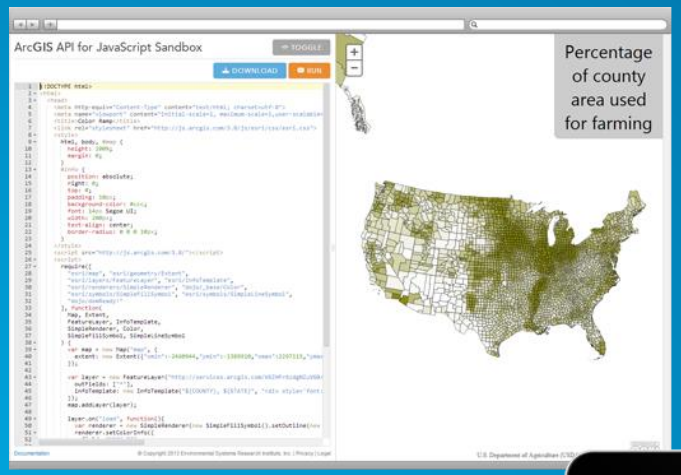
Introduction

Web AppBuilder for ArcGIS

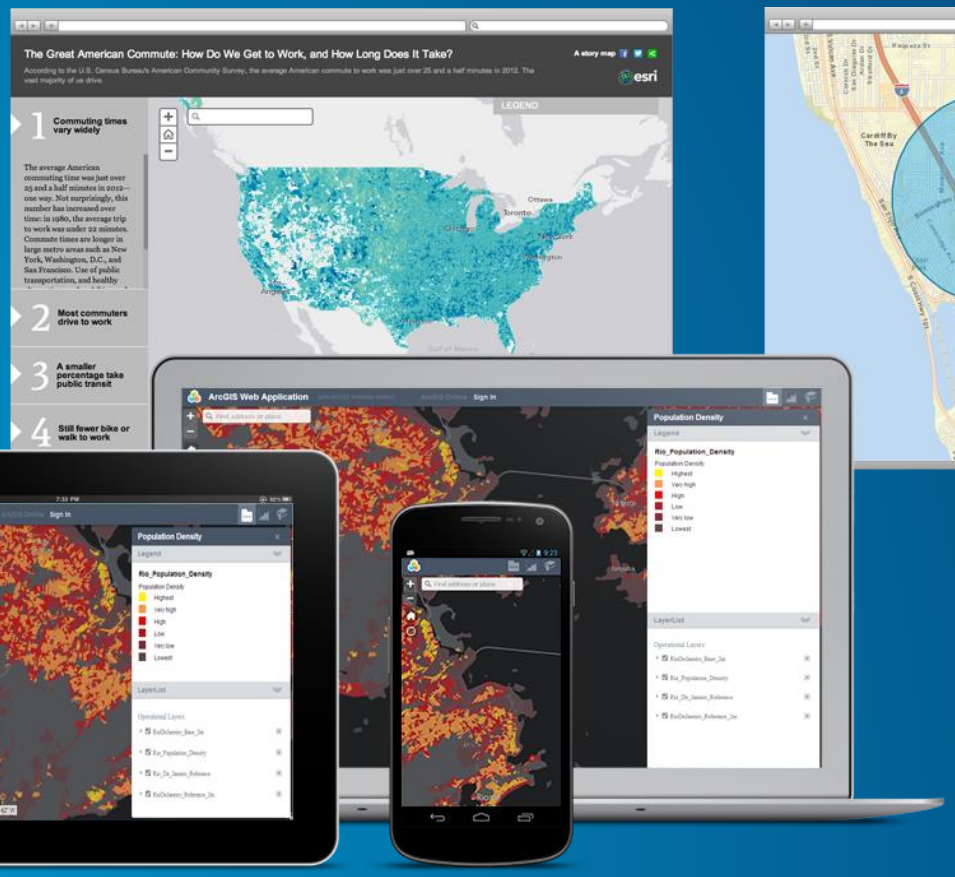
Building Web Apps for Your Organization

Using the ArcGIS API for JavaScript

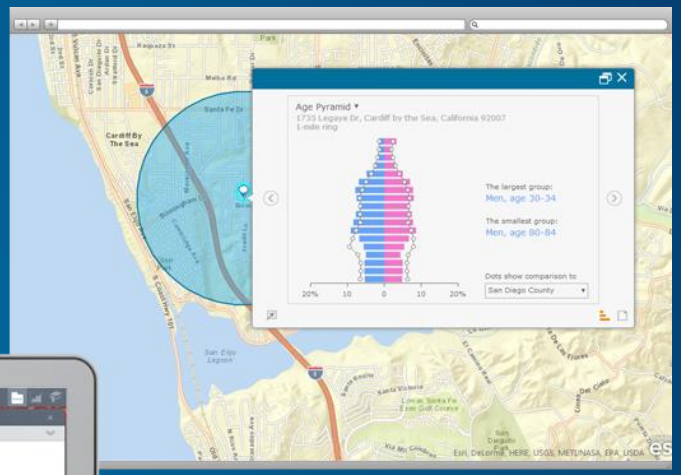
Samples



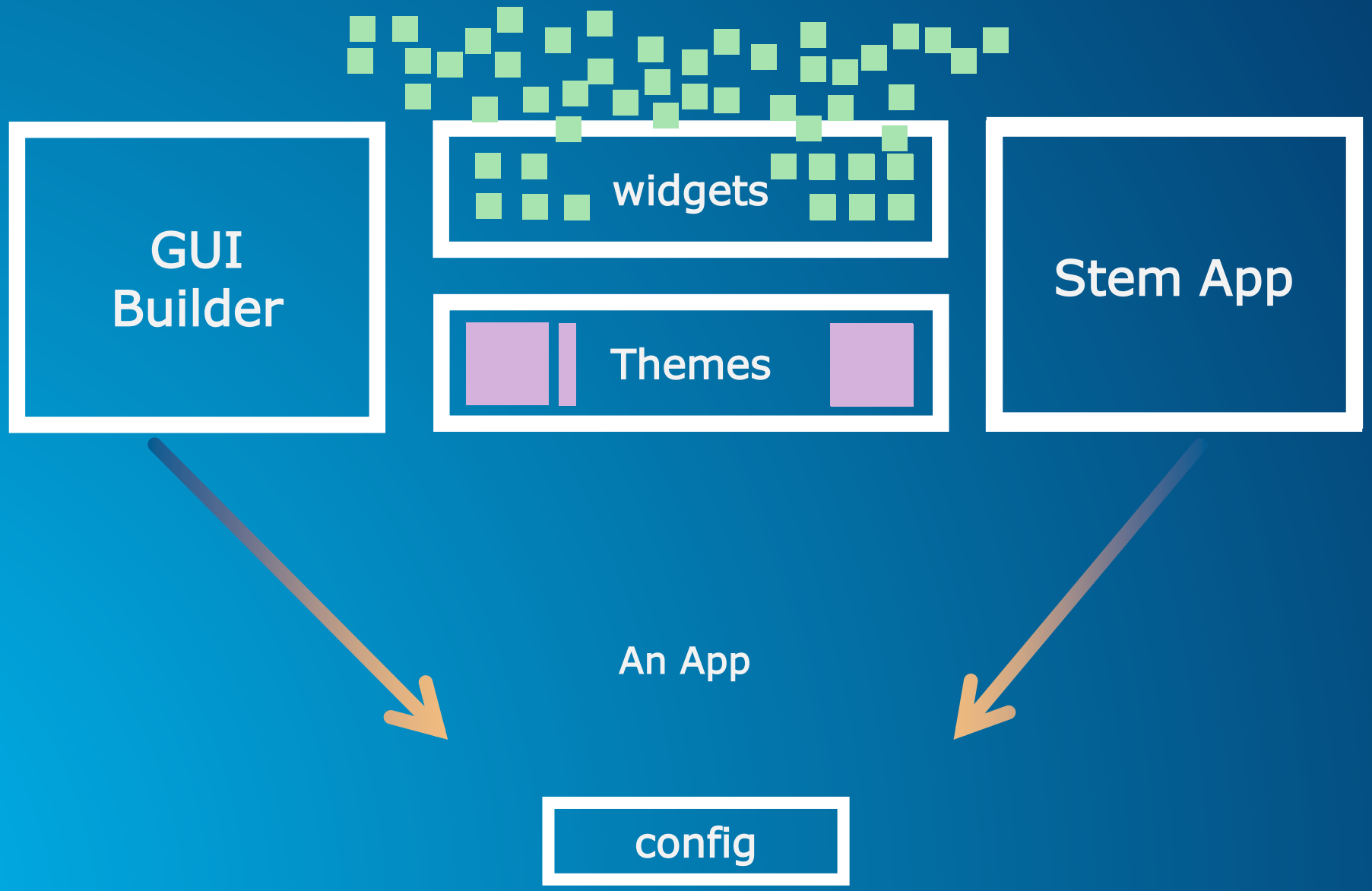
Configurable Apps & Builders



Widgets



App Builder Concept



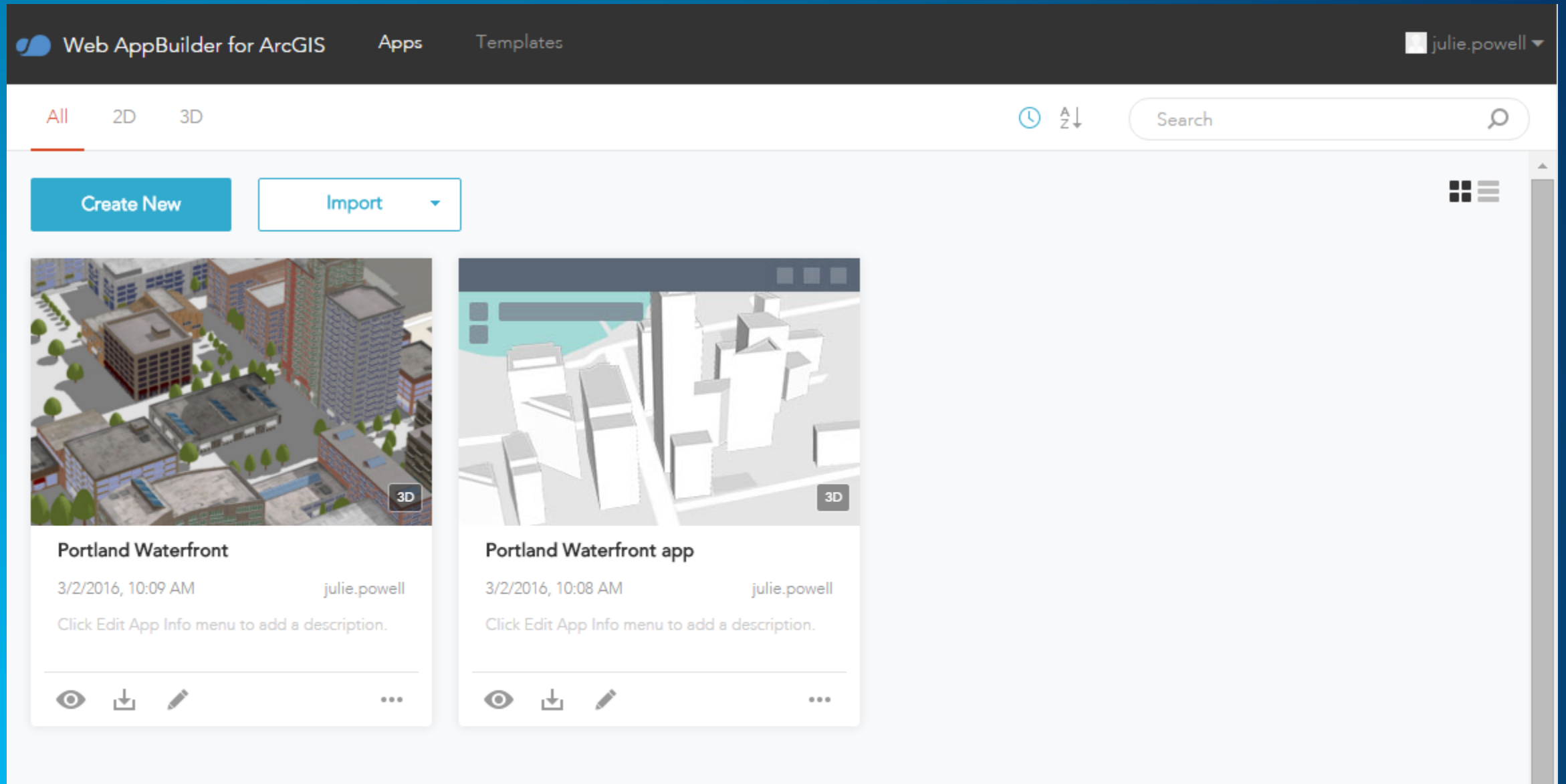
Widget

- Execution at run time
- Configure-in, not cut/paste
- Self sufficient and distributable
- Need container, no coding block
- Has programing framework of container

Theme

- Applied at run time
- Configure-in, not modify css
- Need container
- Self sufficient and distributable
- Has programing framework of container

Web AppBuilder (Developer Edition)



Widgets

Building Blocks of Apps

Break the code into files

MyWidget.css

```
html, body, #map{
  height: 100%;
  margin: 0;
  padding: 0;
}
```

MyWidget.js

```
define(['dojo/_base/declare', 'jimu/BaseWidget'],
function(declare, BaseWidget){
  var clazz = declare([BaseWidget],{
  });
  return clazz;
});
```

MyWidget.html

```
<div id="feedback">
  <h3>Washington State</h3>
  <div id="info">
    <div id="note">
      Note:  This sample requires an ArcGIS Server version 10.1 map service to generate a renderer.
    </div>
    Select a field to use to create a renderer for the counties in Washington state.
  </div>
  <div id="legendWrapper"></div>
  <div id="fieldWrapper">
    Currently selected attribute:
  </div>
</div>
```

Tutorial:

https://developers.arcgis.com/javascript/jshelp/intro_custom_dijit.html

Inheriting from BaseWidget

```
define(['dojo/_base/declare', 'jimu/BaseWidget'],
function(declare, BaseWidget){
  var clazz = declare([BaseWidget],{
  });
  return clazz;
});
```

A widget derived from the BaseWidget class

Dijit lifecycle

- postCreate
- startup
- …

Widget events

- onOpen, onActive
- onClose, onDeActive

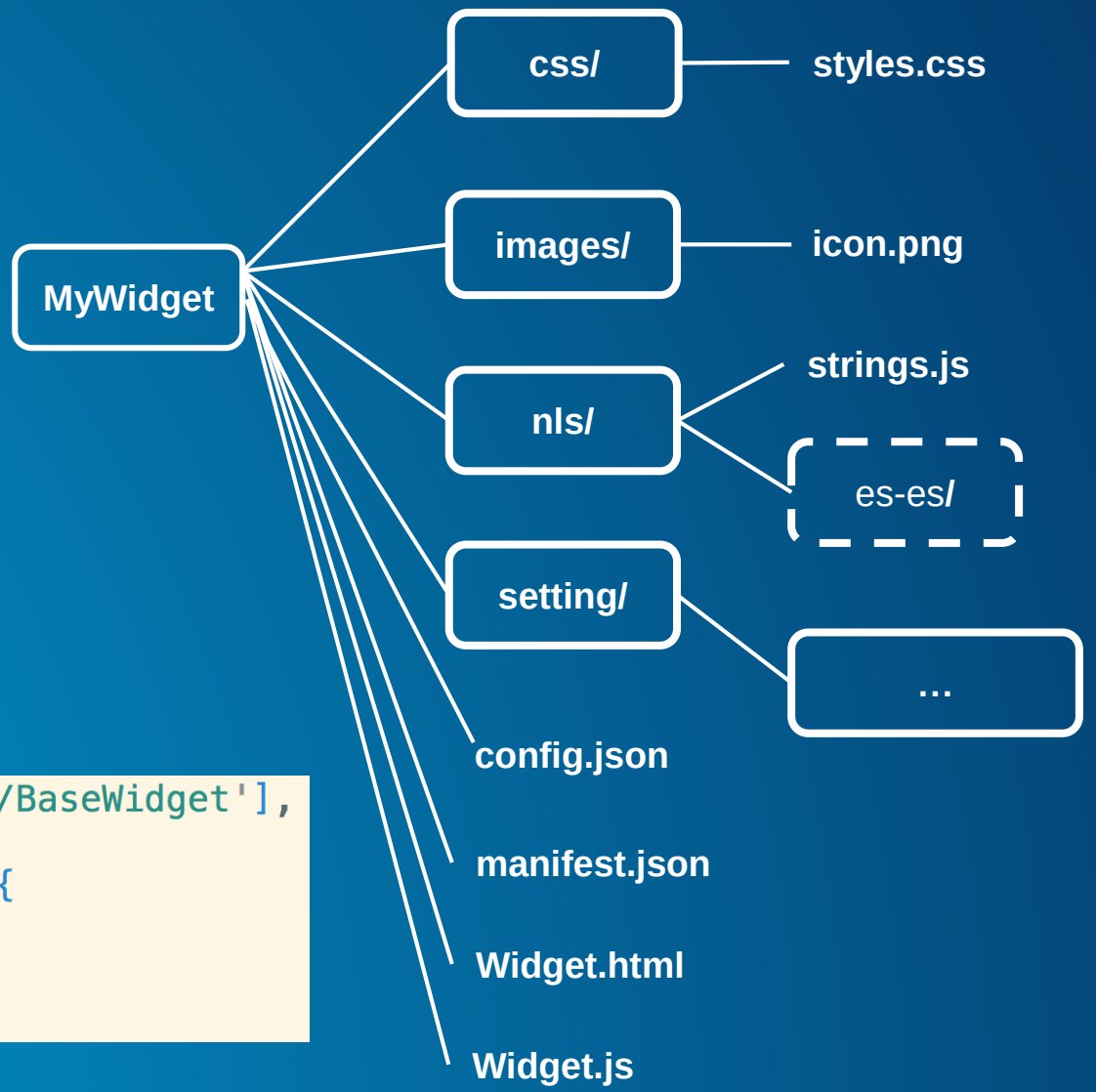
BaseWidget

- App properties (name, icon, localization)
- App config data
- Widget's config data
- Map object
- Widget state (open, closed, active…)
- Events (open/signIn)
- Widget communication

Your job?

- Widget UI (HTML/template)
- Widget config file (JSON)
- Widget styles (CSS)
- Localization
- **Your unique business logic / worklows (JavaScript)**

Widget Files

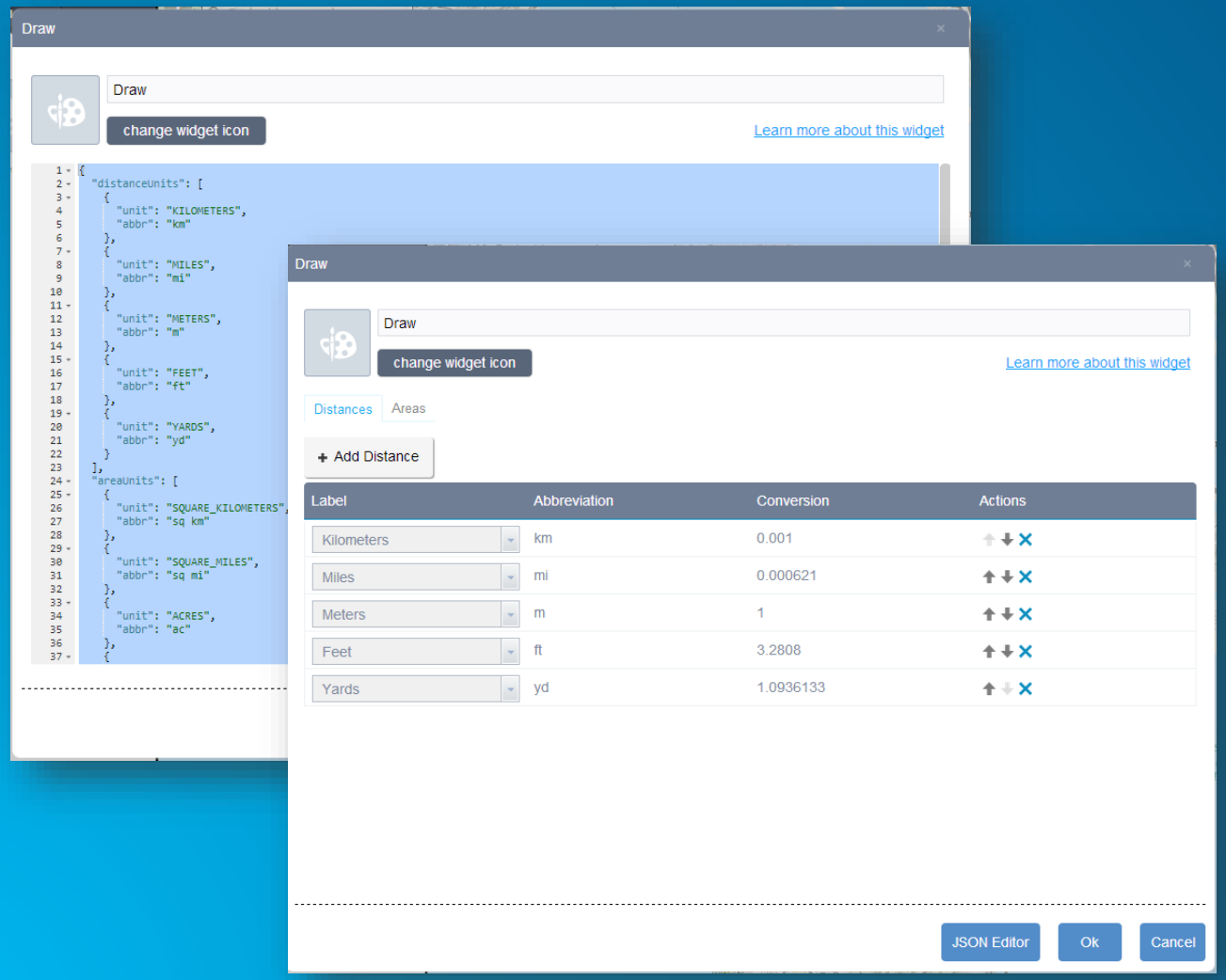



```
define(['dojo/_base/declare', 'jimu/BaseWidget'],
function(declare, BaseWidget){
  var clazz = declare([BaseWidget],{
  });
  return clazz;
});
```

Getting Started…

1. Download developer edition
2. Connect to organization or portal
3. **Copy** widget template
4. **Run** the builder
5. **Create** an app with your widget
6. **Build** your widget in the app

Configure your custom widget inside the builder



Building a UI for the user

- **Setting.js**
  - **Config info**
  - **getConfig, setConfig**
- **Setting.html**
- **Usual localization pattern**
- **css**