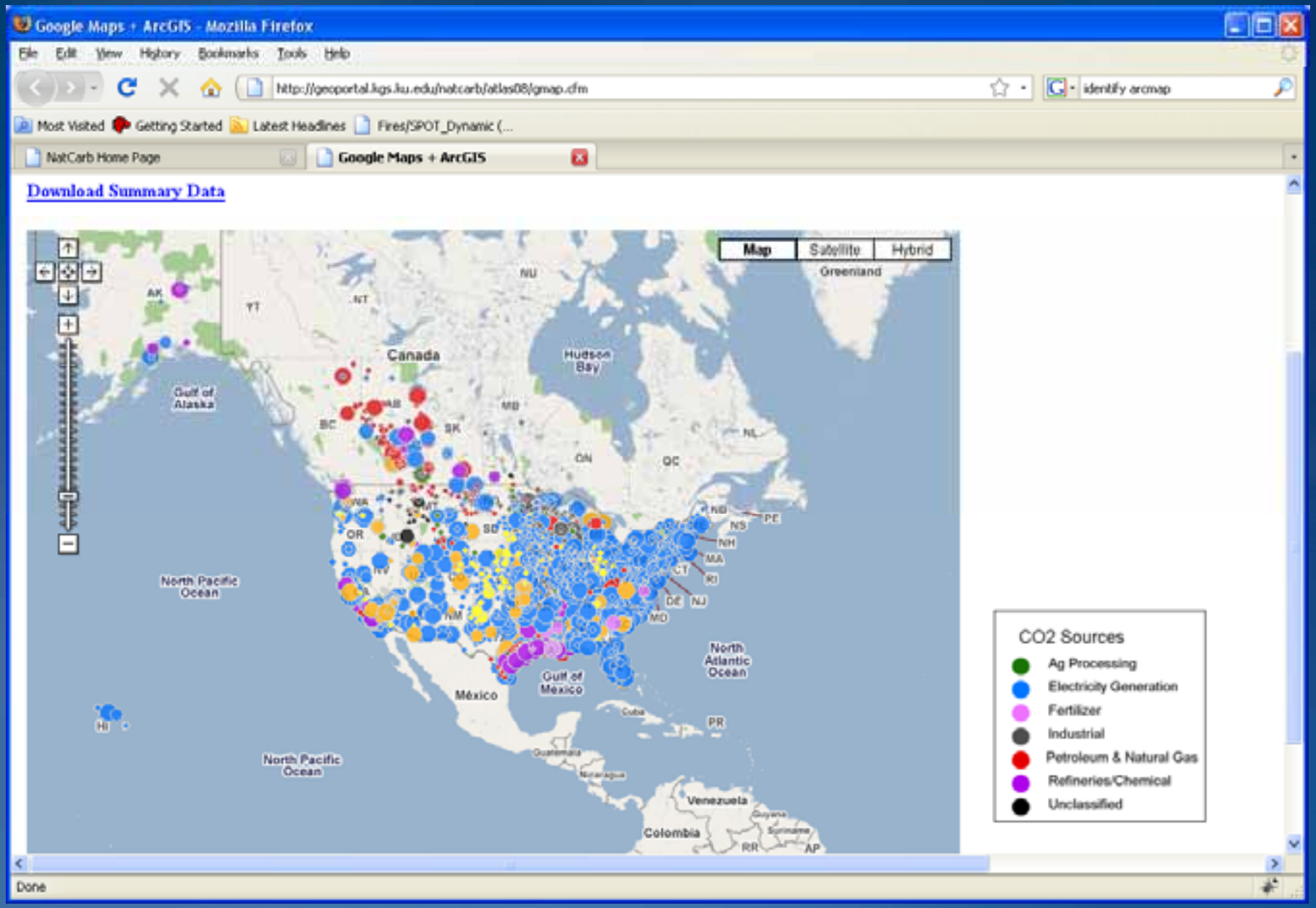
- Mo IMS
  - Don’t close the map window on the server!
- Arcview IMS
- Arcims
  - Html viewer
  - User driven custom apps
  - Server side blending of multiple arcims services


Tools
zoom in
zoom out
pan
identify
Full Extent
Reset Map
Update Map
Gazetteer Search
Go Back
Polygon
Seq. Buffer
Pipeline Cost
Query Features
Bookmarks
Print
Map
Table of Contents
Layer Order
Image History
Legend & Metadata
Help
NatCarb
NETL
a national look at carbon sequestration
Overview Map
Map

A personal history of building web maps

- Google! Tiles! Web Client!

- ArcGIS Server
  - ADF/Tiles/SOAP --server or client side blending of stateless services
  - REST API
  - JavaScript APIs --client blending of stateless services


Google Maps + ArcGIS - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://geoportal.kgs.ku.edu/natcarb/atlas08/gmap.cfm
identify arcmap
Most Visited
Getting Started
Latest Headlines
Fires/SPOT_Dynamic (...
NatCarb Home Page
Google Maps + ArcGIS
Download Summary Data
Map
Satellite
Hybrid
Greenland
Canada
Hudson Bay
Gulf of Alaska
North Pacific Ocean
North Atlantic Ocean
Gulf of Mexico
México
Venezuela
Colombia
CO2 Sources
Ag Processing
Electricity Generation
Fertilizer
Industrial
Petroleum & Natural Gas
Refineries/Chemical
Unclassified
Done

A personal history of building web maps

- Google! Tiles! Web Client!

- ArcGIS Server
  - ADF/Tiles/SOAP --server or client side blending of stateless services
  - REST API
  - JavaScript APIs --client blending of stateless services

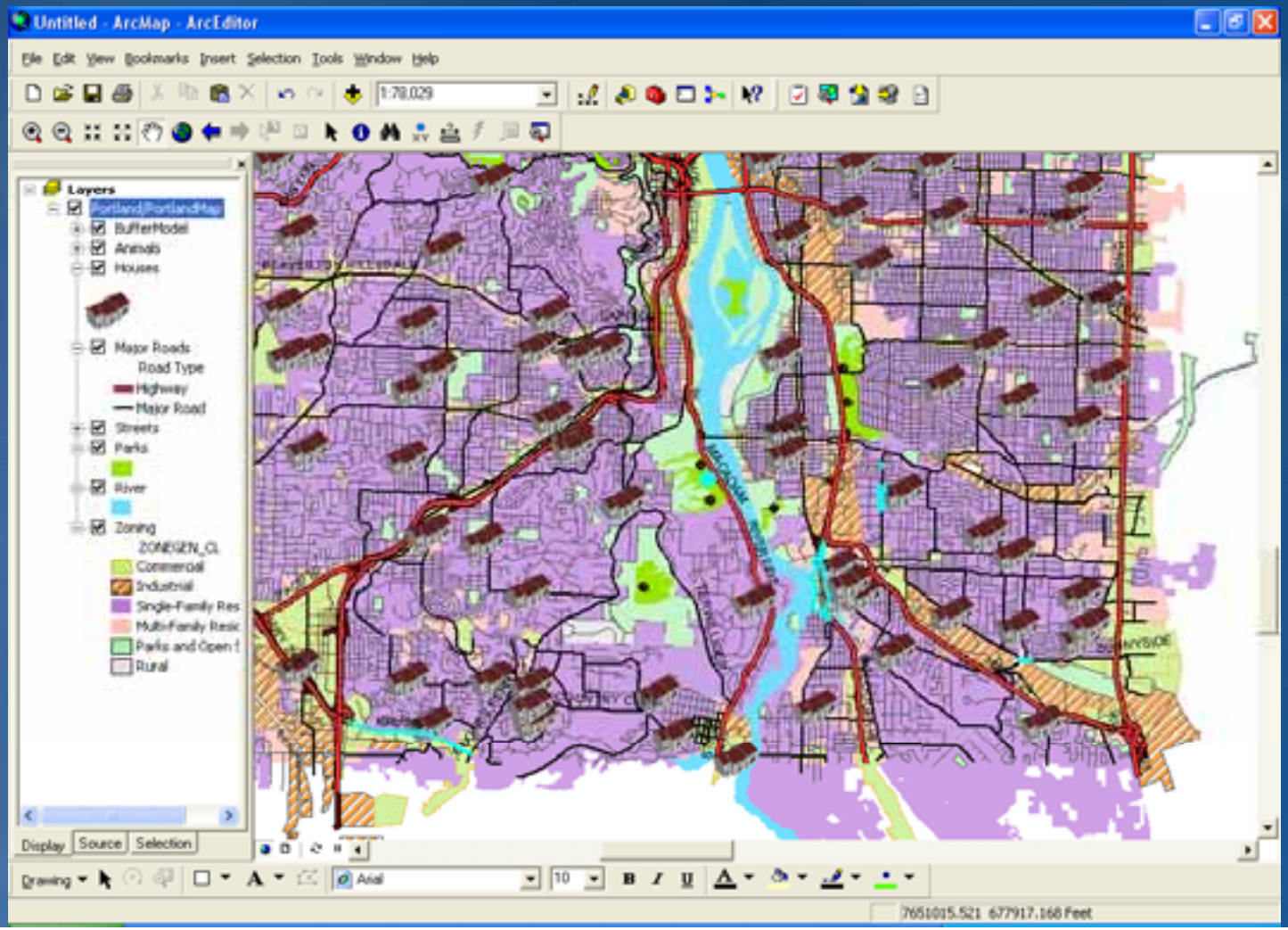
Technology has changed

Users expectations have changed

Problems have stayed the same

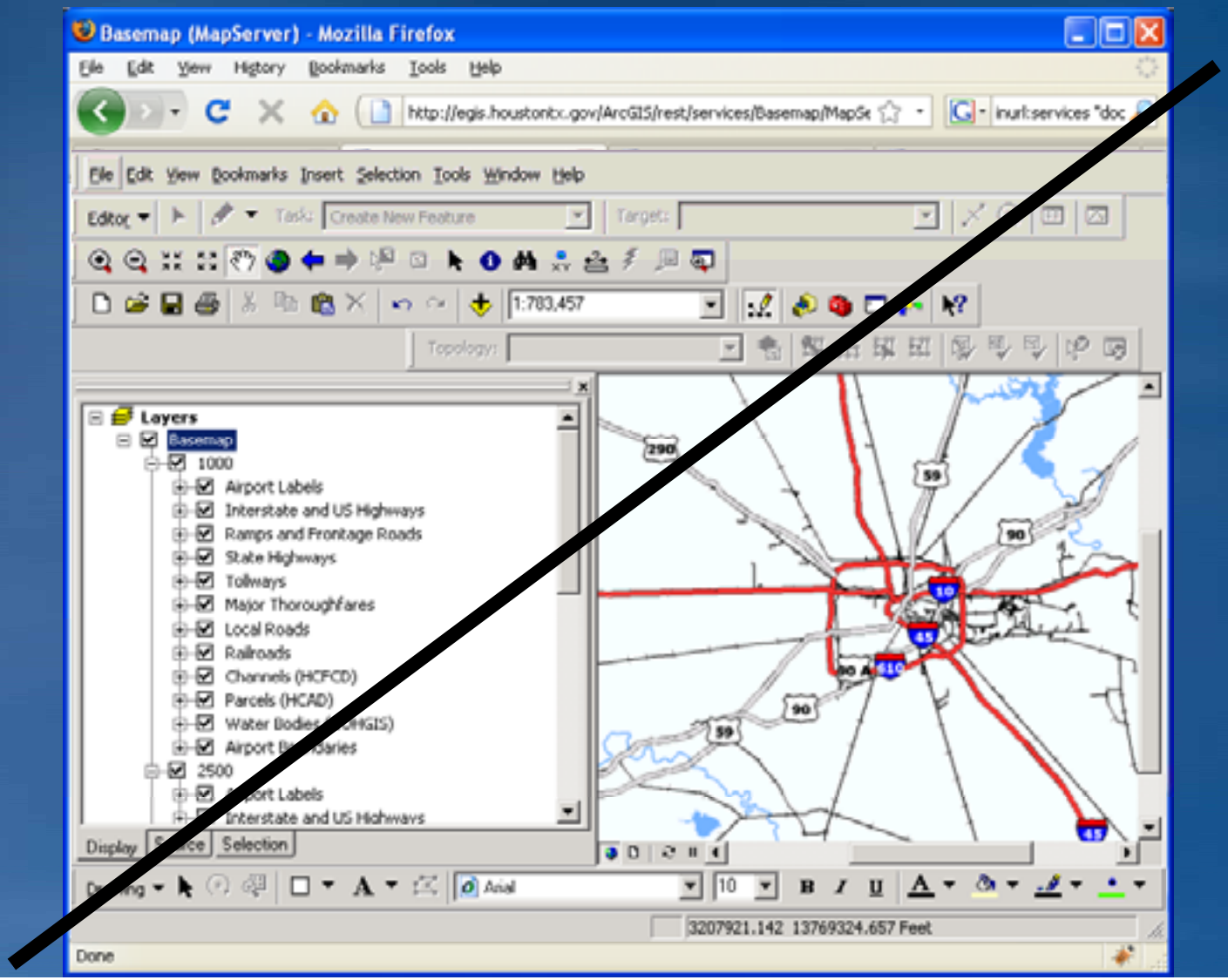
A Great Web Map…

- Is not made up of a single MXD
  - MXD is not the same as a Web Map



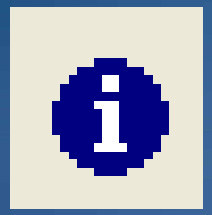
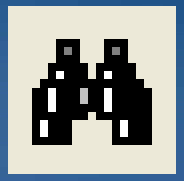
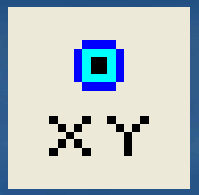
A Great Web Map…

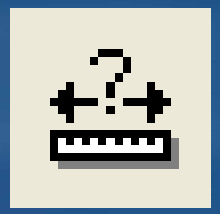
- **Is not ArcView On the Web**



A Great Web Map…

- Does not contain GIS jargon

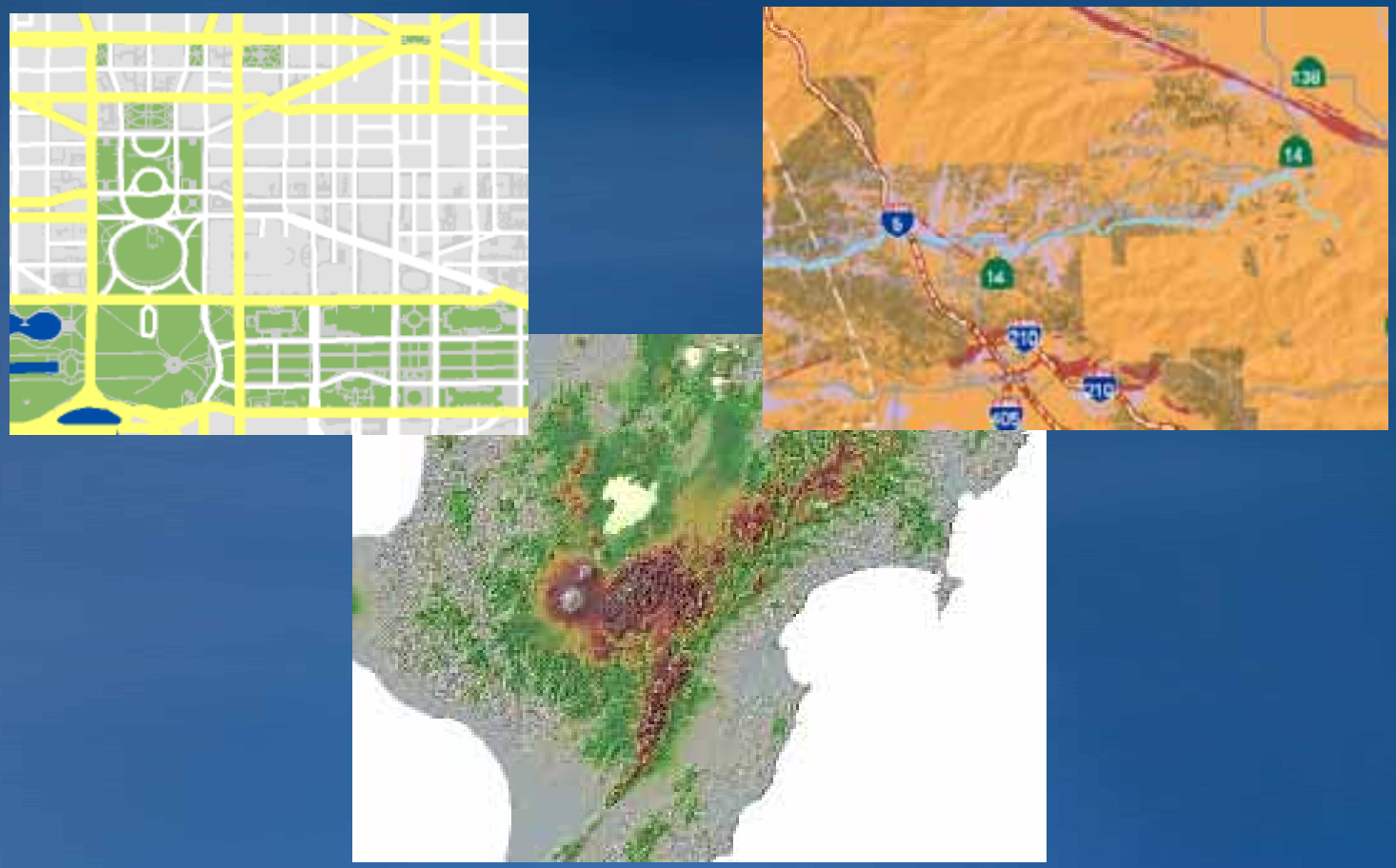






- Is not designed for people who know as much about GIS as you do

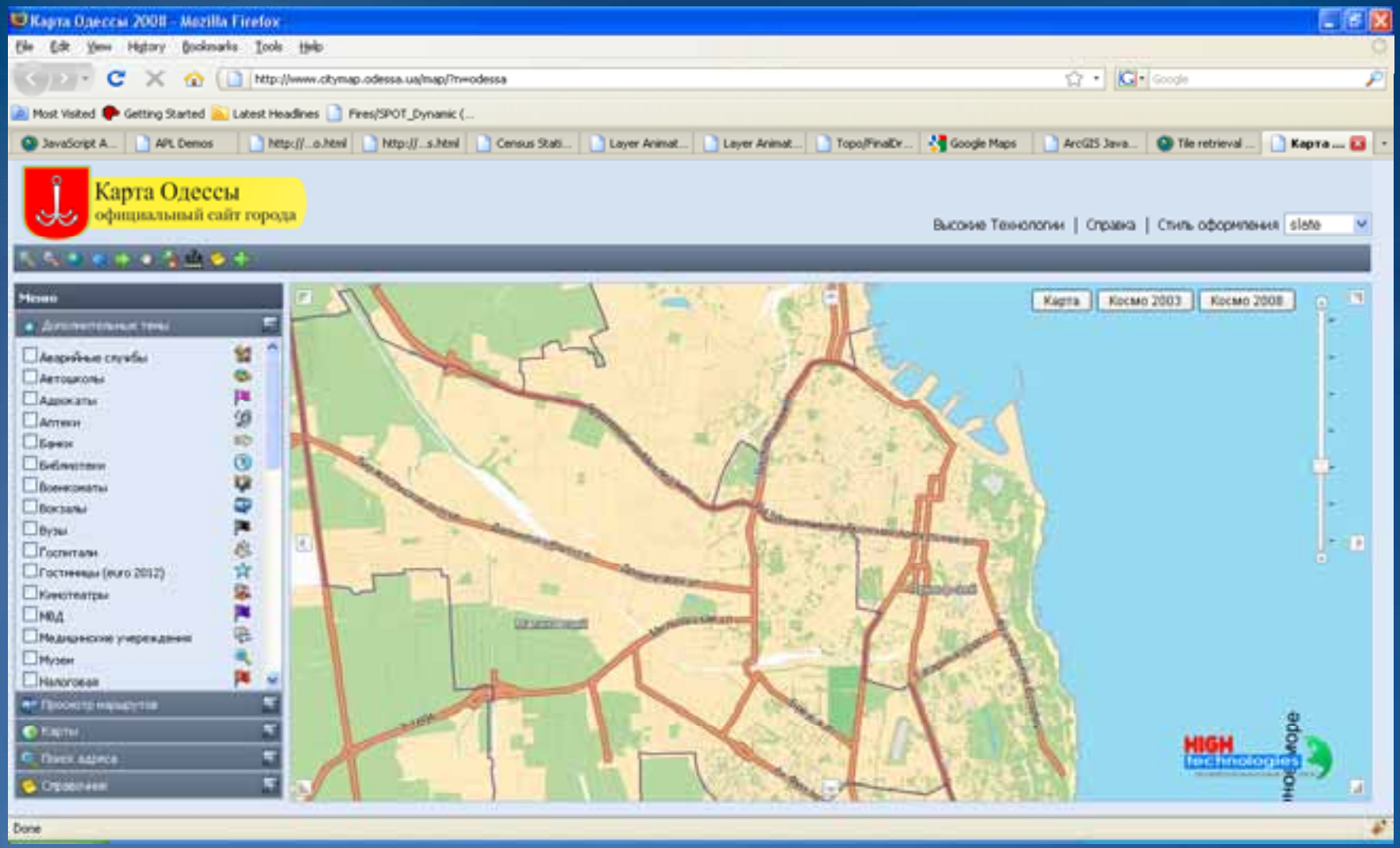
A Great Web Map…

- Is Fast



A Great Web Map…

- Is Designed for the end user



A Great Web Map…

- Contains the necessary map service layers needed to bring the web application to life
  - Basemap (provides a geographic frame of reference)
    - ArcGIS Online, Virtual Earth, Google Maps, your own designed
  - Operational Layers (show a focused item of interest on top of the base map)
    - Multiple ways to implement
- Contains as much pre-computed information as possible

How do you get your data/maps in a Web Map App?

- Author GIS resources (eg maps, locators, models) using desktop.
- Publish GIS resources to create GIS Services.
- Each capability is exposed to consumers as an independent GIS Web Service accessible over HTTP via SOAP or REST
- MXD/MSD becomes consumable as *map service layer* in the web map

Bringing your Web Map Application to Life

Bringing your Web Map Application to Life

- Web Map is made up of multiple client side layers
- Basemap Layers (provides a geographic frame of reference)
  - ArcGIS Online, Virtual Earth, Google Maps, your own designed basemap
- Operational Layers (show a focused item of interest on top of the base map)
  - vehicle locations, cadastral features, utility networks, weather data, traffic incidents, and ??
  - Multiple ways to implement

Designer and Developer View

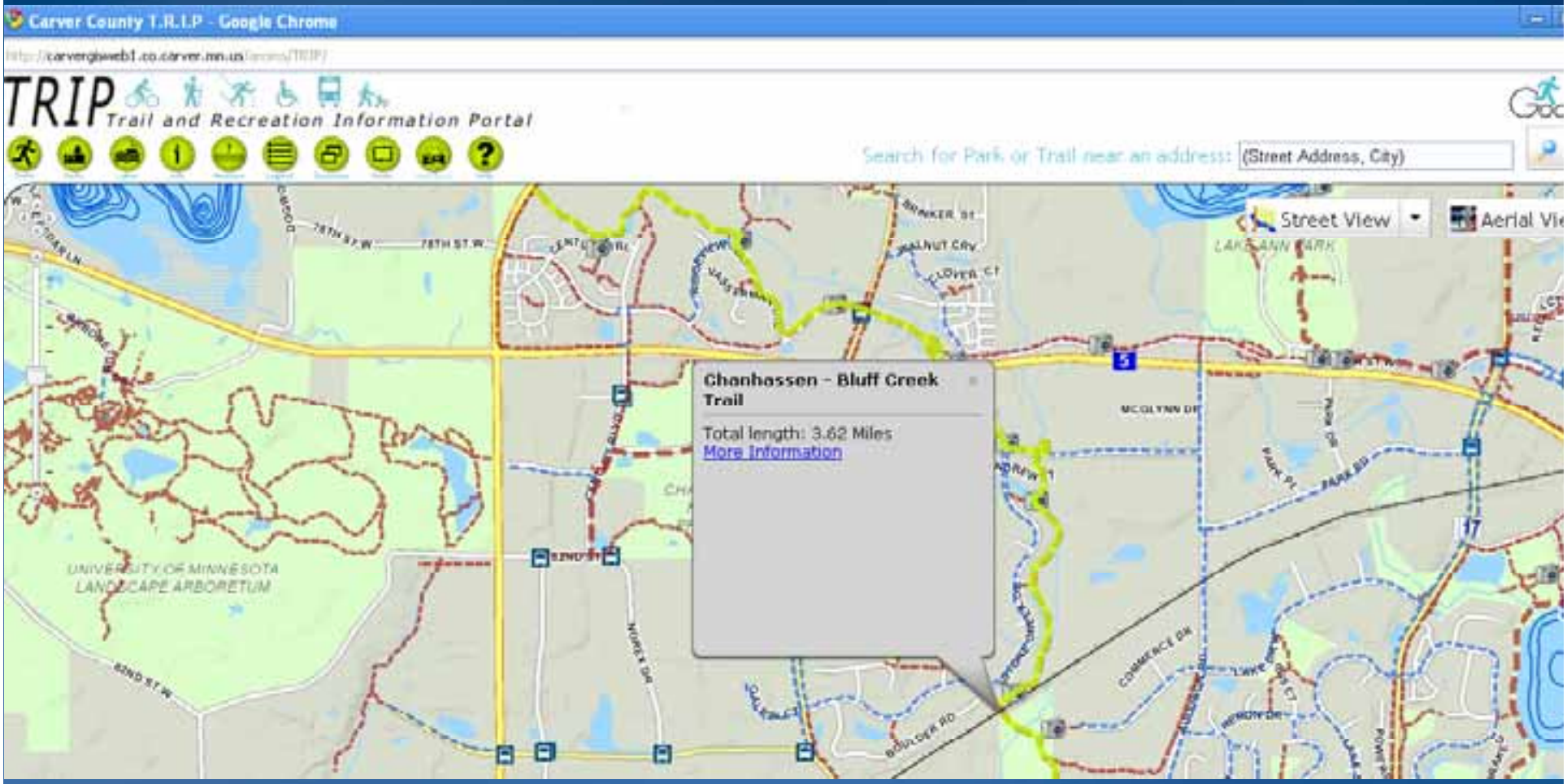
```
dojo.connect(map, "onZoomStart", startTimer);
dojo.connect(map, "onPanStart", startTimer);

var basemapLayer = new esri.layers.ArcGISTiledMapServiceLayer("http:
dojo.connect(basemapLayer, "onUpdate", endTimer);
map.addLayer(basemapLayer);

var operationalLayer = new esri.layers.ArcGISTiledMapServiceLayer("h
  { tileServers: [
      "http://globe2.arcwebservices.com/ArcGIS/rest/services/Referen
      "http://globe1.arcwebservices.com/ArcGIS/rest/services/Referen
    ]
  });
dojo.connect(operationalLayer, "onUpdate", endTimer);
map.addLayer(operationalLayer);
```

Basemap + Operational Layers

End Users View



Base Maps

- Types: Imagery, Streets, Terrain, any type that provides contextually useful information for your operational layers
- Generally Cached
- Usually does not change frequently

Base Maps should be beautiful!

Operational Layers

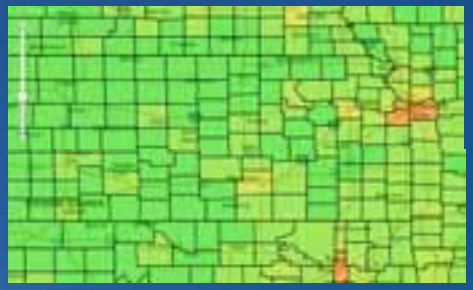
- May change frequently
- Display operational layer:
  - as client-side graphics (result of a task operation)
  - as a dynamic map service (MSD based map service)
  - as a cached map service

Display the operational layer as client-side graphics

- Query on demand as user pans and zooms around the map, query a layer in a map service and display results
  - Only load the features that you need
  - Takes advantage of the CPUs available on your users' machines

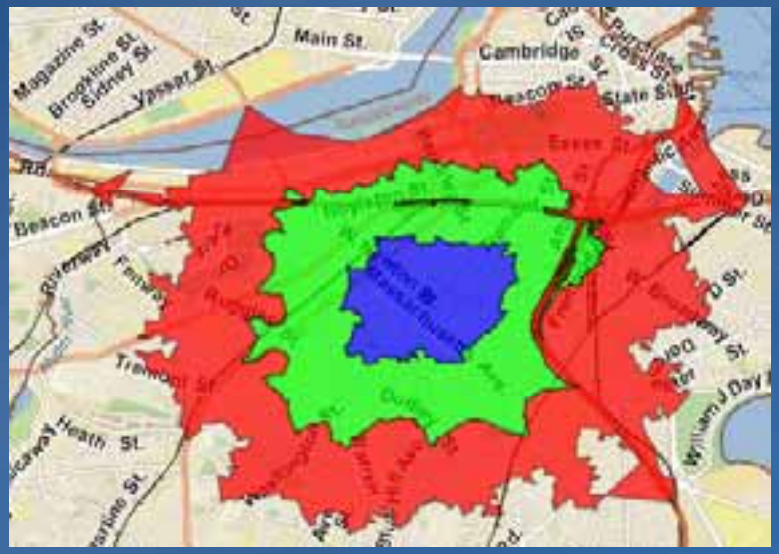
- Only display features when they are requested by clients

- Expose display options and custom symbology to the clients.



Display the operational layer as client-side graphics

- **Graphic Feature Results from ANY ArcGIS Server service**
  - **Graphic Feature Results from a Geoprocessing Task**



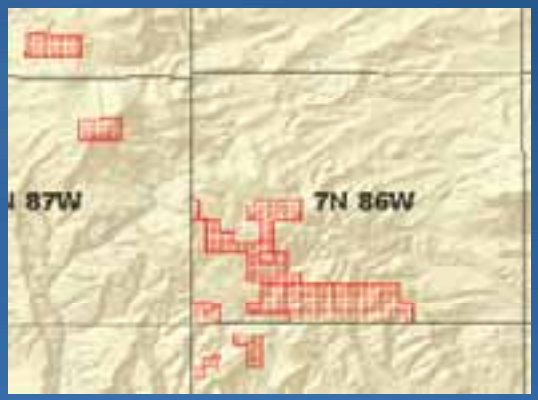
Display the operational layer as client-side graphics -- considerations

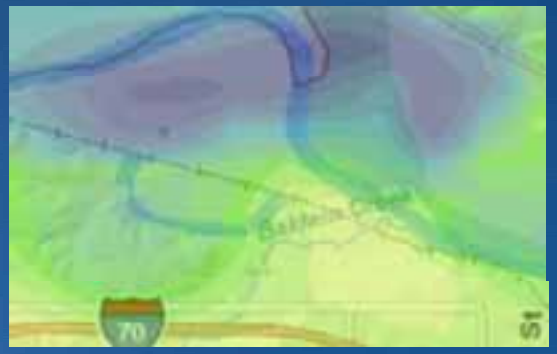
- Limit to number of client side graphics that can be supported.
- Clustering of point features on the client

- No native Labeling in web map
  - Can place text, but not label

Display the operational layer as a dynamic map service

- Client requests an image at the requested extent, with the selected layers and layer definitions
- Server draws the map
- Web map overlays the returned image on the base map
- Web map can also use GP Raster Results drawn by the map service

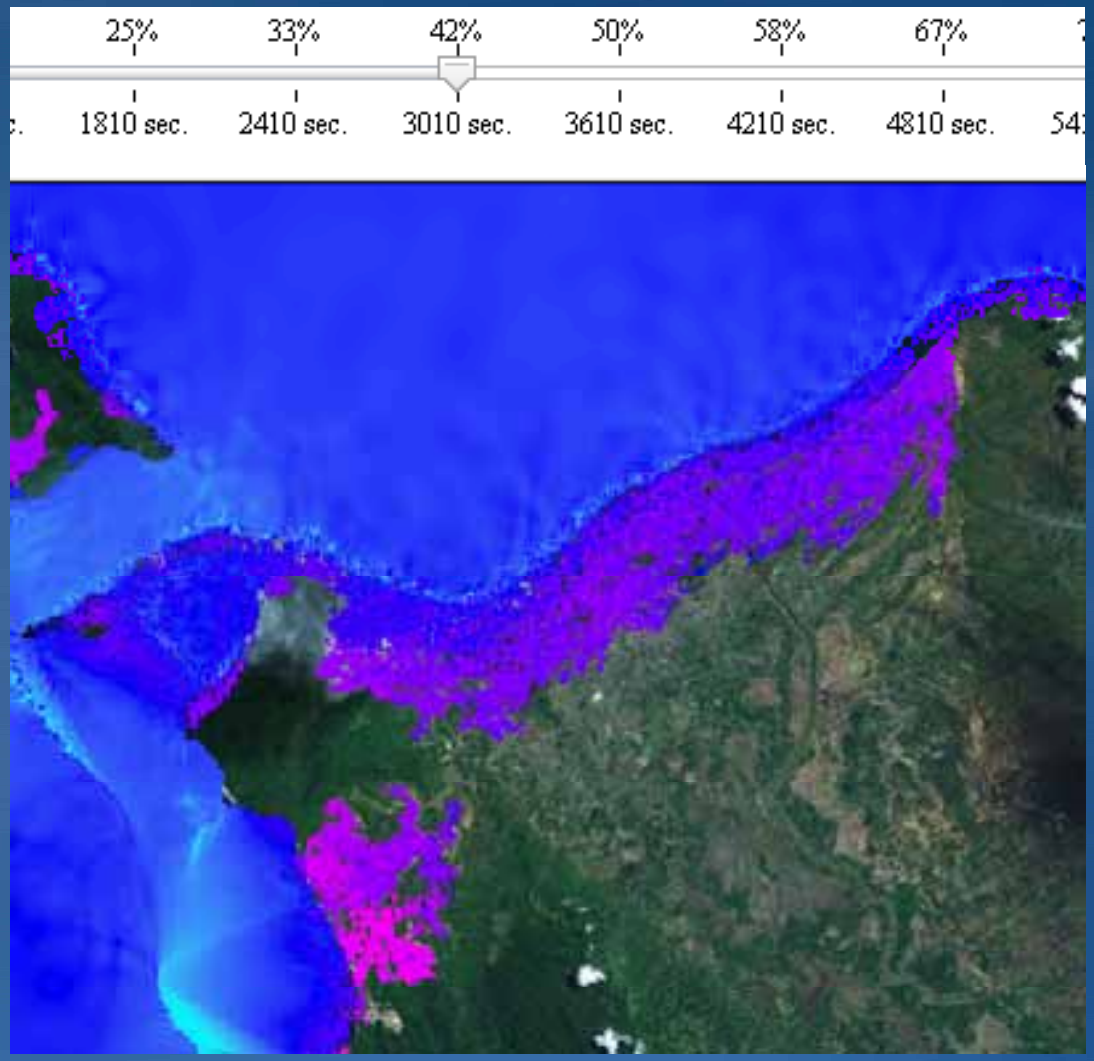




Display the operational layer as a dynamic map service

- Use Cases
  - Data changes frequently
  - Too much data to push to the client
  - Need to support individual map service layer selection (animation)
  - Need to support layer definitions

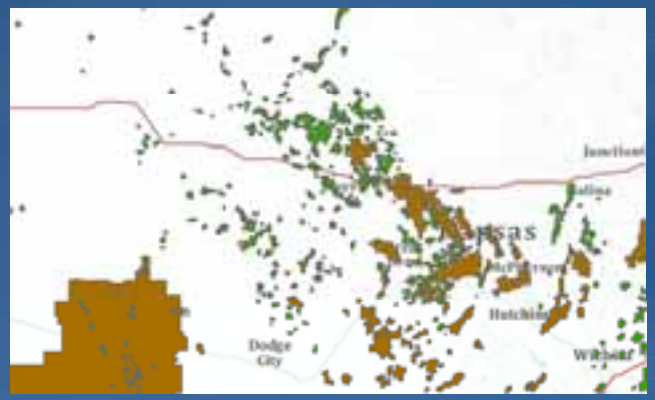
Display the operational layer as a dynamic map service

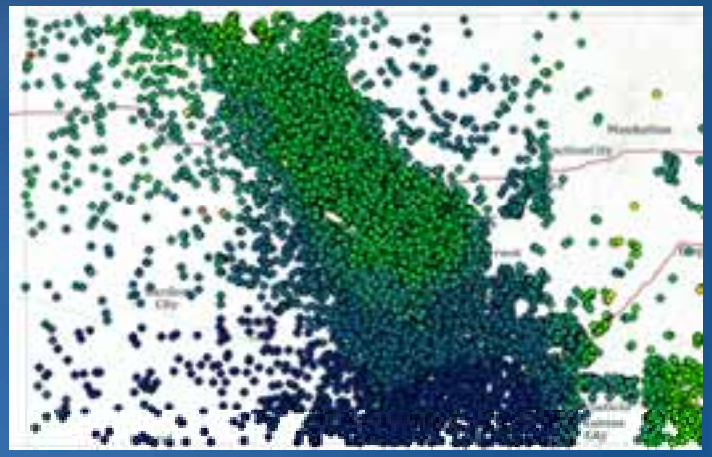


Animation

Display the operational layer as a dynamic map service

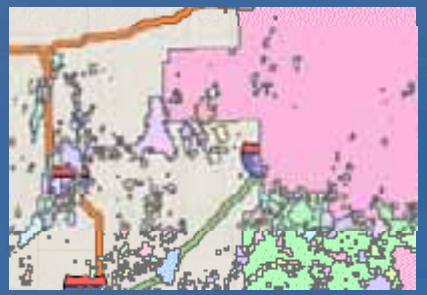
- Map requests with layer definitions

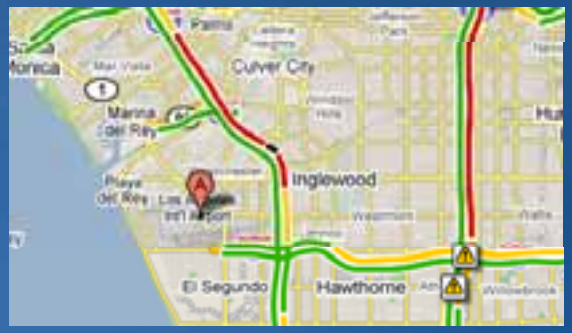




Display the operational layer as a tiled map service

- Operational layer is a cached map service layer
- High performance
  - Browser requests pre-made tiles
- Good if data changes infrequently or can be easily created

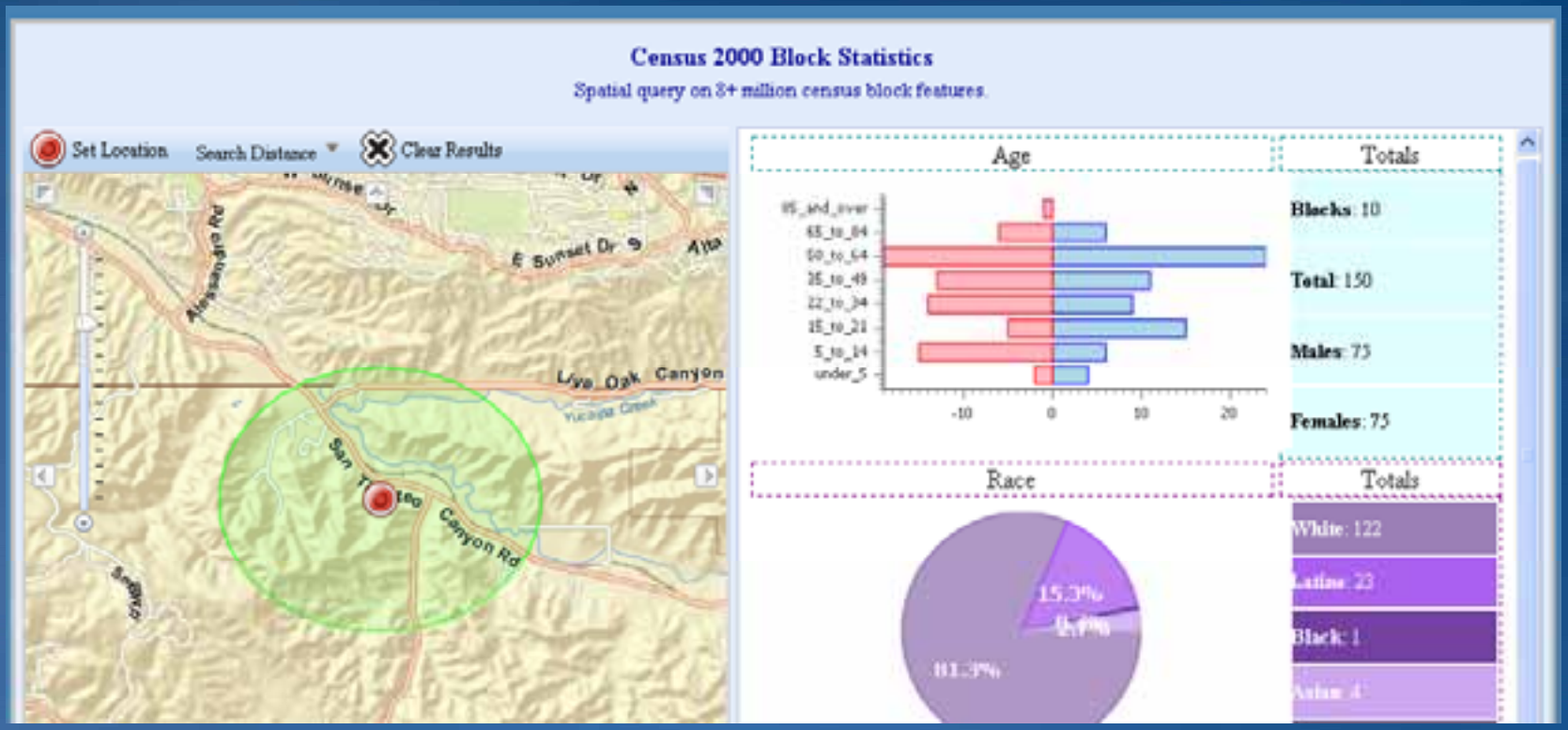




Other Examples of sites that make use of the Base Map + Operational Layer pattern

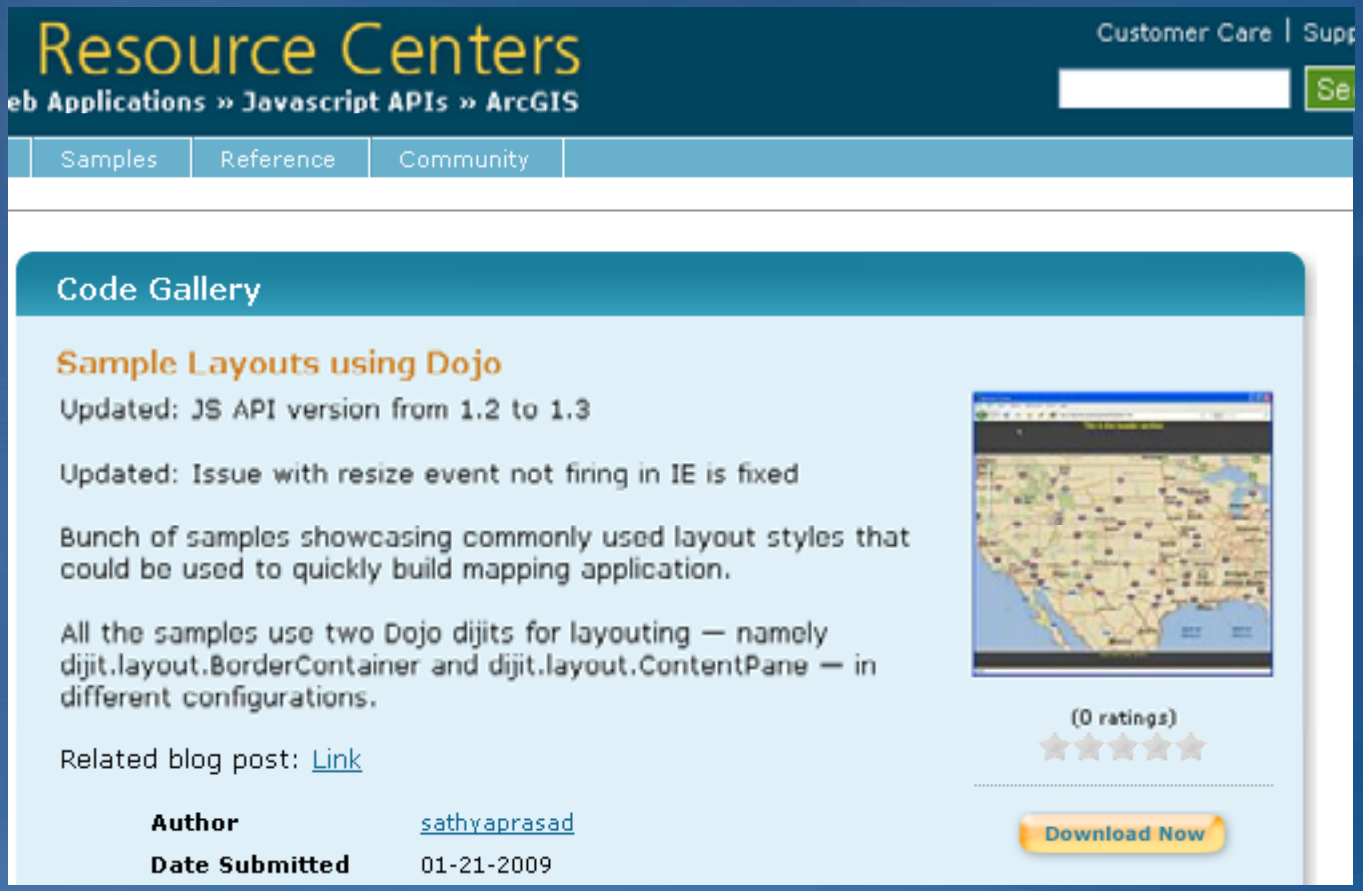
- Do what is best for your application
  - Pre-compute
  - Use type of operational layer that best meets your needs

Application layout and design



Application layout and design

- **Work with the grain of the JavaScript Framework that you are using**



Application layout and design

- Give the user feedback
  - From Guardian Map

- Make use of Animation Rate and Duration
- Learn what other non mapping sites are doing
- User experience is key!

Questions and Discussion