



Building Mobile Apps with the ArcGIS API for JavaScript

Javier Gutierrez

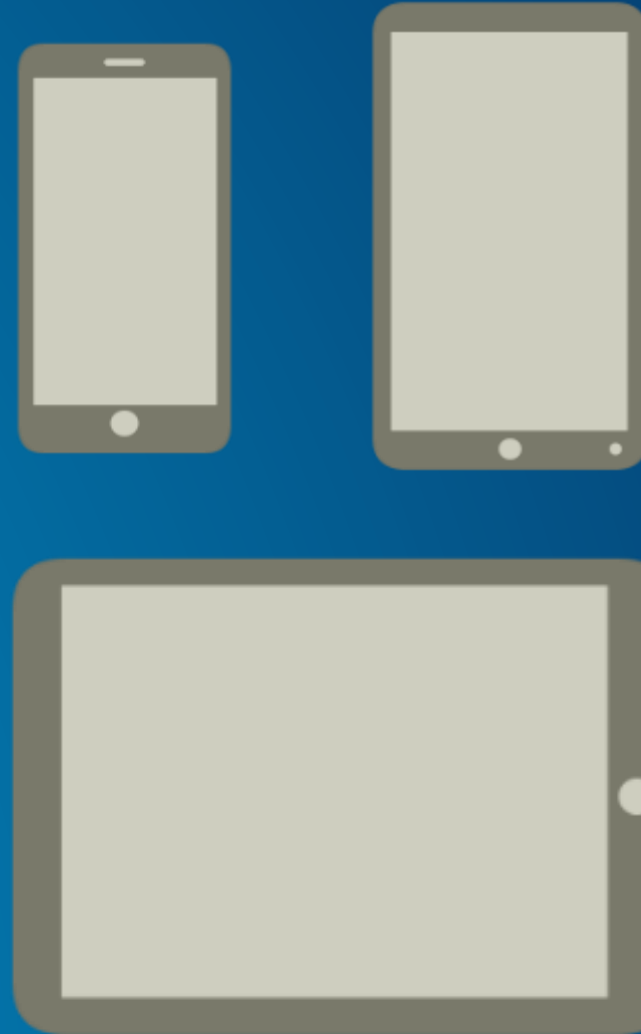
Esri Middle East and Africa
User Conference & Developer Summit

#esrijs

Why mobile web apps

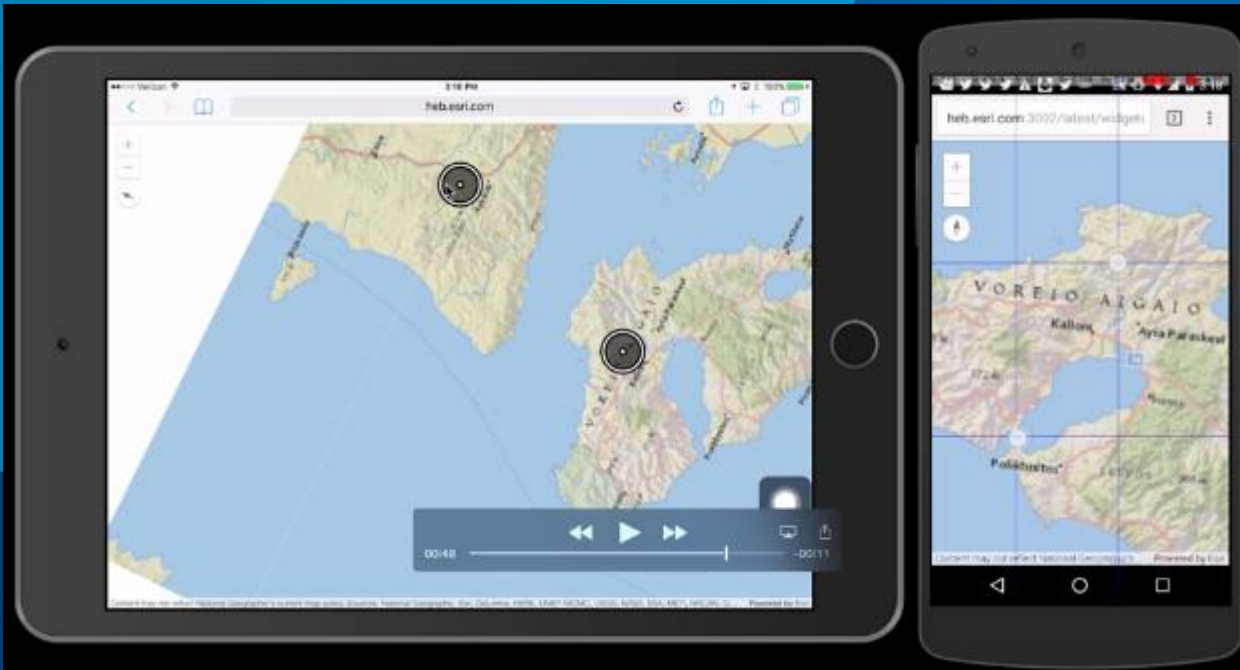
- **Write once, run everywhere**
 - User experience
 - Cost to build and maintain
 - Time to market
 - Fit for development organization
- **Tradeoffs**
 - Native SDKs become a black box
 - Potential limitations in native feature development
 - Potential performance issues

Expectations



Capabilities in the ArcGIS API

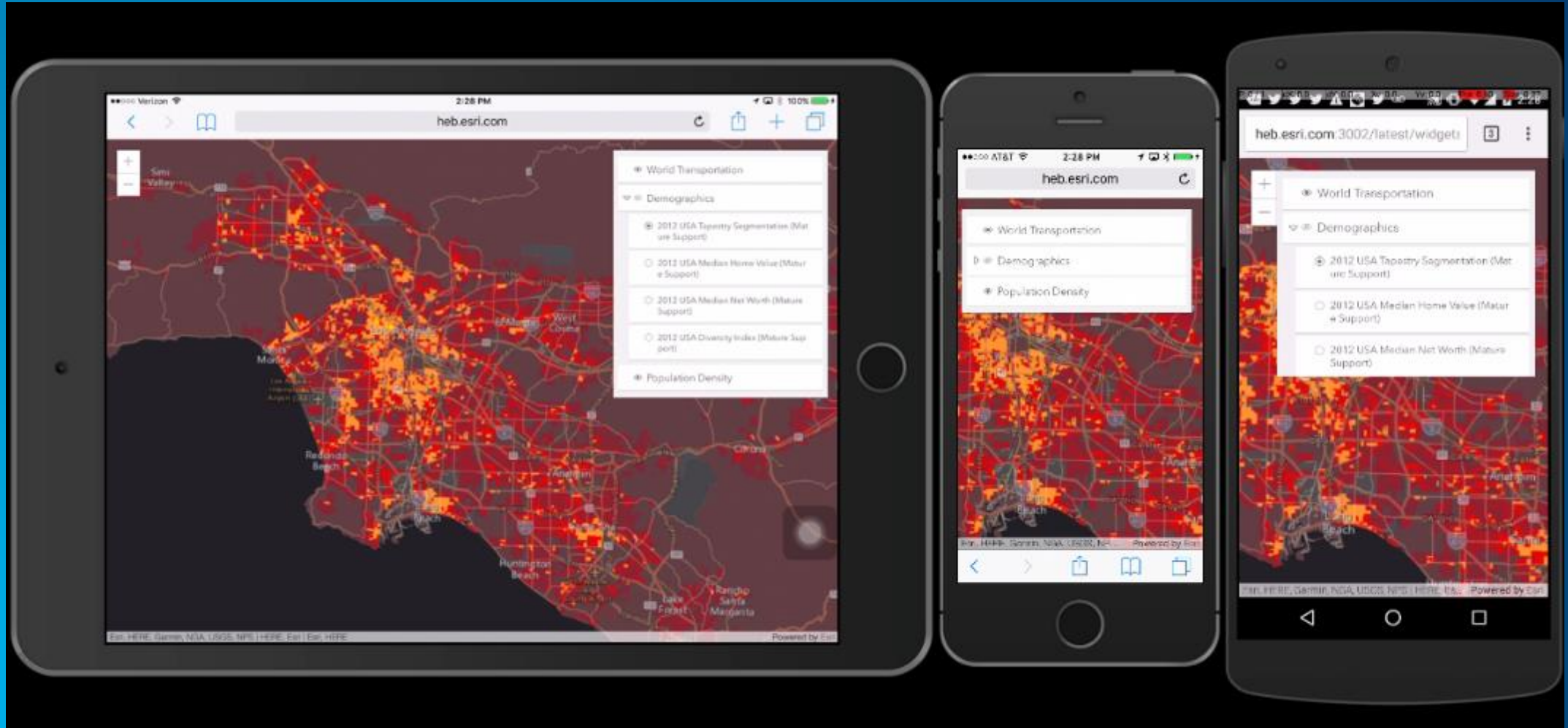
Mobile First | Responsive design



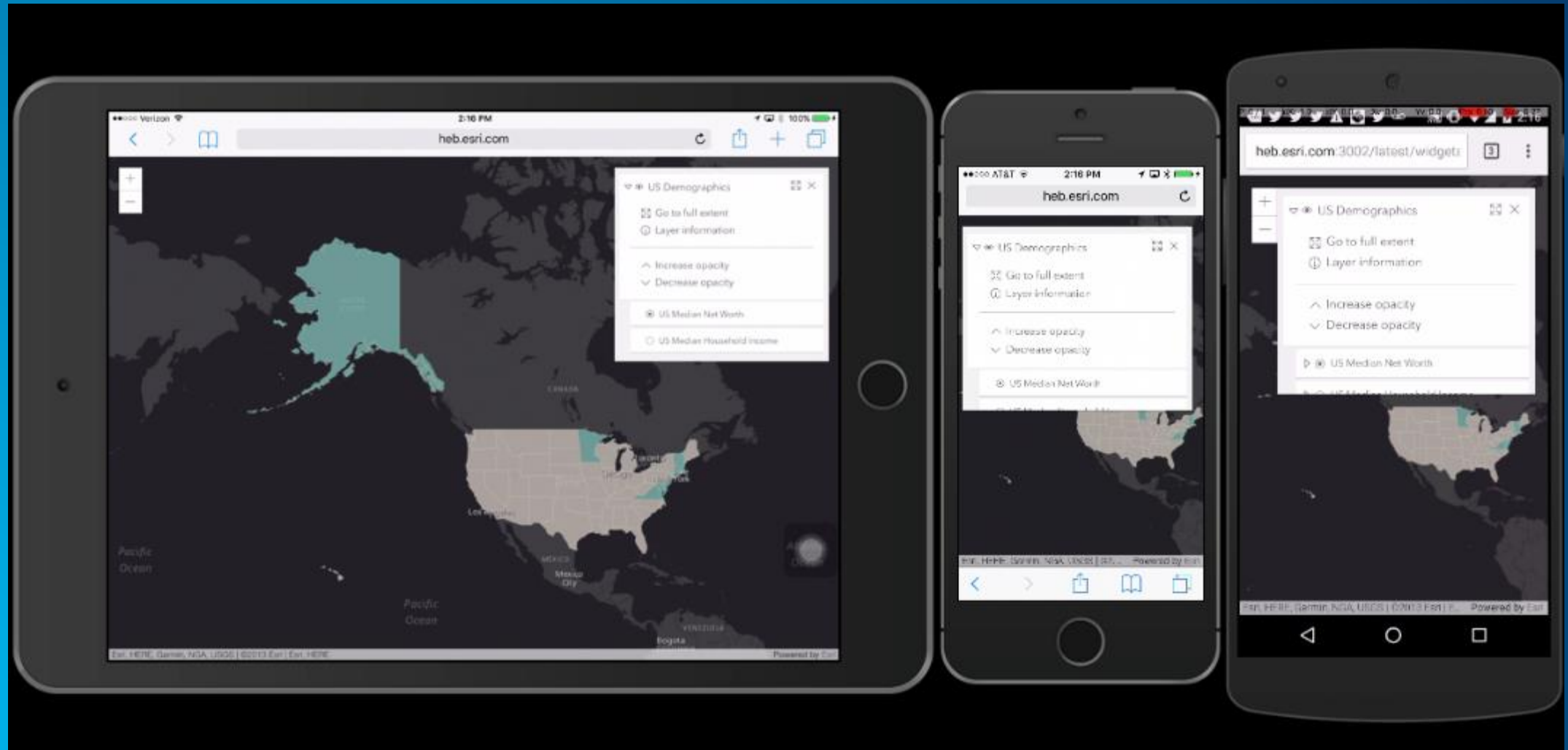
Touch

Demo

LayerList



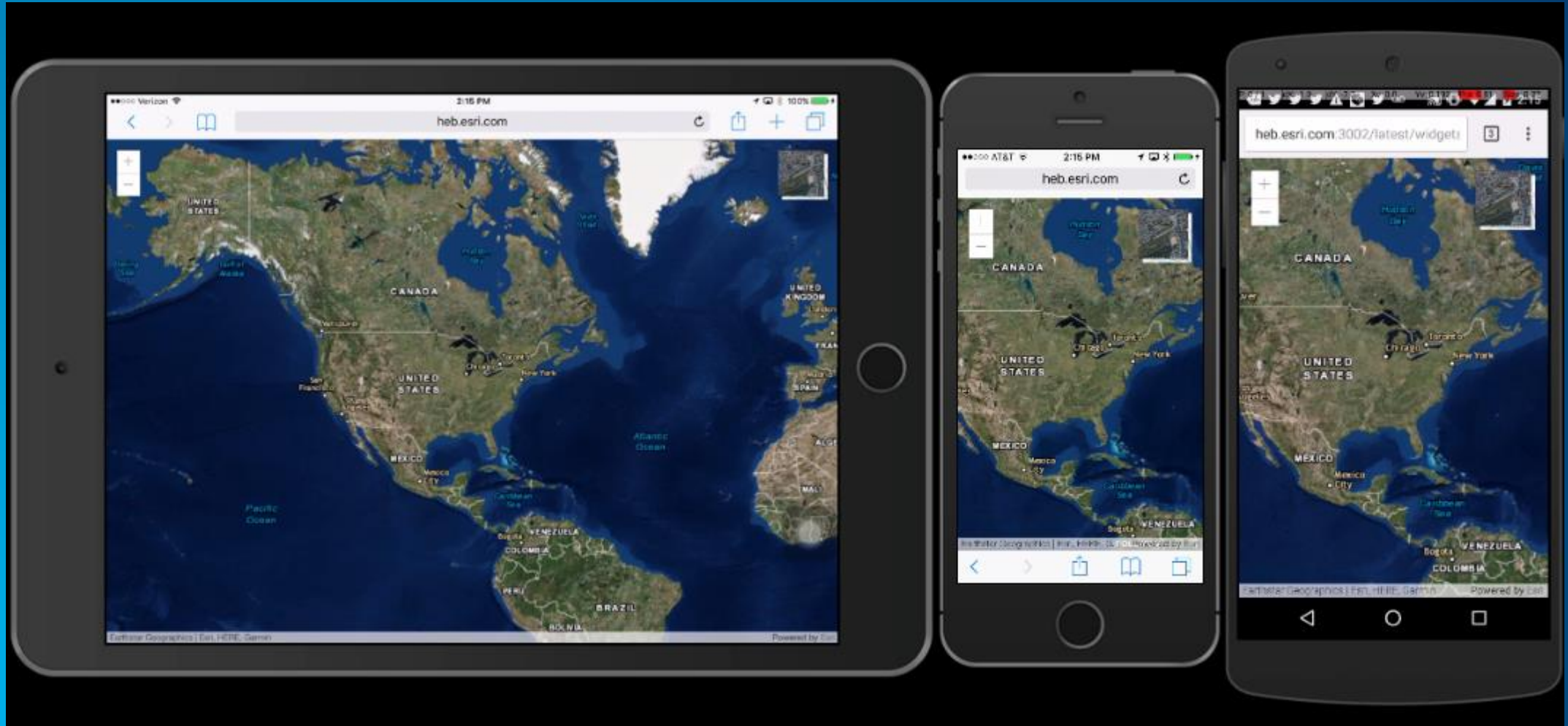
LayerList action



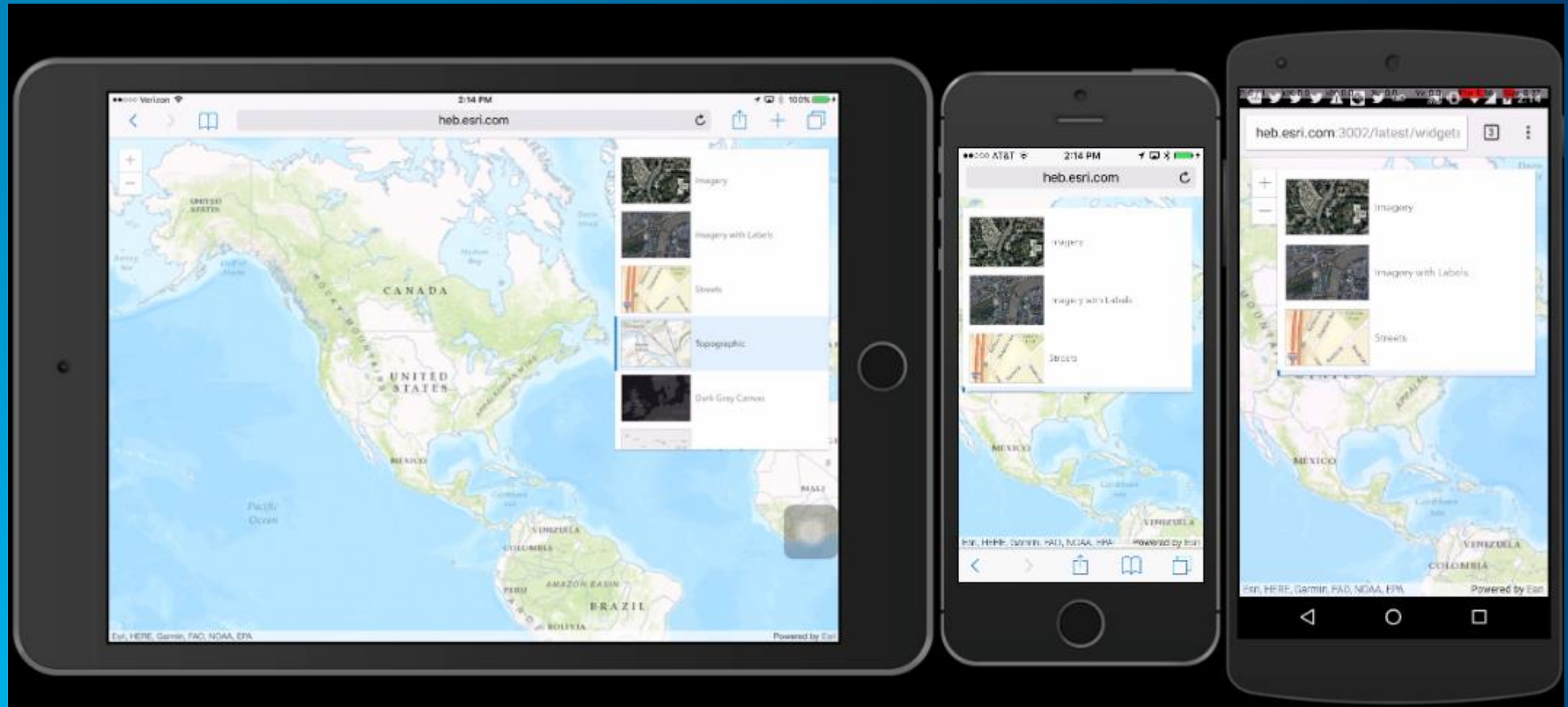
BasemapToggle

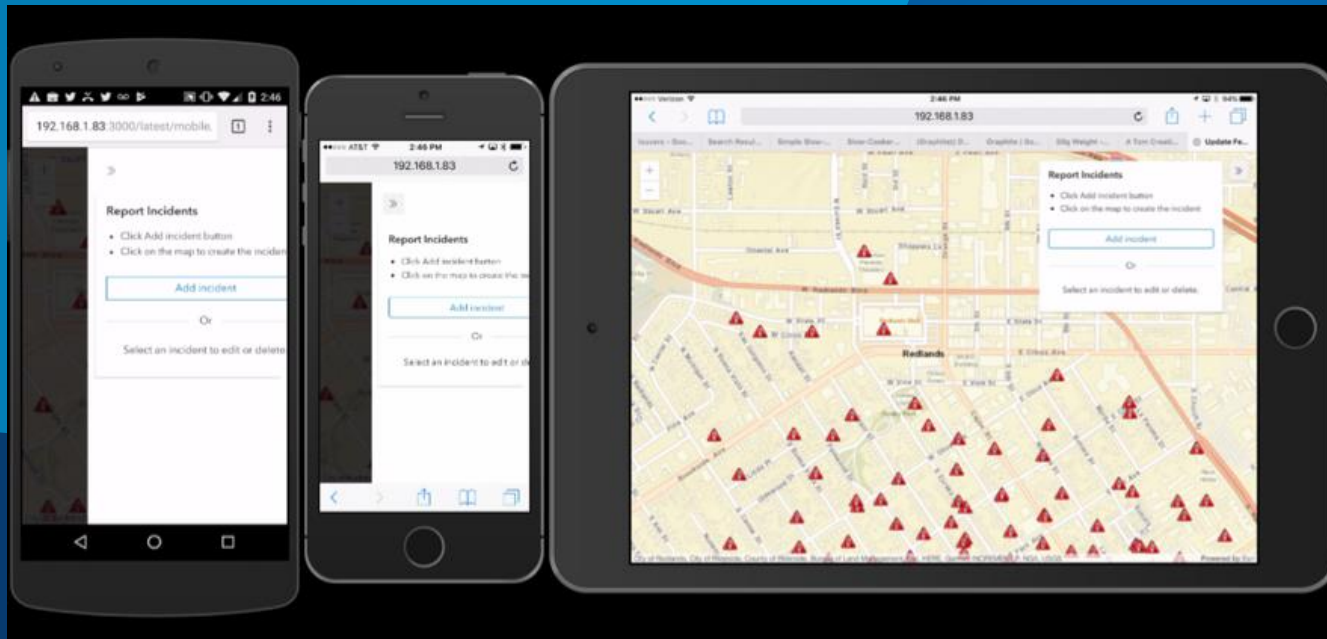


BasemapToggle



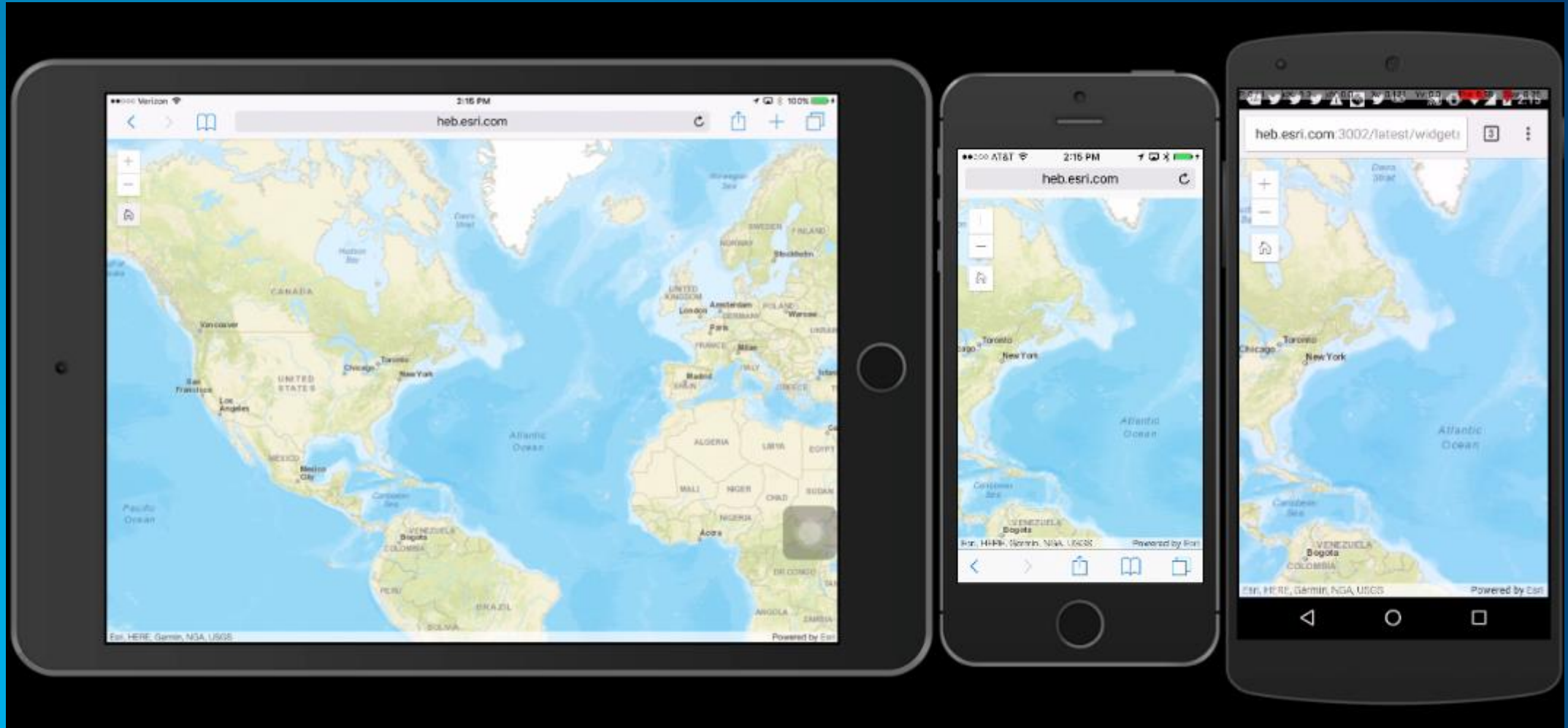
BasemapGallery



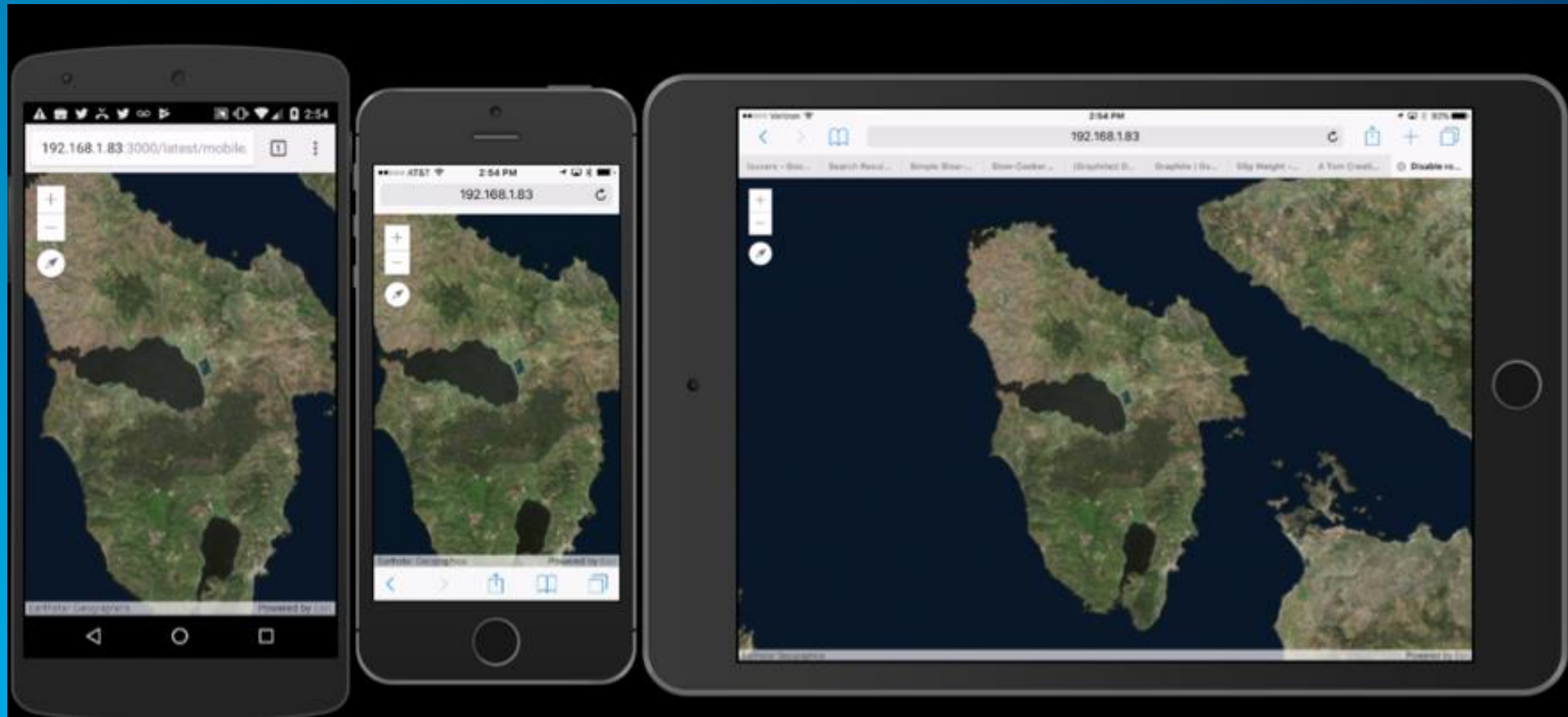


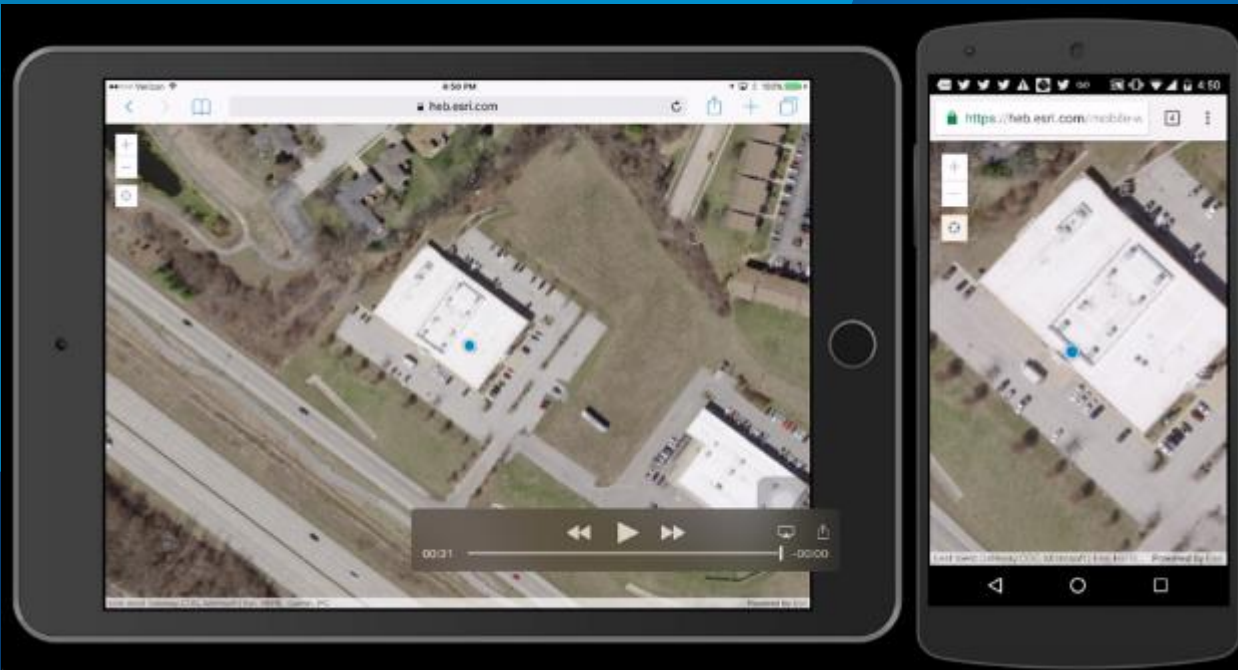
Expand Demo

Home



Compass

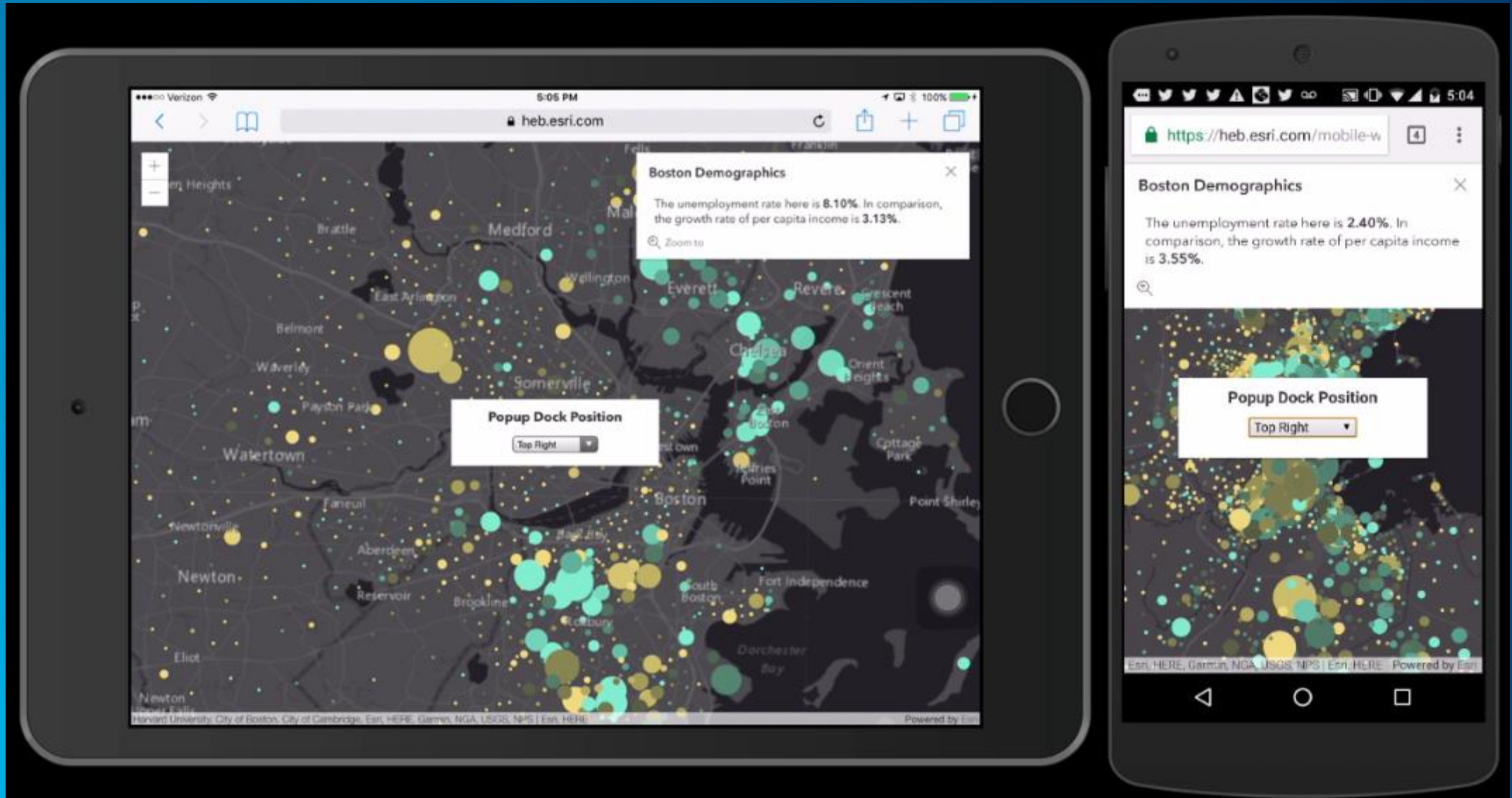




Locate

Demo

Popups



Some more tips to get started

Calcite Maps

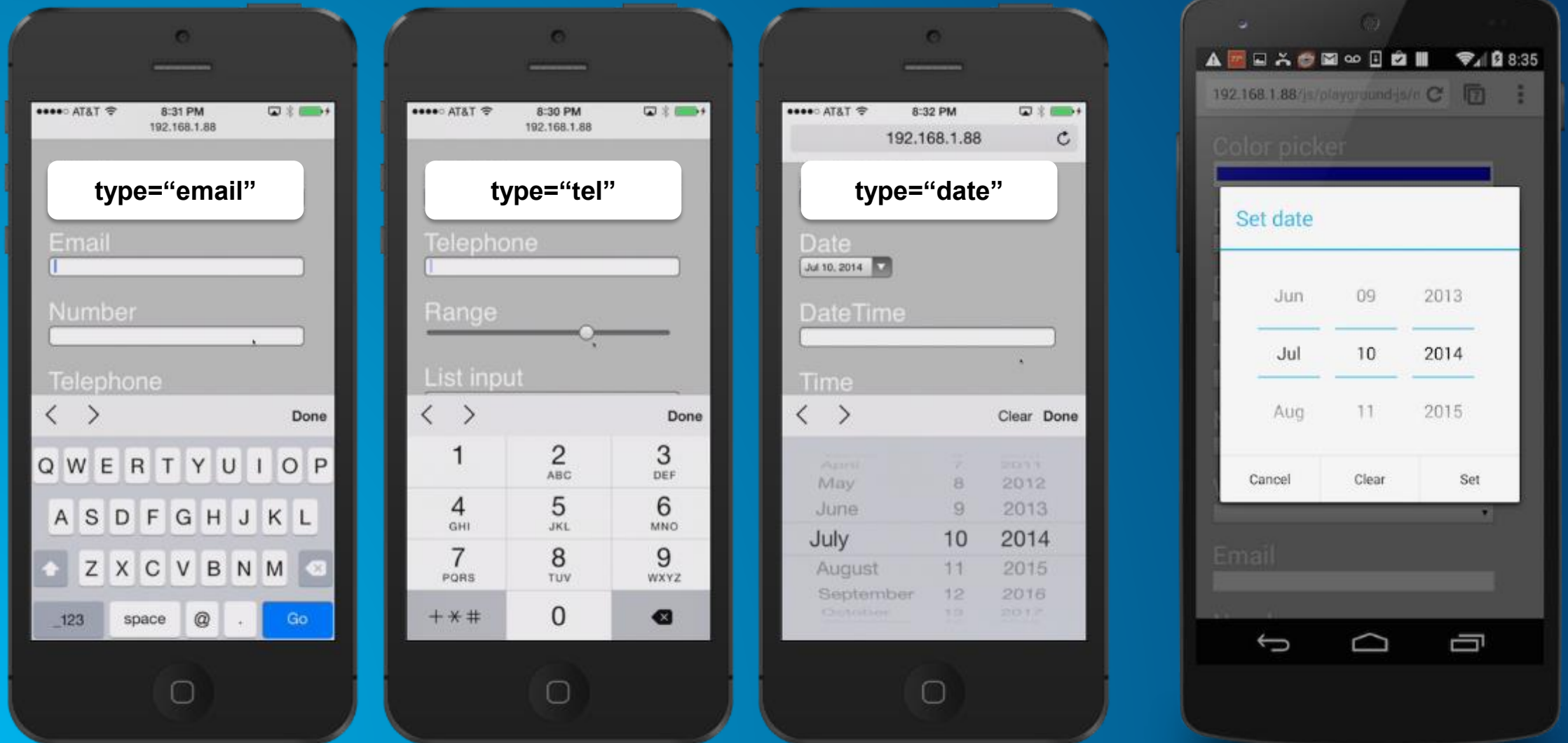
- Esri Project
- Not part of the ArcGIS API for JavaScript, but uses it
- Starting point for responsive UI
- With themes

<https://github.com/Esri/calcite-maps>

HTML5 components

- When not using a modern, responsive, mobile-first toolkit like calcite/bootstrap
- Or not using components from ArcGIS API for JS
 - Use HTML5 default components to ensure best experience on mobile

HTML5 input types



CSS Media Queries

- **Detect specific device/browser characteristics and parameters**

<u>width</u>	Width of the viewport
<u>height</u>	Height of the viewport
<u>aspect-ratio</u>	Width-to-height aspect ratio of the viewport
<u>orientation</u>	Orientation of the viewport
<u>resolution</u>	Pixel density of the output device

- **Used for**
 - **conditionally applying styles (@media and @import)**
 - **conditionally loading <link>, <source> and other HTML elements**
- **Can be used in HTML/CSS code, and also in JavaScript**

Mobile-first with media queries

- Means designing for mobile before designing for desktop or any other device
 - Better UI design & Better performance
- Mobile first design in CSS
 - DON'T change style when window is smaller than 800px
 - DO change style when window is larger than 800px

DEMO

Managing initial load times

Initial loading in a mobile browser

- Often misunderstood
- Many things happening
- Vulnerable to timing issues
- Consider user's experience



Initial loading in a mobile browser

CSS

3rd party libraries

Frameworks

Framework plugins

ArcGIS JS API

GIS queries

GET map tiles



Initial loading in a mobile browser

CSS

3rd party libraries

Frameworks

Framework plugins

ArcGIS JS API

GIS queries

GET map tiles



Initial loading in a mobile browser

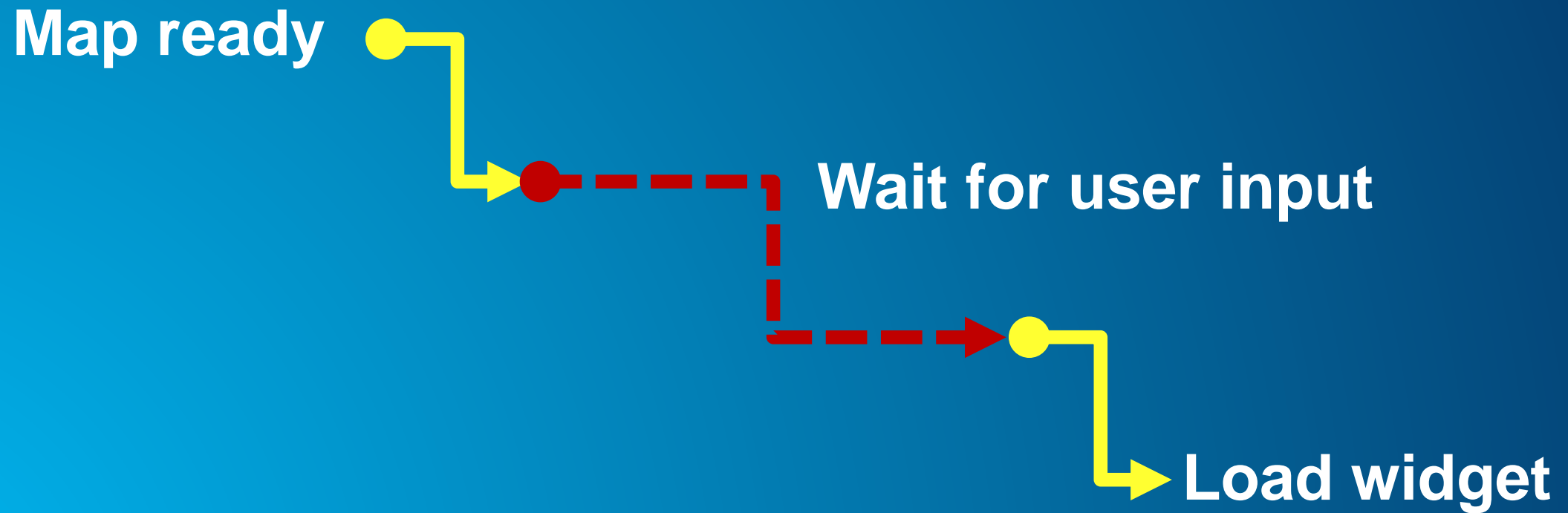
Wait for framework initialization

- In Angular: implements OnInit
- In Cordova: 'deviceready' event
- In ArcGIS: dojo/domReady!

Consider synchronous patterns



Lazy load modules and libraries



Lazy load demo

```
function lazyLoadPointCloudLayer(){  
  require([  
    "esri/layers/PointCloudLayer"  
  ], function(PointCloudLayer) {  
    var plc = new PointCloudLayer();  
  });  
}
```

Demo

Reserve async for after app load

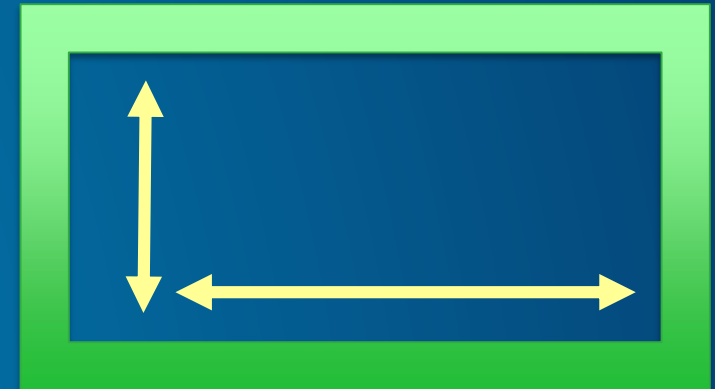
Wait for framework initialization

- In Angular: implements OnInit
- In Cordova: 'deviceready' event
- In ArcGIS: dojo/domReady!

MapView ready?

The View has a Map
View container size is greater than zero
View has a spatial reference

[API Reference MapView.ready](#)



Loadable Patterns

Any resource that handles async data access

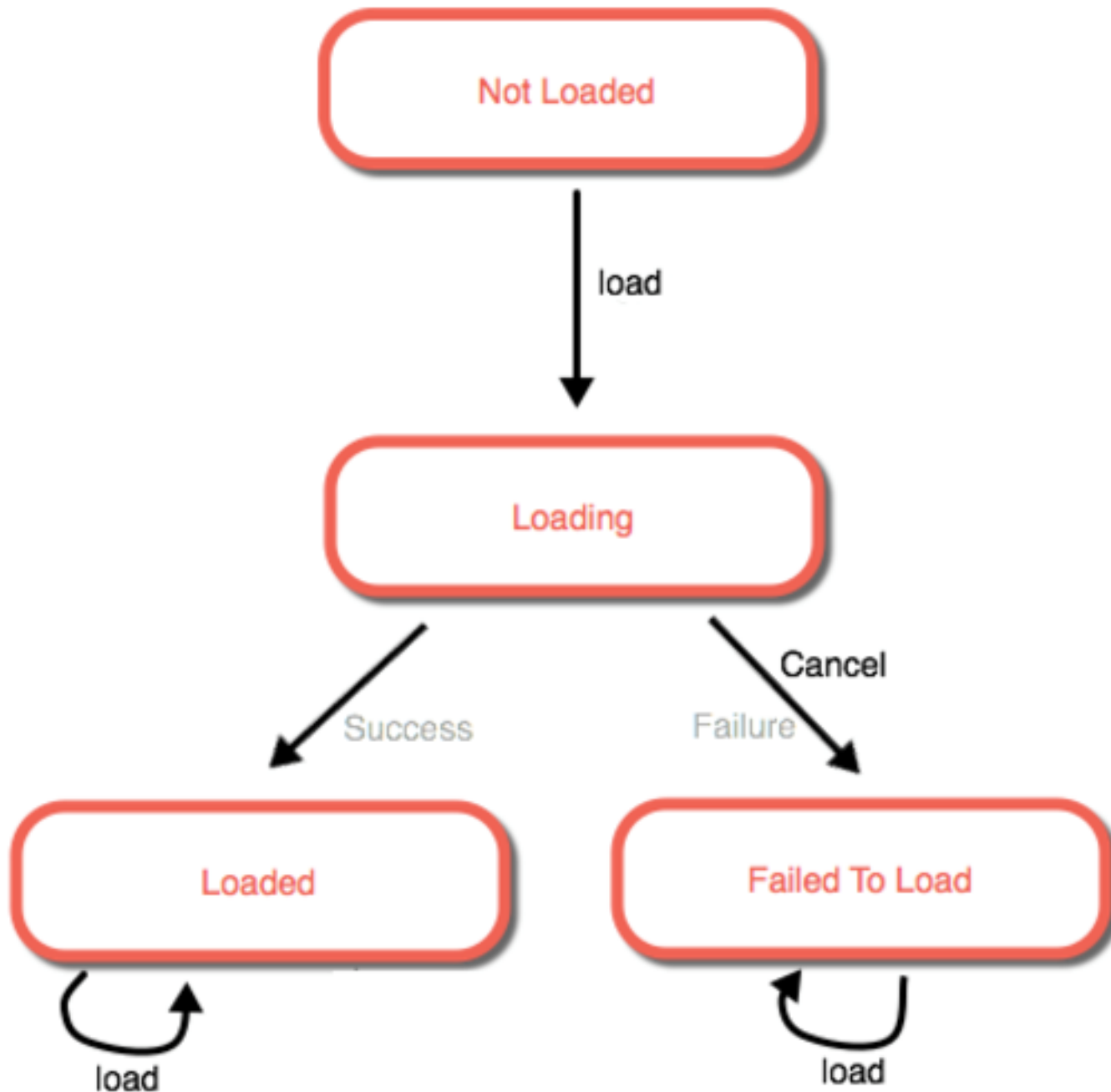
Examples: feature layers, tasks, web maps

`not-loaded`

`loadStatus` `String` `readonly`

Represents the status of a `load` operation.

Value	Description
<code>not-loaded</code>	The object's resources have not loaded.
<code>loading</code>	The object's resources are currently loading.
<code>loaded</code>	The object's resources have loaded without errors.
<code>failed</code>	The object's resources failed to load. See <code>loadError</code> for more details.



API reference
[WebMap.loadStatus](#)

load() example

```
var webmap = new WebMap({
  portalItem: { // autocasts as new PortalItem()
    id: "f2e9b762544945f390ca4ac3671cfa72"
  }
});

var view = new MapView({
  //map: webmap,
  container: "viewDiv"
});

webmap.load().then(function(r){
  view.map = webmap;
  console.log(r);
  console.log("Load Status: " + webmap.loadStatus)
}).otherwise(function(err){
  console.log("Problem loading webmap: " + error);
});
```

Demo

Splash Screens

Distract end user from long load times
Gives you control over when main app visible
Wait for `MapView.then()`

<https://codepen.io/jgut028/pen/wRKmQq>

Working with Cordova and Ionic

Hybrid apps Mobile

Cordova/Phonegap

- Wait for “deviceready” event
- <https://github.com/Esri/quickstart-map-phonegap>

Ionic and loading ArcGIS JavaScript modules

Many helper libraries!

- [angular-esri-loader](#) (Angular 4+, wraps esri-loader)
- [esri-loader](#) (platform agnostic)
- [Ionic2-esri-map](#) (Sample app)
- [ArcGIS TypeScript definitions](#)

Working with Frameworks

Mobile browser-only

Other frameworks

- [Plain Bootstrap](#)
- [calcite-maps](#)
- [ember-esri-loader](#)
- [esri-loader-react](#)
- [esri-system-js](#)

Debugging and testing

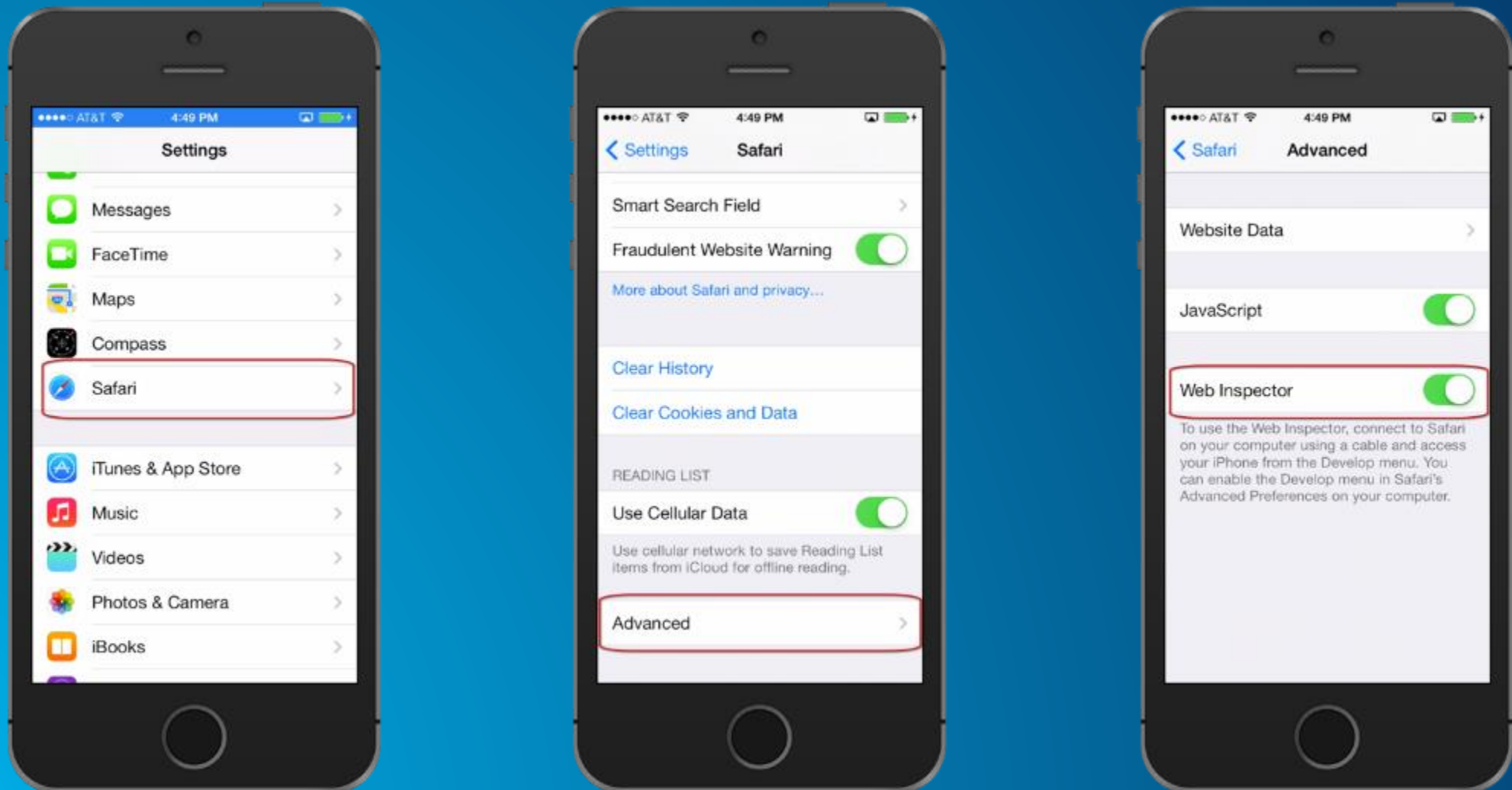
Physical devices



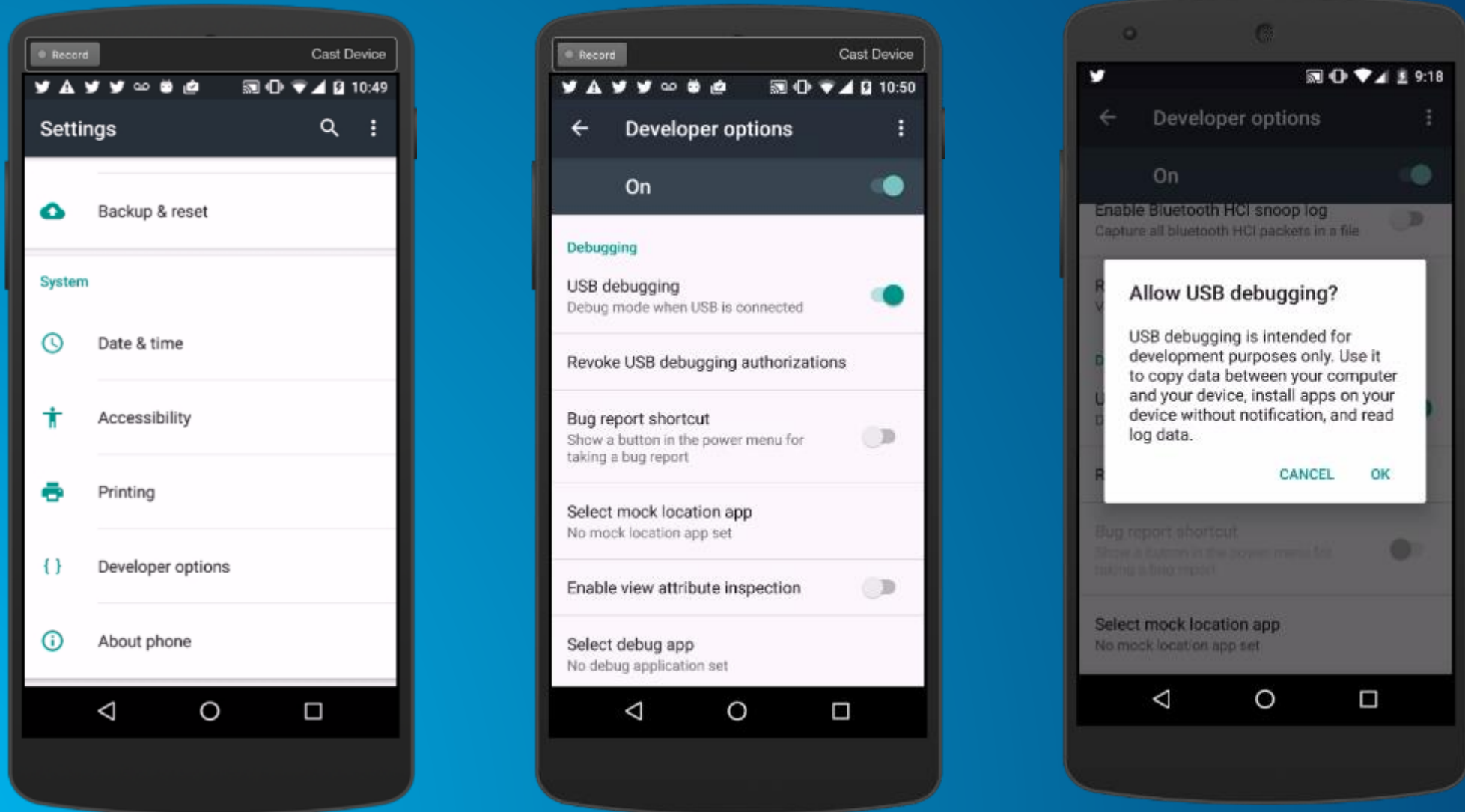
Remote debugging

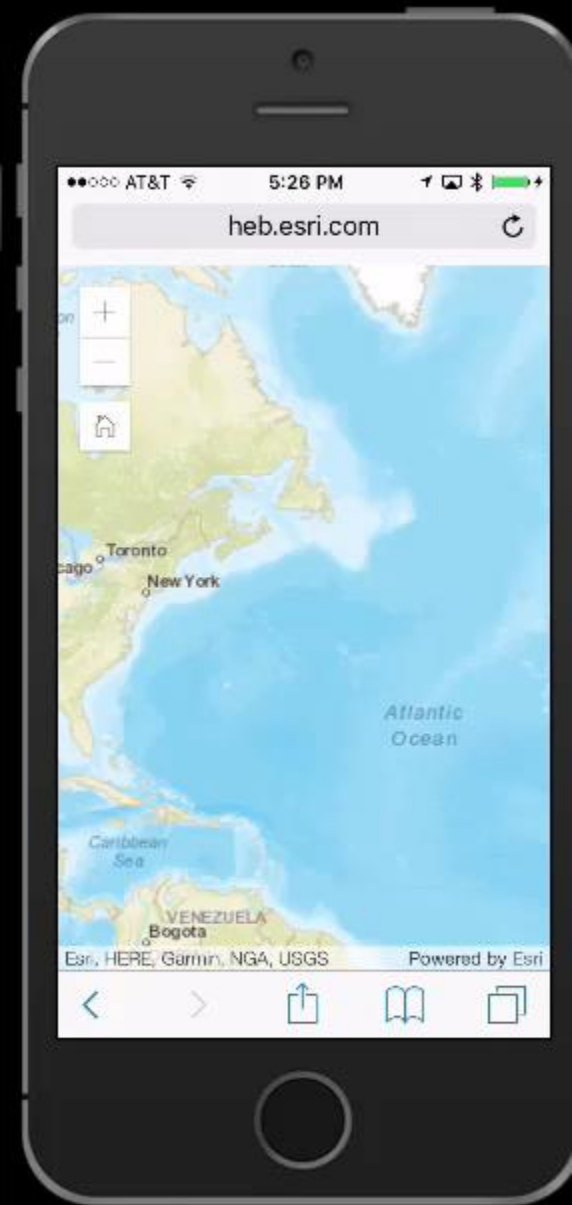
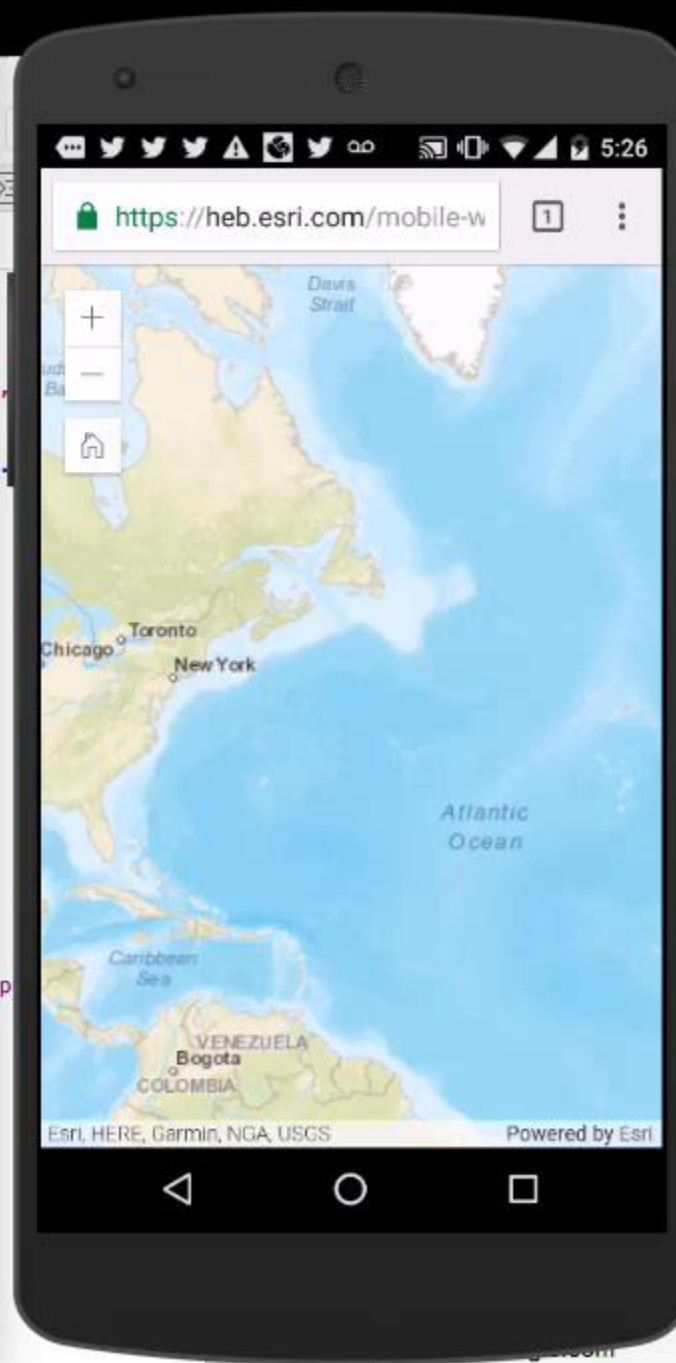
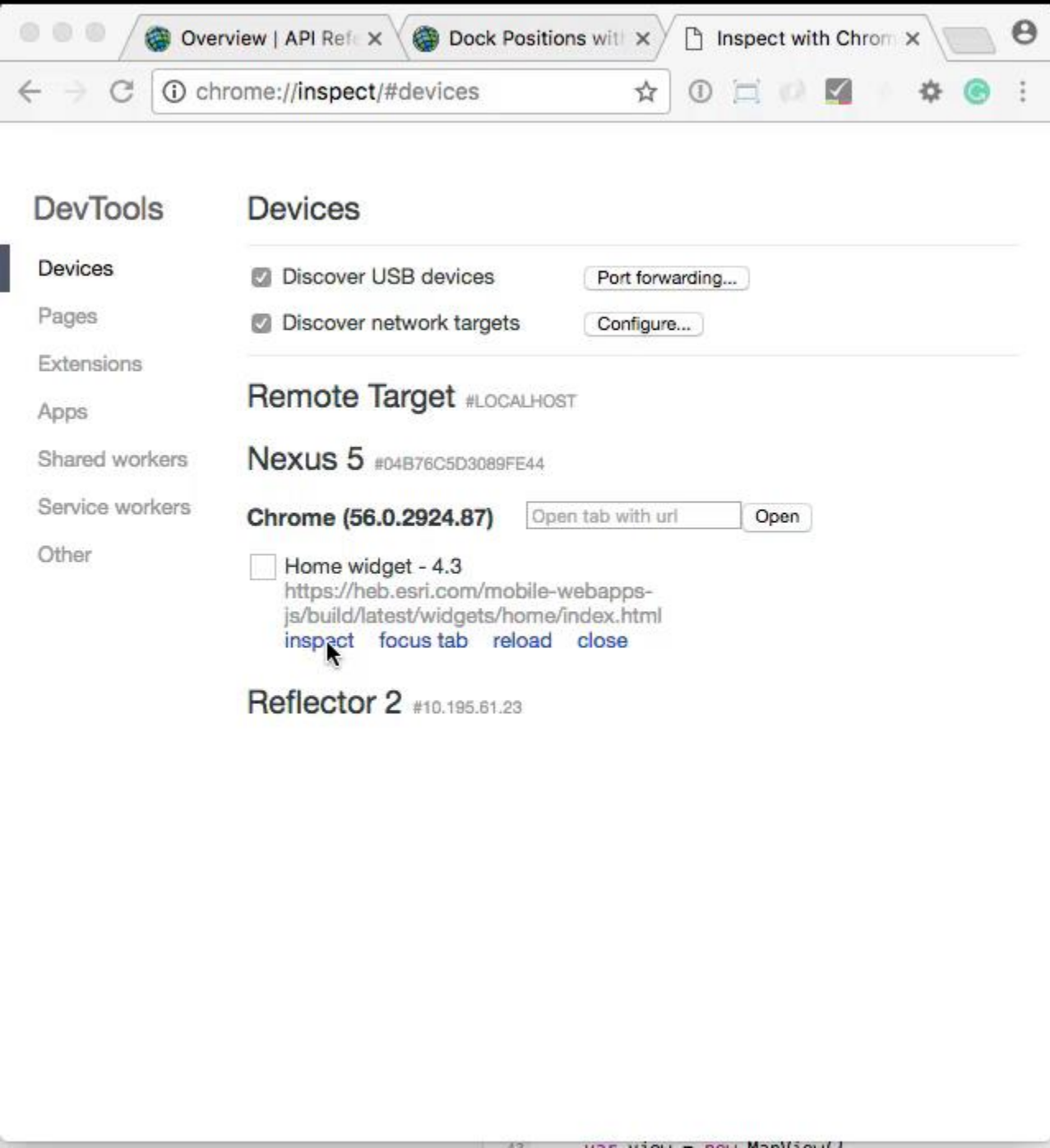
- Safari Web Inspector Remote
- Google Chrome remote debugging
- <http://livereload.com/>
- Browser-sync

Enable iOS remote web inspection



Enable Android remote web inspection





Working with 3D on Mobile

Questions?



esri

THE
SCIENCE
OF
WHERE