



ESRI DEVELOPER SUMMIT 2023

ArcGIS Experience Builder: Customizing and Extending

Gavin Rehkemper

Mark Torrey

Slides: esriurl.com/exb-ce-ds2023

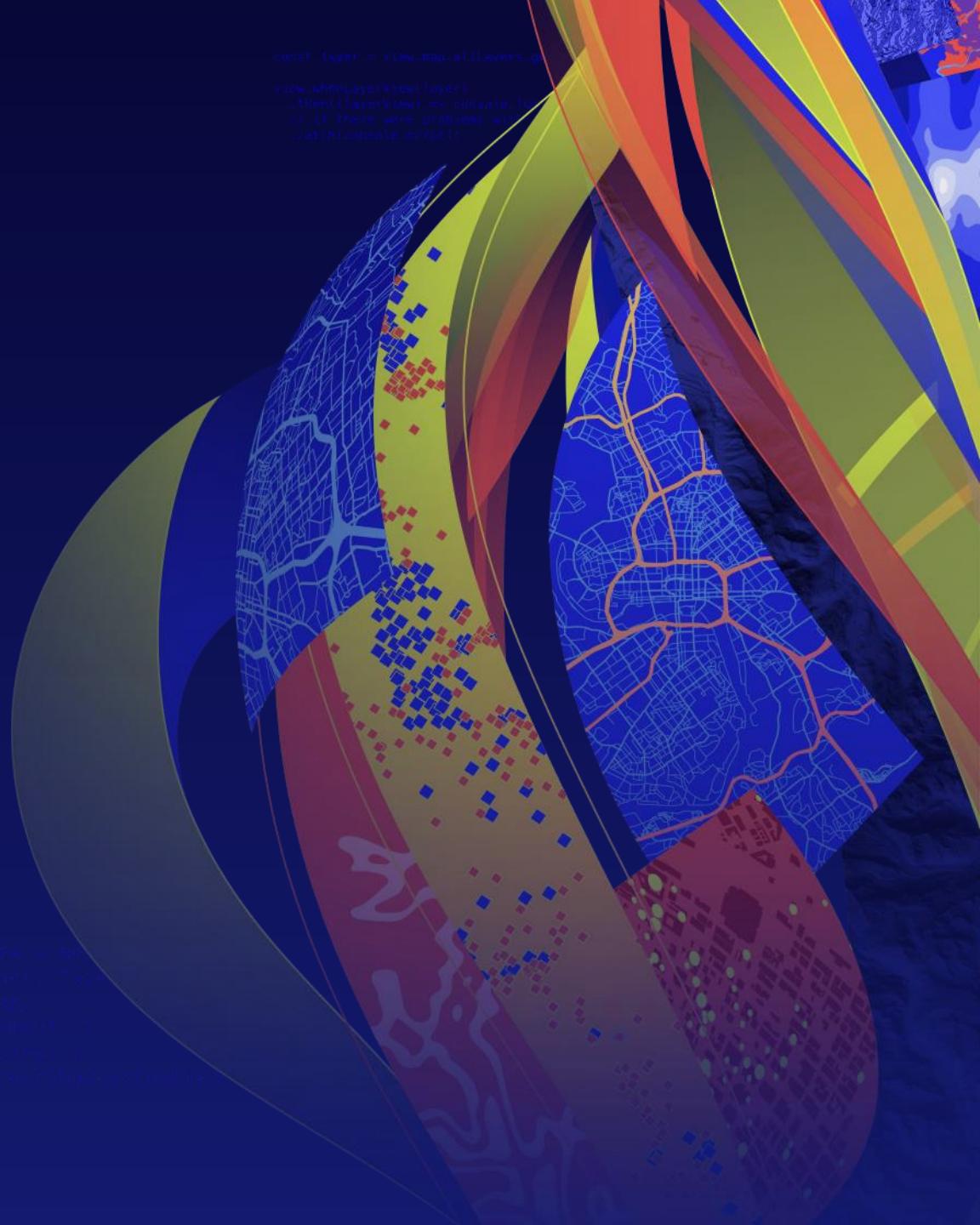
Agenda

- What's New
- Overview
- Custom Widgets
- Custom Themes
- Deployment
- Automated Deployment (CI/CD)

What's New

```
const view = new  
  ol.View({  
  center: [0, 0],  
  zoom: 10,  
  rotation: 0  
});  
  
const layers = [  
  new ol.layer.Tile({  
    source: new ol.source.OSM()  
  }),  
  new ol.layer.Vector({  
    source: new ol.source.Vector({  
      url: 'https://api.mapbox.com/v4/  
        mapbox/tiles/11/{z}/{x}/{y}.json?  
        access_token=pk.eyJ1IjoiZG9tYWluYXVzIiwiYSI6ImNqdzBv  
        ZmJvZTcifQ.1DgkOOGdLWzCwPfHhRzJLg' //<--  
        Mapbox tiles  
    })  
  })  
];
```

```
const layer = view.getLayers().get(1);  
  
view.whenLayerView(layer)  
.then(layerView => console.log(  
  // If there were problems with  
  // catching errors:  
  layerView.getFeatures().catch(error =>  
    console.error(error))  
)
```

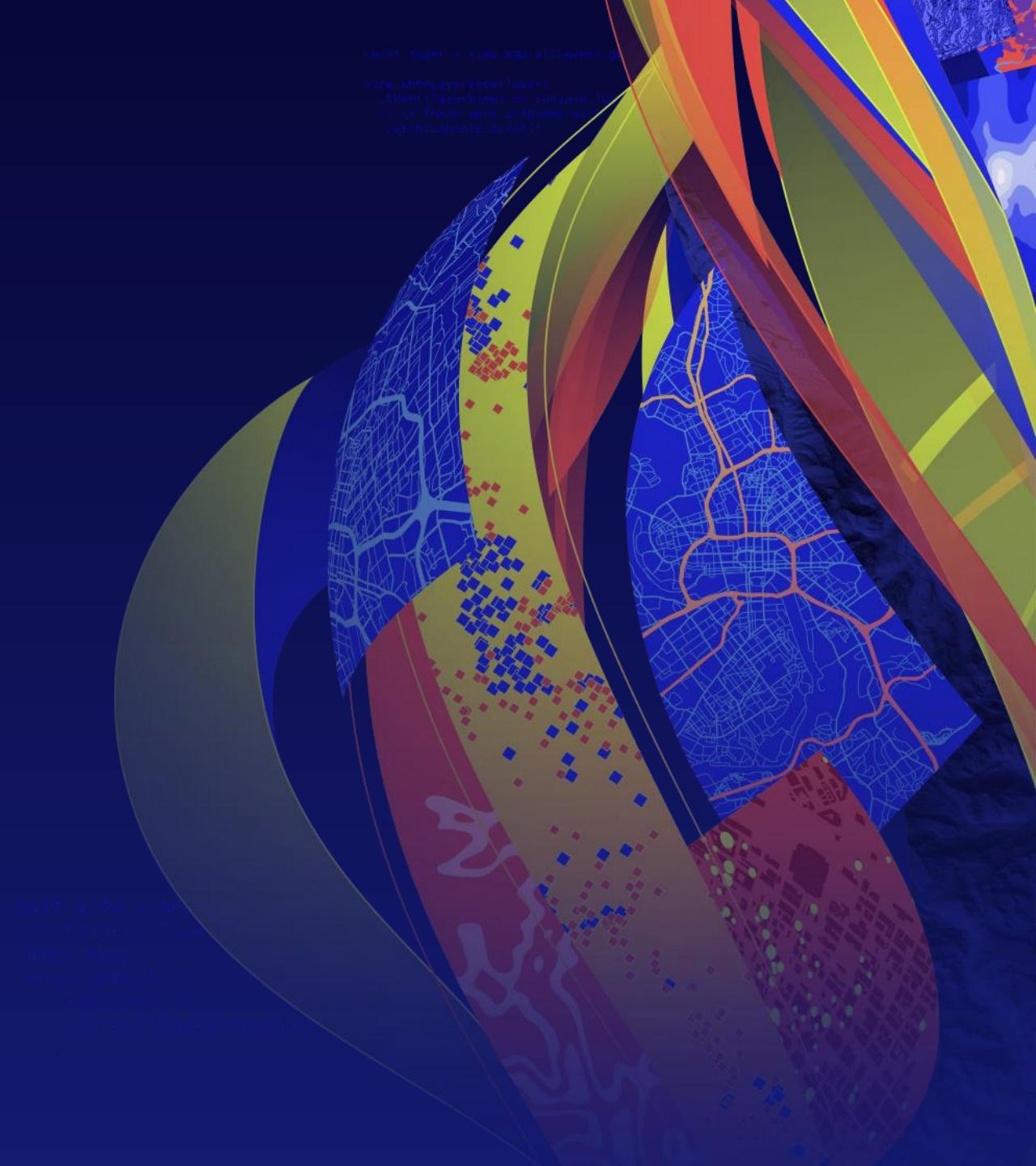


Updates in 1.10

Highlights

- Builder
 - Grid layout
 - Templates
 - Full-Screen: Kit, Chronology, Checkerboard, and Black grid
 - Multipage: Illustrator
 - Two new sample widgets
 - Run developer edition behind a reverse proxy.
- Widgets
 - New: Coordinates, Grid
- Go to the developer guide to see a full list updates
<https://developers.arcgis.com/experience-builder/guide/whats-new/>

Overview

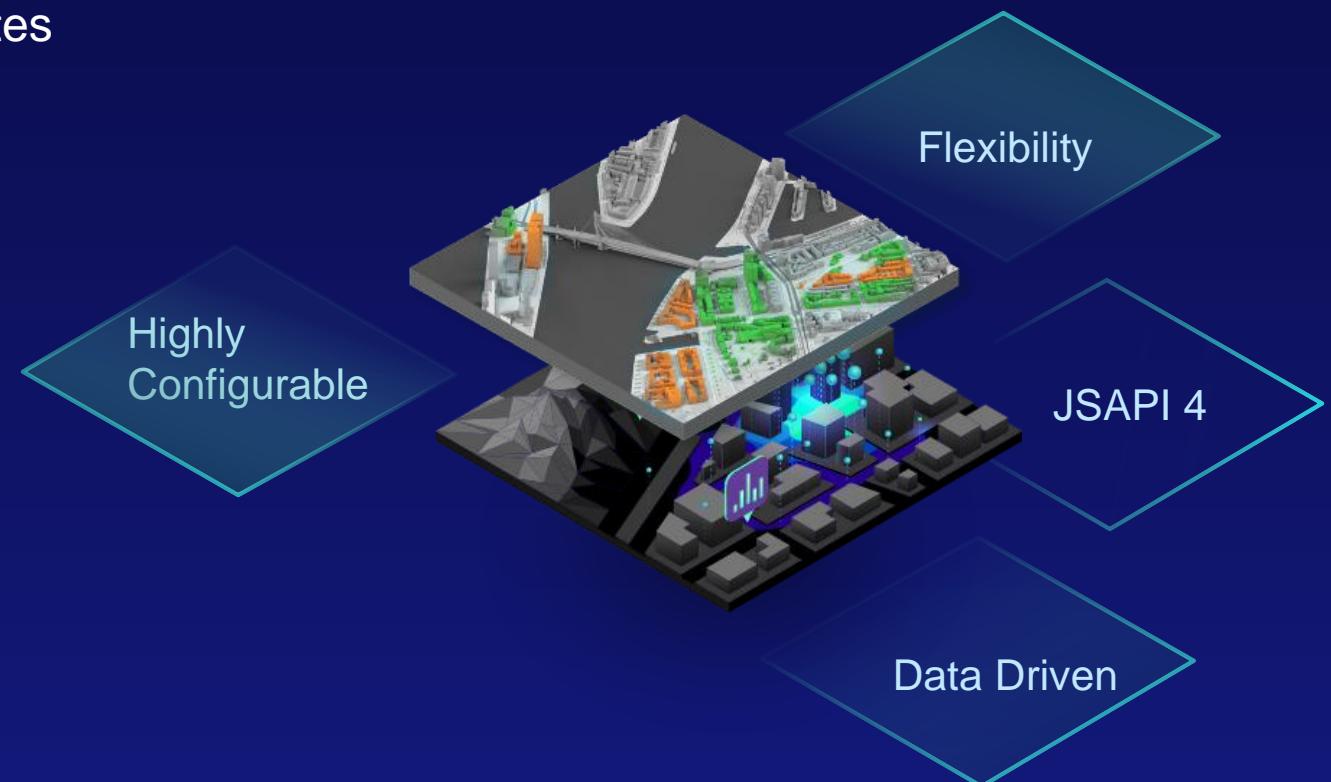


```
const layer = view.map.getLayers().get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with the layer
    // catch(console.error);))
```

```
const view = new MapView();
view.container.id = "map";
view.map.setCenter([10, 40], 10);
view.map.addLayer(new HeatLayer({
  data: [
    {lat: 35, lon: -10, weight: 1000},
    {lat: 35, lon: -10, weight: 1000},
    {lat: 35, lon: -10, weight: 1000},
    {lat: 35, lon: -10, weight: 1000}
  ]
}));
```

What Experience Builder can do

- Build web apps and pages
 - Explore **design** decisions with templates
 - Access ready-to-use **widgets**
 - Build custom tools
- Data driven
- Modern interface
- Better performance



Key Features

- Mobile optimization
- Flexible design
- 2D and 3D in one app
- Communication between widget
- Integration with other ArcGIS apps
- Extensibility





Design/
config

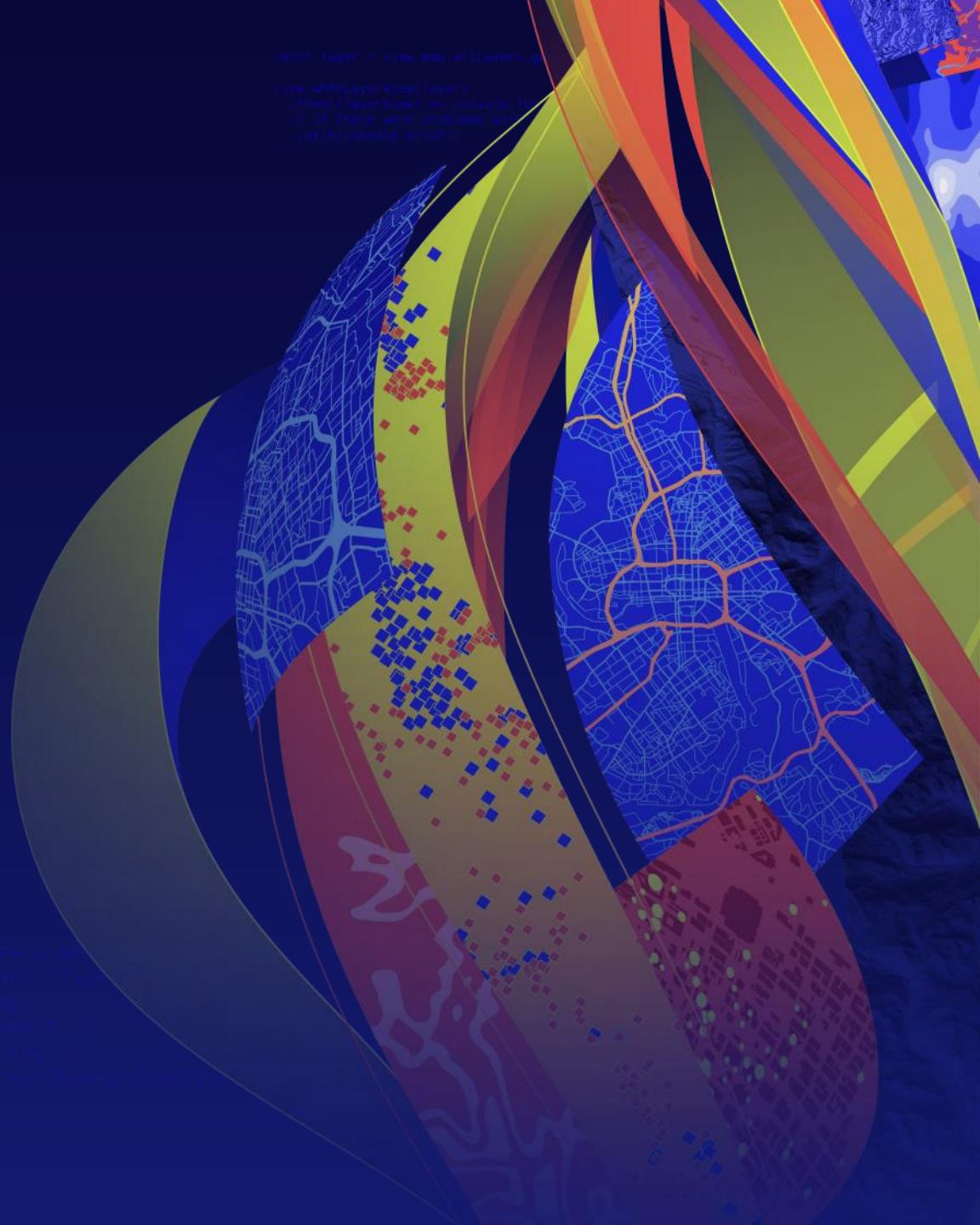


Custom Widgets

Building blocks of an experience

```
const layer = view.map.allLayers.get();
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with
    // catching errors:
```

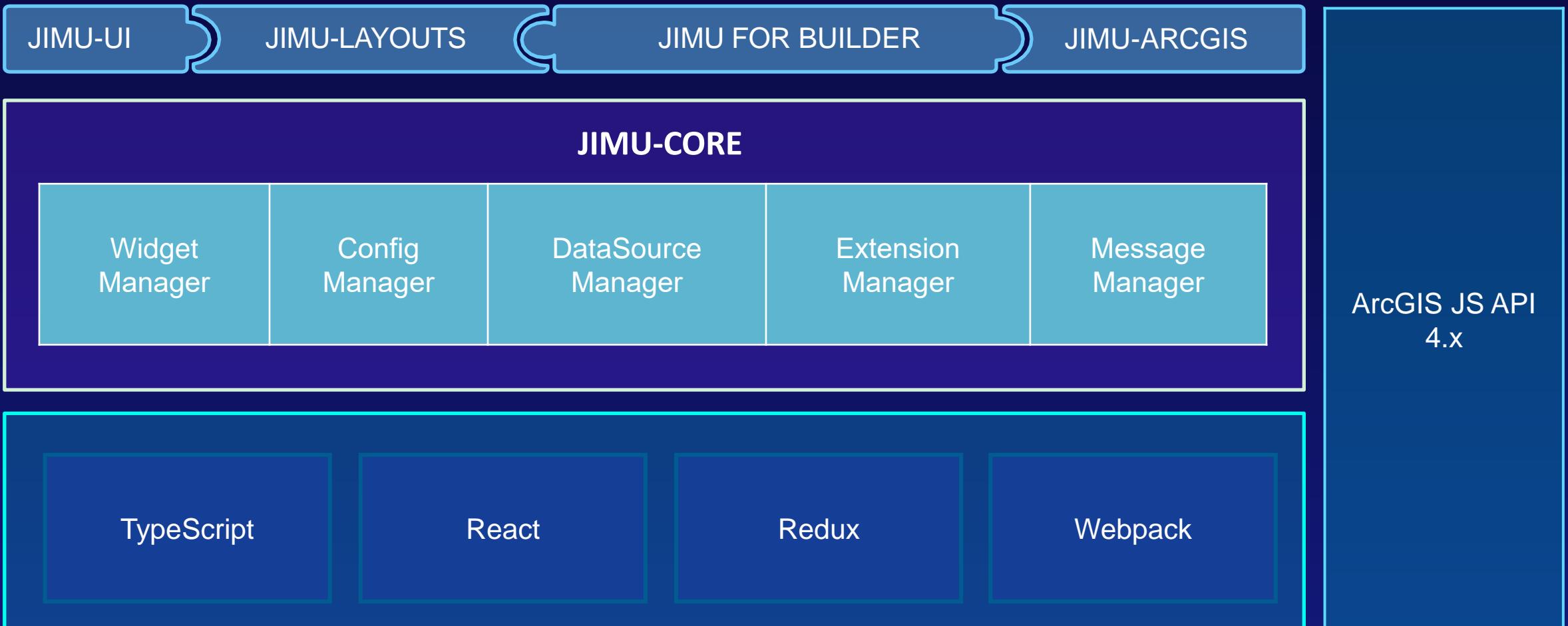
```
const view = new
  ol.View({
    center: [0, 0],
    zoom: 10
  });
const map = new
  ol.Map({
    view: view,
    layers: [
      new ol.layer.Tile(),
      new ol.layer.Vector()
    ]
  });
view.setCenter([0, 0]);
view.setZoom(10);
```



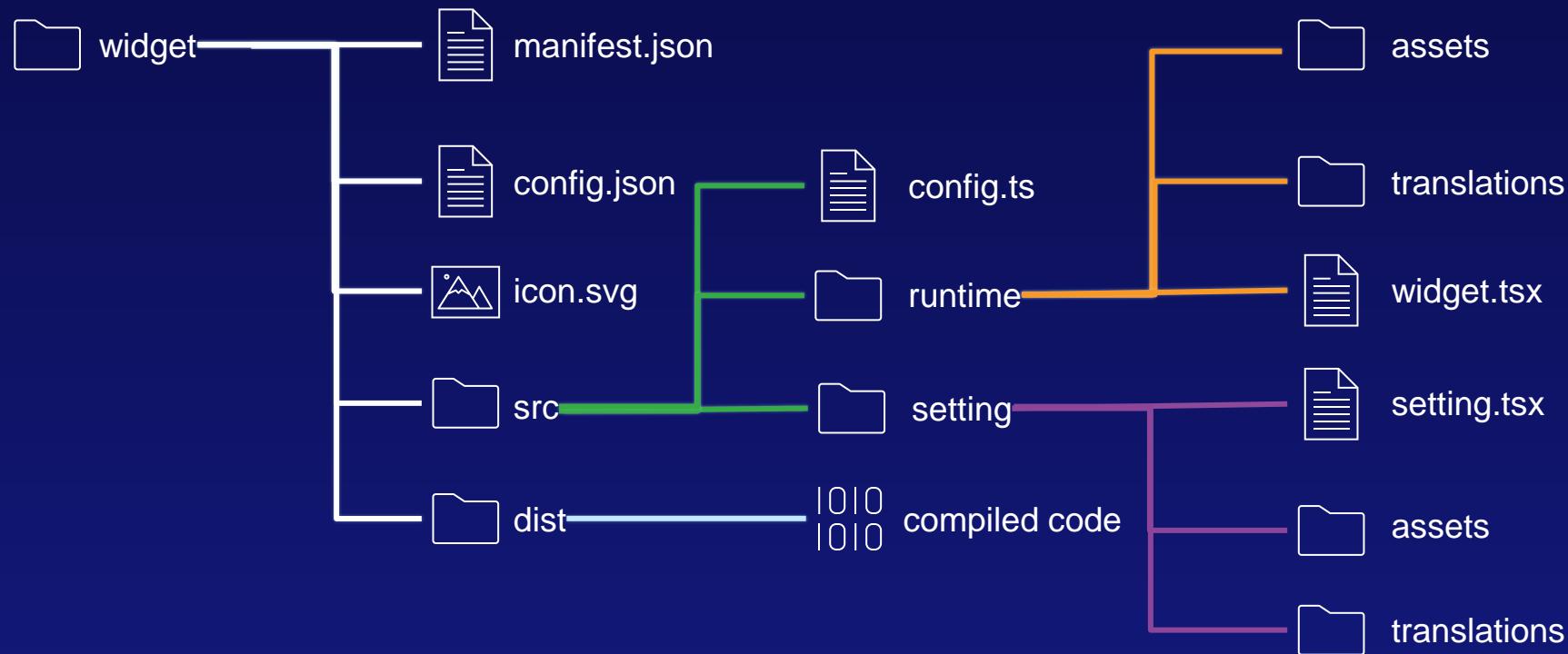
Prerequisites

- React – <https://reactjs.org/tutorial/tutorial.html>
- Typescript – <https://www.typescriptlang.org>
- Jimu - <https://developers.arcgis.com/experience-builder/api-reference/>

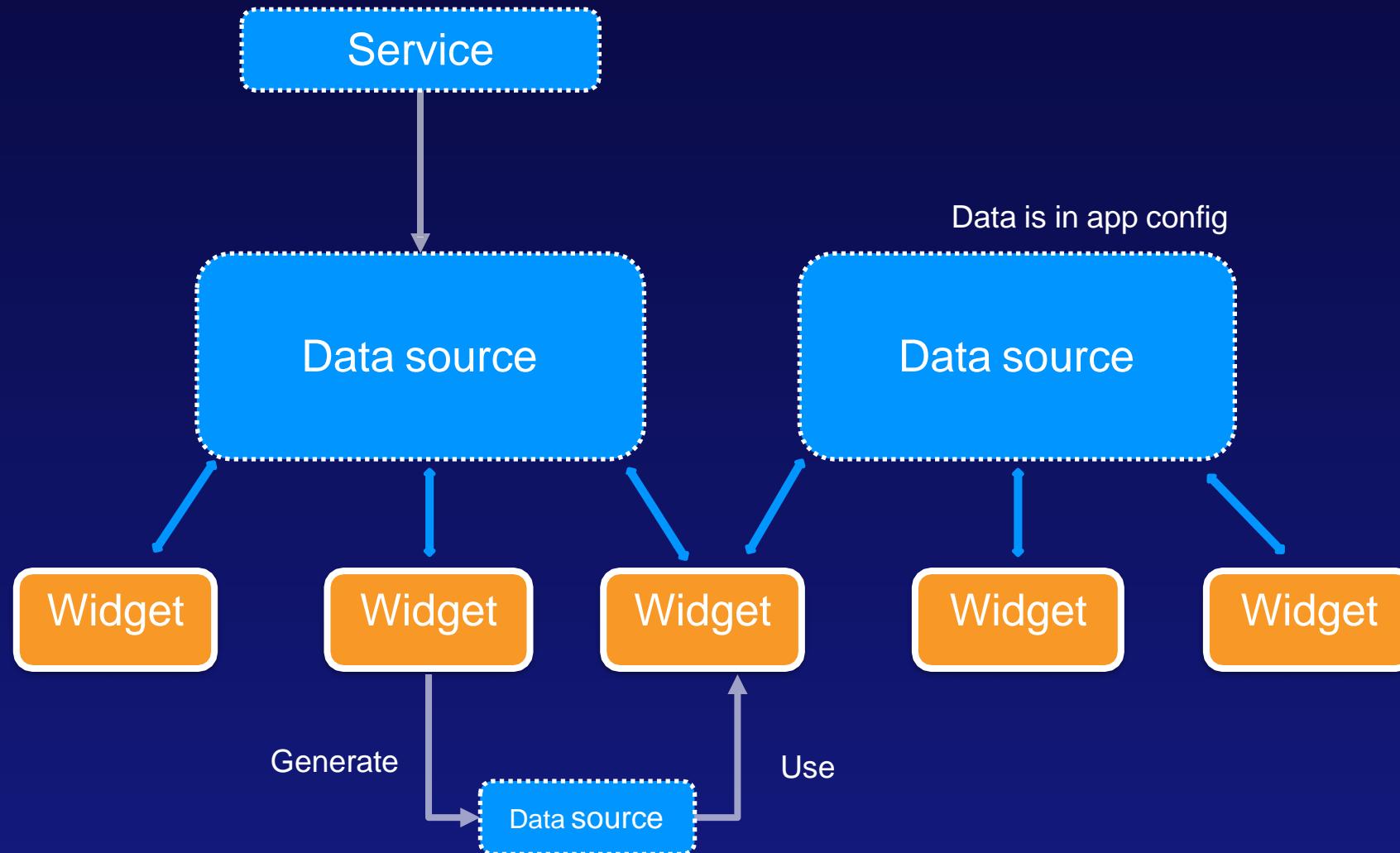
Technology stack



Widget Structure

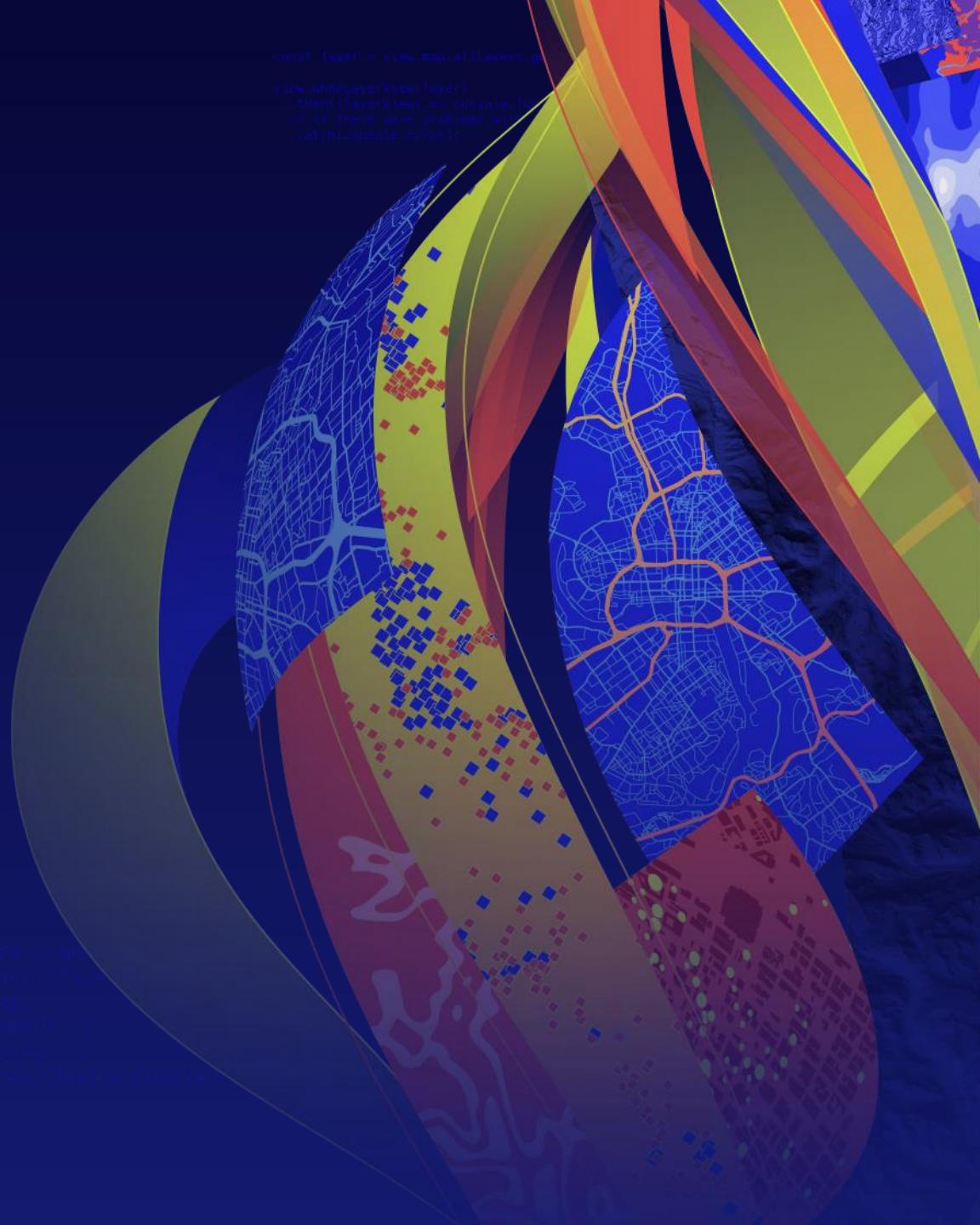


Data sources



Widget Demo

Custom Themes



```
const layer = view.map.getLayers().get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with the layer
    // catch(console.error);))
```

```
const view = new
  ol.MapView({
    layers: [
      new ol.layer.Tile(),
      new ol.layer.Vector()
    ],
    controls: [
      new ol.control.ScaleLine(),
      new ol.control.MousePosition(),
      new ol.control.OverviewMap(),
      new ol.control.Rotate(),
      new ol.control.Zoom()
    ],
    view: new ol.View({
      center: [0, 0],
      zoom: 10
    })
  })
```

What is available

Basic:

- Colors
- Typography
- Spacing

Advanced

- Style override via Sass CSS
- Assets: fonts, images, etc.

Files

- variables.json
- thumbnail.png
- manifest.json

The image shows a complex 3D visualization of a scene graph. It features several translucent, curved surfaces in shades of blue, red, and green. Overlaid on these surfaces are various data layers: a grid of red and blue squares, a map with yellow and orange roads, a white wireframe model of a person's head, and a series of white contour lines. The background is dark, making the colorful layers stand out. In the bottom right corner, there is some white text and code snippets related to scene graph rendering.

Basic Theme

Demo

Override CSS

- style.ts
 - Use the Emotion library for CSS-in-JS
- style.scss
 - More traditional way
 - Imported last

```
const view = new SceneView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
})
```

let {
</STYLE>

```
const layerView = map.allLayers.getItemAt(index);  
layerView.on("layerview-change", (layerView) => {  
  console.log(layerView)  
});  
// If you try to do something with the layerView, you'll get an error here
```

Override CSS

Demo

Custom Fonts

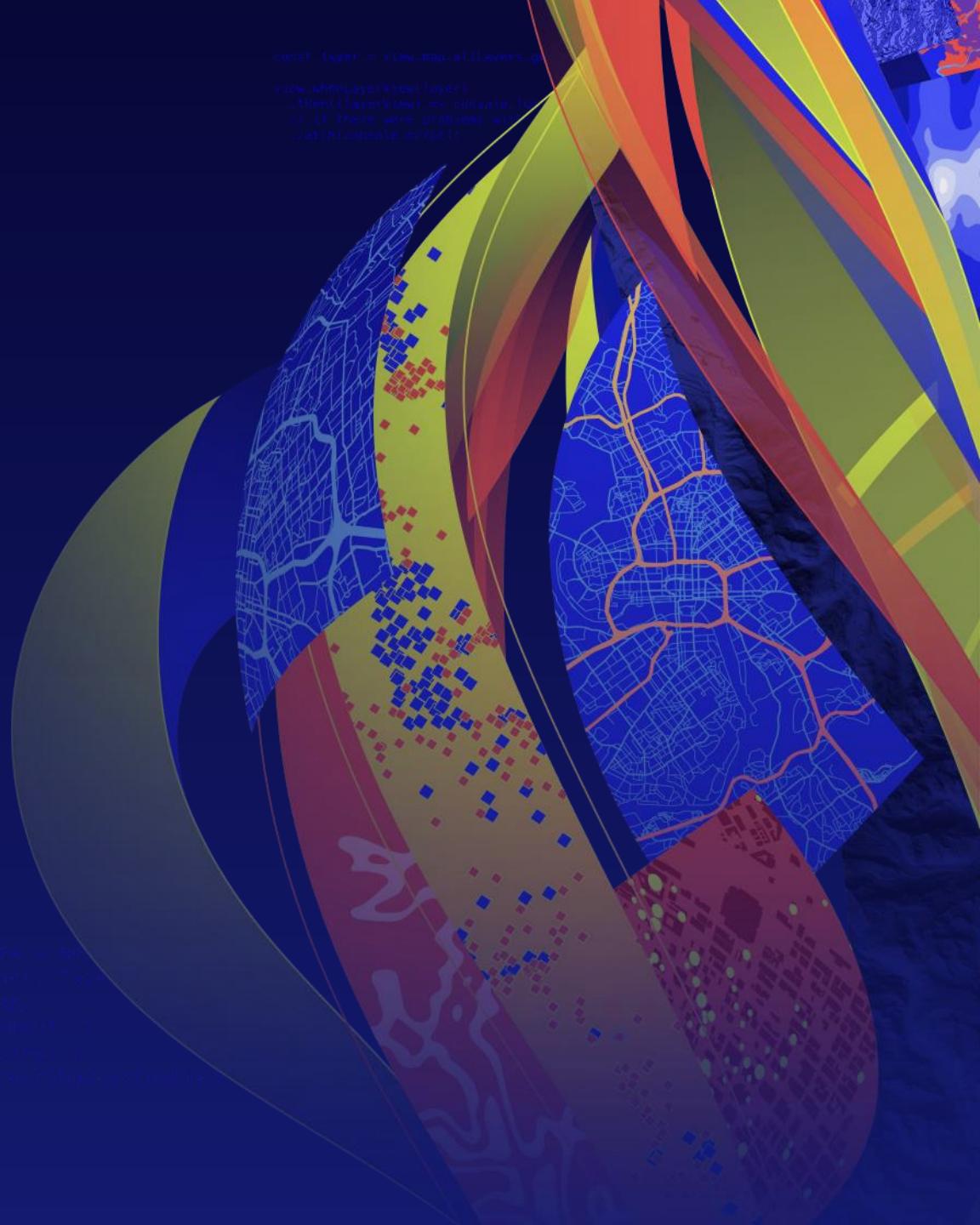
- Use the “Override CSS” method:

```
@import url('https://fonts.googleapis.com/css?family=Open+Sans');
```

A complex collage of various data visualization and programming concepts. It includes a 3D scene view with a yellow sphere containing code, a map with red and blue data points, a bar chart with a red gradient, a pie chart with a grid pattern, a scatter plot with a grid background, and several text snippets related to programming and data science.

Custom Fonts

Deployment



```
const layer = view.map.allLayers.get();
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with
    // catching errors:
```

```
const view = new MapView({
  container: 'map',
  map: map,
  controls: [
    'geolocation',
    'attribution',
    'zoom'
  ]
});
```

```
const layer = view.map.allLayers.get();
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with
    // catching errors:
```

Two Patterns

- ArcGIS Enterprise
- Developer edition > host on a web server

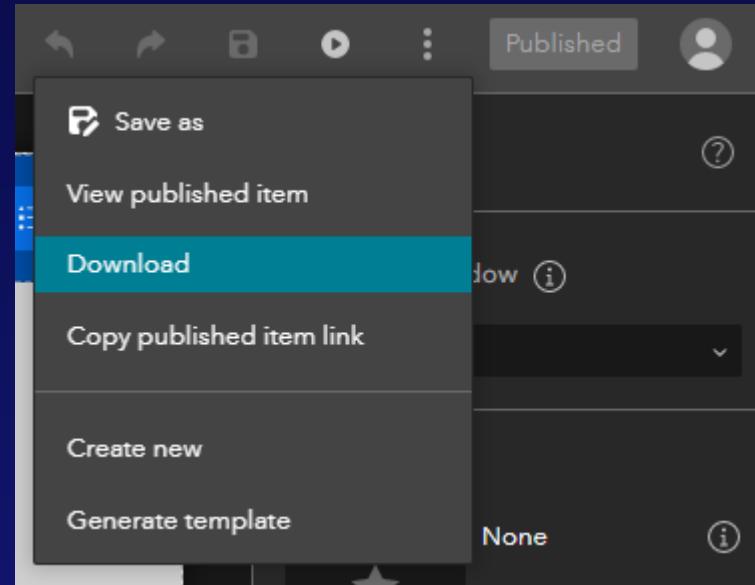
Deployment: ArcGIS Enterprise

- Deploy custom widgets to Experience Builder within ArcGIS Enterprise
- Users within ArcGIS Enterprise can build Experiences with those custom widgets
- ArcGIS Enterprise 11.0 or higher

Detailed instructions: esriurl.com/exb-widget-enterprise

Deployment: Developer Edition

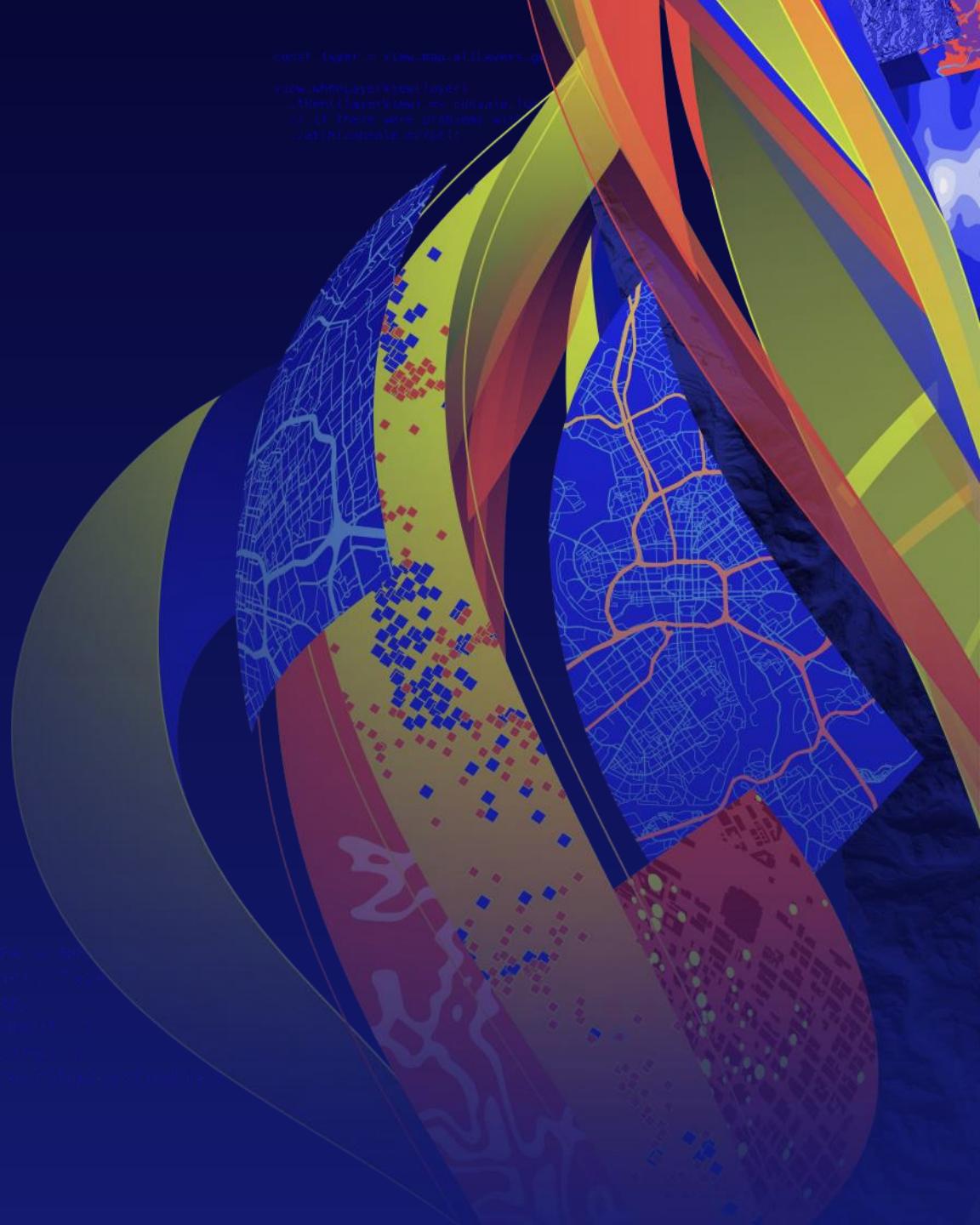
1. Build the Experience with your custom widgets
2. Download as ZIP



3. Unzip files
4. Add `clientId` to `cdn/1/config.json`
5. Host the files on a web server

Automated Deployment

CI/CD



```
const layer = view.map.getLayers().get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(
    // If there were problems with the layer
    // catch(console.error);
```

Why automation?

- Faster builds
- Consistent builds
- Easier for testers
- The "reference" application

How to automate

- Store the reference version of each app
- Auto-build using your CI/CD env of choice:
 - Azure DevOps
 - GitHub Actions*
 - GitLab Actions
 - Jenkins
 - Etc.

Folder structure

- manifest.json
- apps/
 - 0/
 - 1/
 - 2/
 - ...
- widgets/
 - custom-widget-1/
 - custom-widget-2/
 - ...

Automatic build

- Create file:

`github/workflows/build-app.yml`

- Demo repo:

github.com/gavinr/experience-builder-devops-example

The image shows a complex 3D visualization of a scene graph. It features several translucent, overlapping spheres and curved surfaces in shades of yellow, green, blue, and red. A prominent yellow sphere contains white text: 'Inst view = new SceneView({', 'container: "viewDiv",', 'map: map,', 'environment: {', 'lighting: {', 'directShadowsEnabled: true', '}'). To the left of this sphere is a dark blue surface with white text: 'const layerView = map.allLayers.getItemAt(index);', 'layerView.setLayer(theLayer);', '// ...'. Below these are several other graphical elements: a red surface with a grid of blue and red squares; a yellow surface with a blue and red pixelated pattern; and a blue surface with a white map-like grid. The background is a dark blue gradient.

Automatic Build Demo

Automatic build outcomes

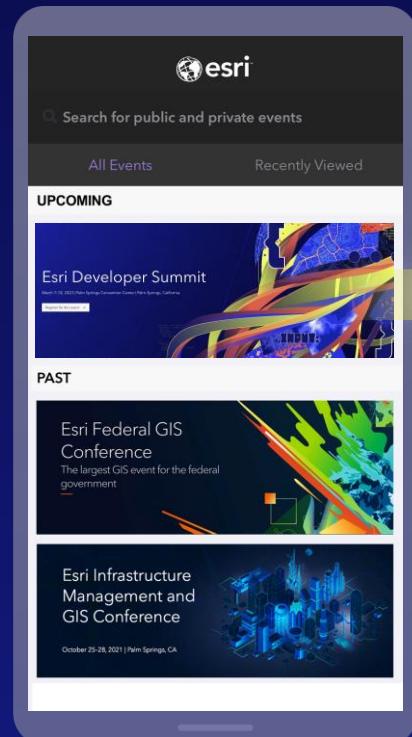
- Consistent outputs
- Easier version management (via Git branching)
- Quicker bug investigations (is it happening on the reference app?)
- More organized management of app config versions

Resources

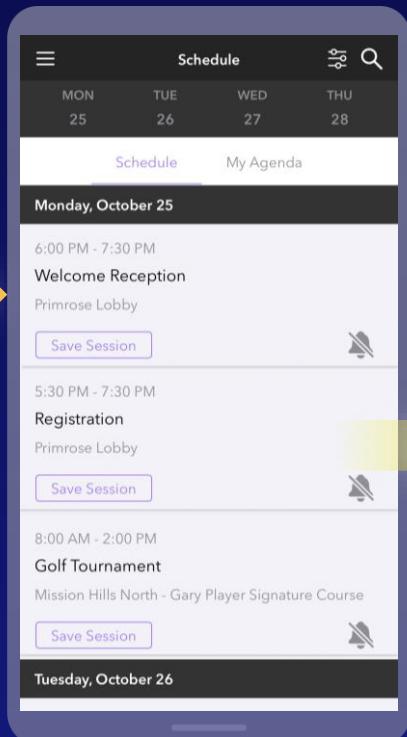
- Code for this presentation: esriurl.com/exb-ce-ds2023
- Samples: github.com/Esri/arcgis-experience-builder-sdk-resources
- Documentation: developers.arcgis.com/experience-builder
 - Guide
 - Sample code
 - API reference
- Questions and feedback in the Esri Community:
community.esri.com/t5/arcgis-experience-builder/ct-p/arcgis-experience-builder

Please Share Your Feedback in the App

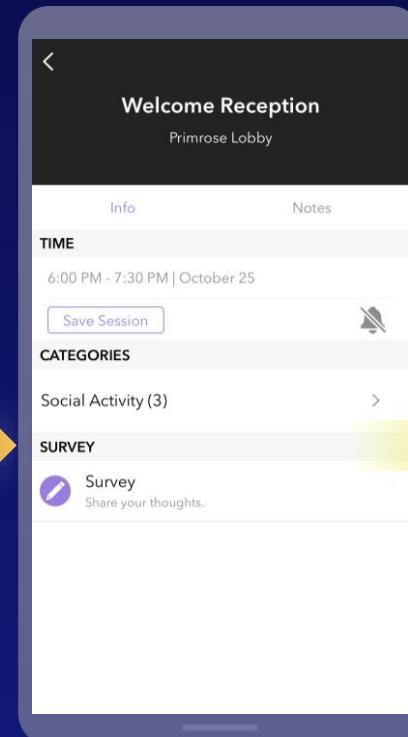
Download the Esri Events app and find your event



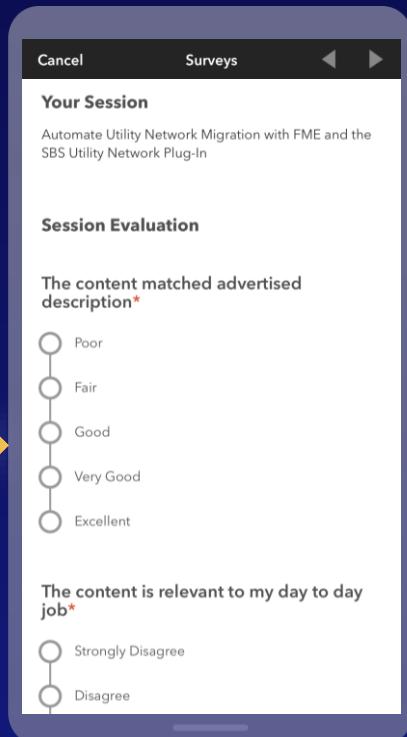
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



Connect With Us On Social

And Join the Conversation Using #DevSummit



twitter.com/EsriDevs

#esridevsummit



twitter.com/EsriDevEvents



youtube.com/c/EsriDevelopers



links.esri.com/DevVideos



github.com/Esri



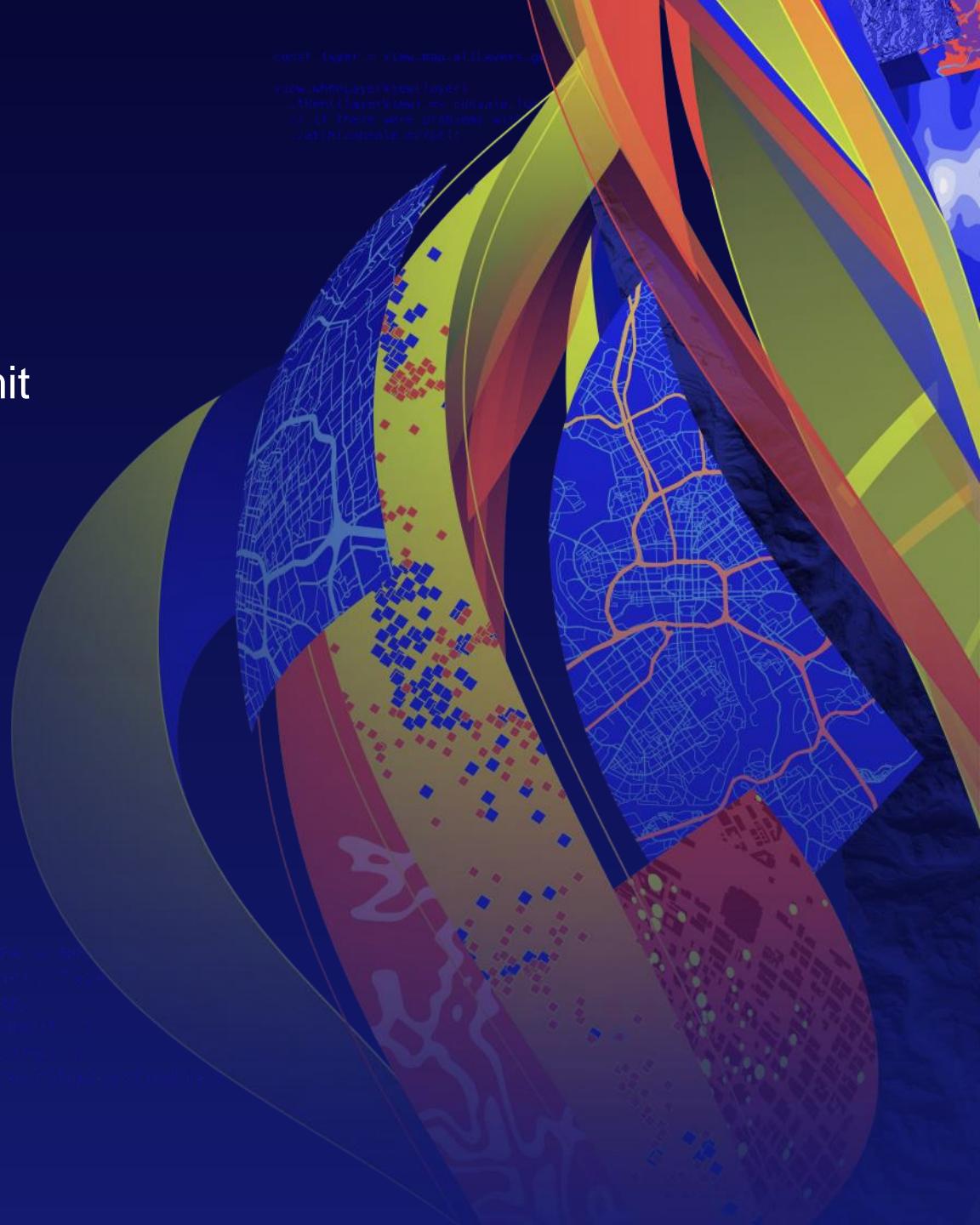
github.com/EsriDevSummit



links.esri.com/EsriDevCommunity

```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => {
    console.log('Layer View');
    // If there were problems with the layer
    // catch them here.
  })
  .catch(error => {
    console.error(error);
  });

```





esri®

THE
SCIENCE
OF
WHERE®

Copyright © 2023 Esri. All rights reserved.

```
const view = new SceneView();
view.container = "viewDiv";
view.map = map;
view.environment = {
    ligthing: true,
    atmosphere: true
};
```

E/SCRIPT>

LIVE
BY
THE
CODE