Esri Developer Summit 2015

The Road Ahead: Web 3D and Native Mobile

with Web AppBuilder for ArcGIS

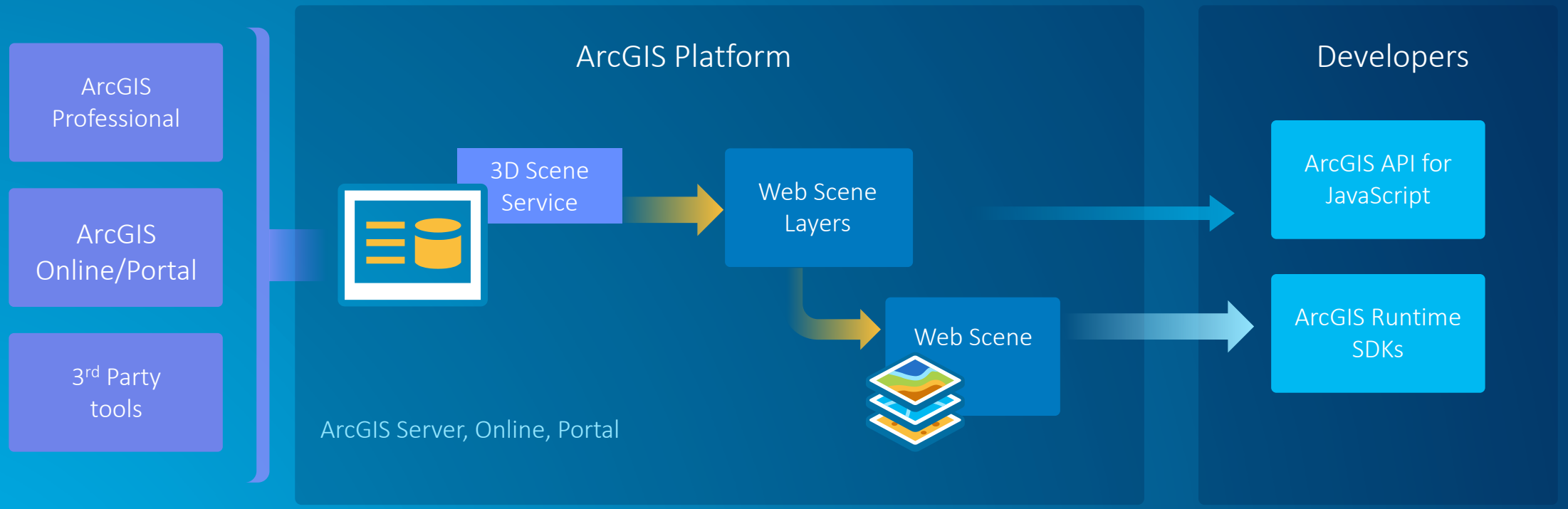
Moxie Zhang, Esri R&D Center Beijing


What is Esri Web 3D
Web 3D App with JavaScript
Web 3D App with Web AppBuilder
3D Visualization with Web AppBuilder
Native Mobile App with Web AppBuilde

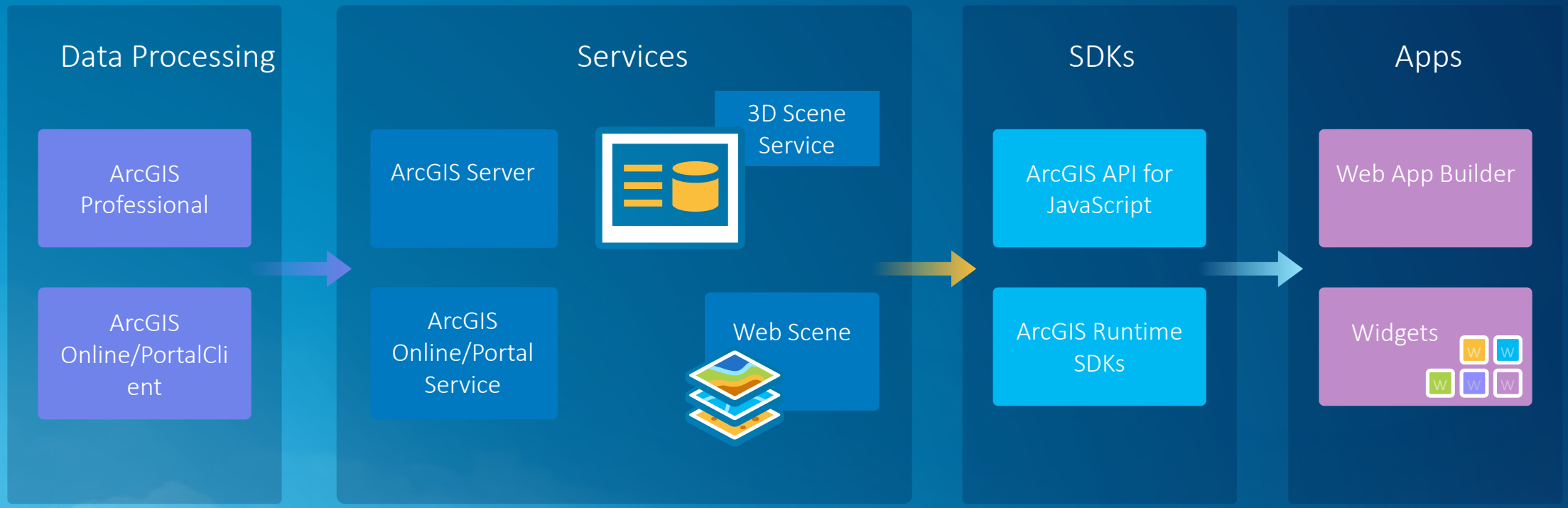
What is Web 3D

for developers

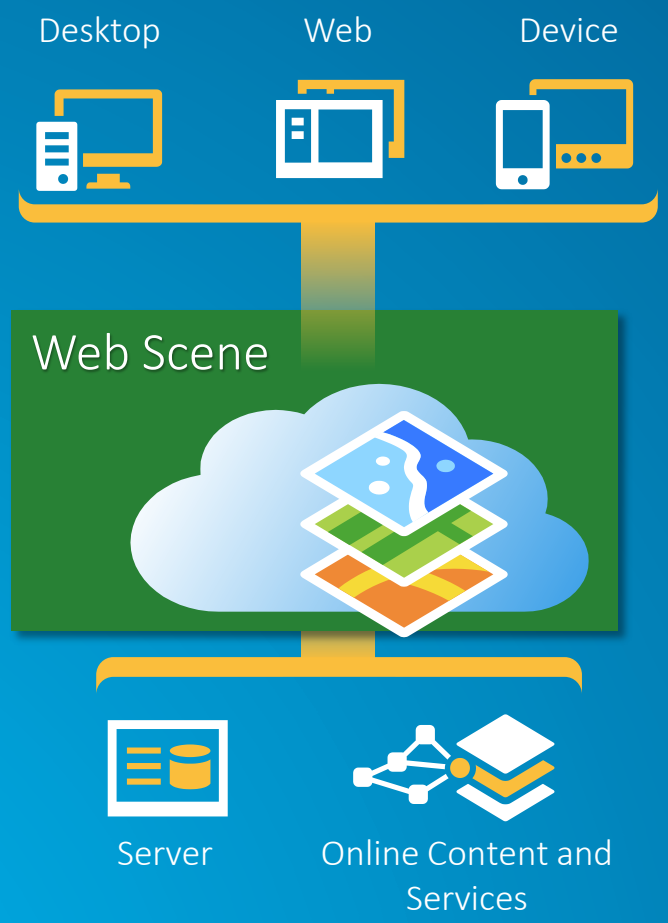
Web 3D Products for ArcGIS



Web 3D Development Architecture (Web AppBuilder)



Web Scene



- New in ArcGIS Online and Portal
- Mash-up of 3D / 2D layers
- Consumed by all clients

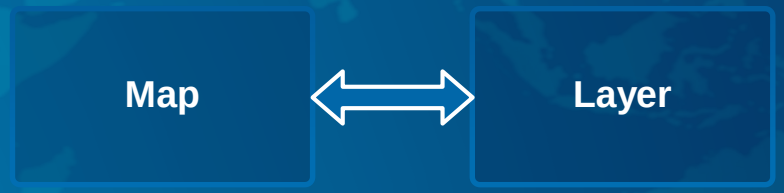
Web Scene – designed for 3D

- 3D Layers
- 3D Symbology
- 3D Labels
- Table of Contents
- 3D Popups
- Tours
- …

General 3D Web Application Architecture

Web 3D Client Architecture

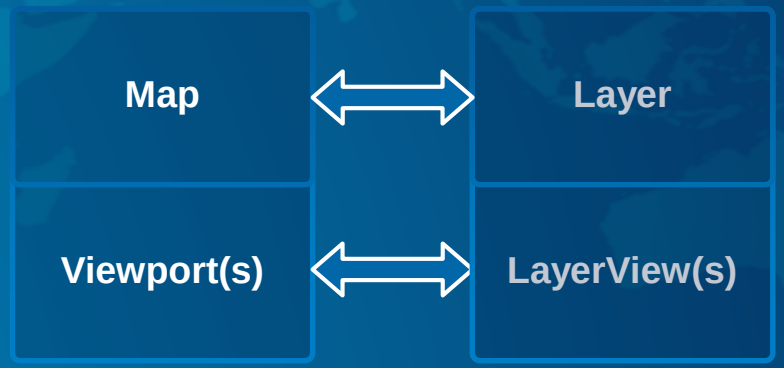
- **ArcGIS API for JavaScript with 3D capabilities**
  - **New internal architecture but same* public JS API classes**



- ** 99% backwards compatible code + new classes for 3D*

Web 3D Client Architecture

- **ArcGIS API for JavaScript with 3D capabilities**
  - **New internal architecture but same* public JS API classes**



- ** 99% backwards compatible code + new classes for 3D*

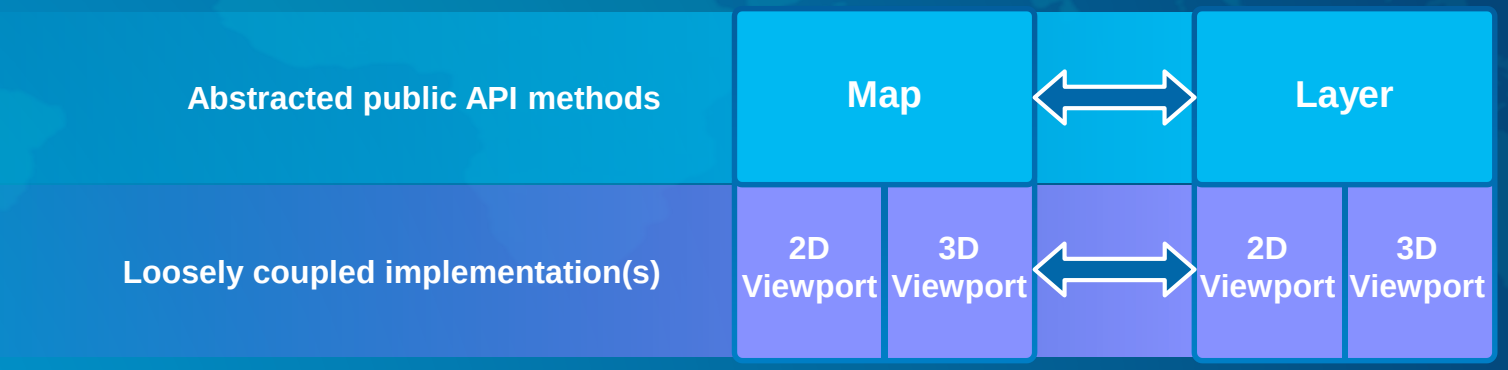
Web 3D Client Architecture

- **ArcGIS API for JavaScript with 3D capabilities**
  - **New internal architecture but same* public JS API classes**

- ** 99% backwards compatible code + new classes for 3D*

Web 3D Client Architecture

- **ArcGIS API for JavaScript with 3D capabilities**
  - **New internal architecture but same* public JS API classes**



- ** 99% backwards compatible code + new classes for 3D*

App Development in 3D

3D Runtime SDK

- Full 3D Runtime functions
- High performance
- Large data set
- Machine Native code
- Mobile and desktop

JavaScript (WebGL Render)

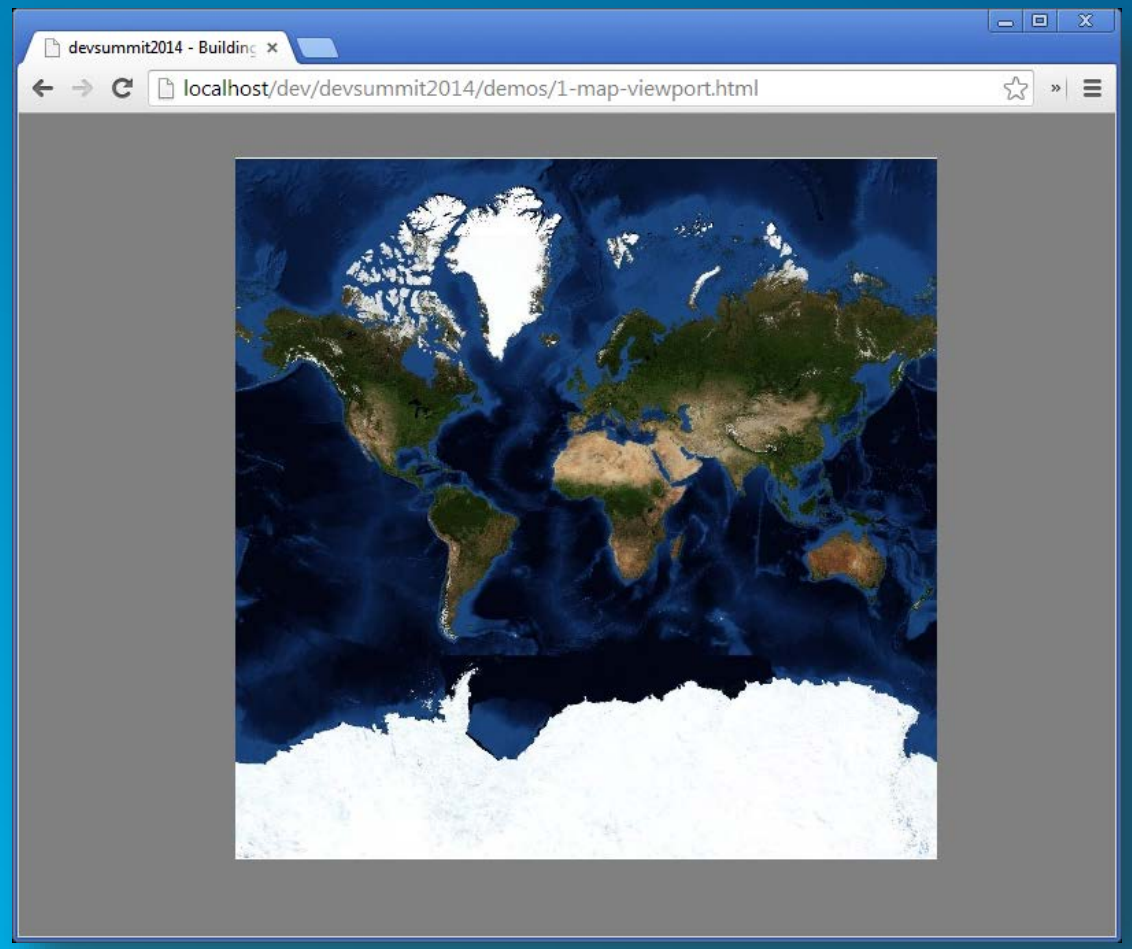
- Pure browser app without plugin
- WebGL is maturing
- Under heavy development
- Performance and large data set support are improving
- Cross browser support

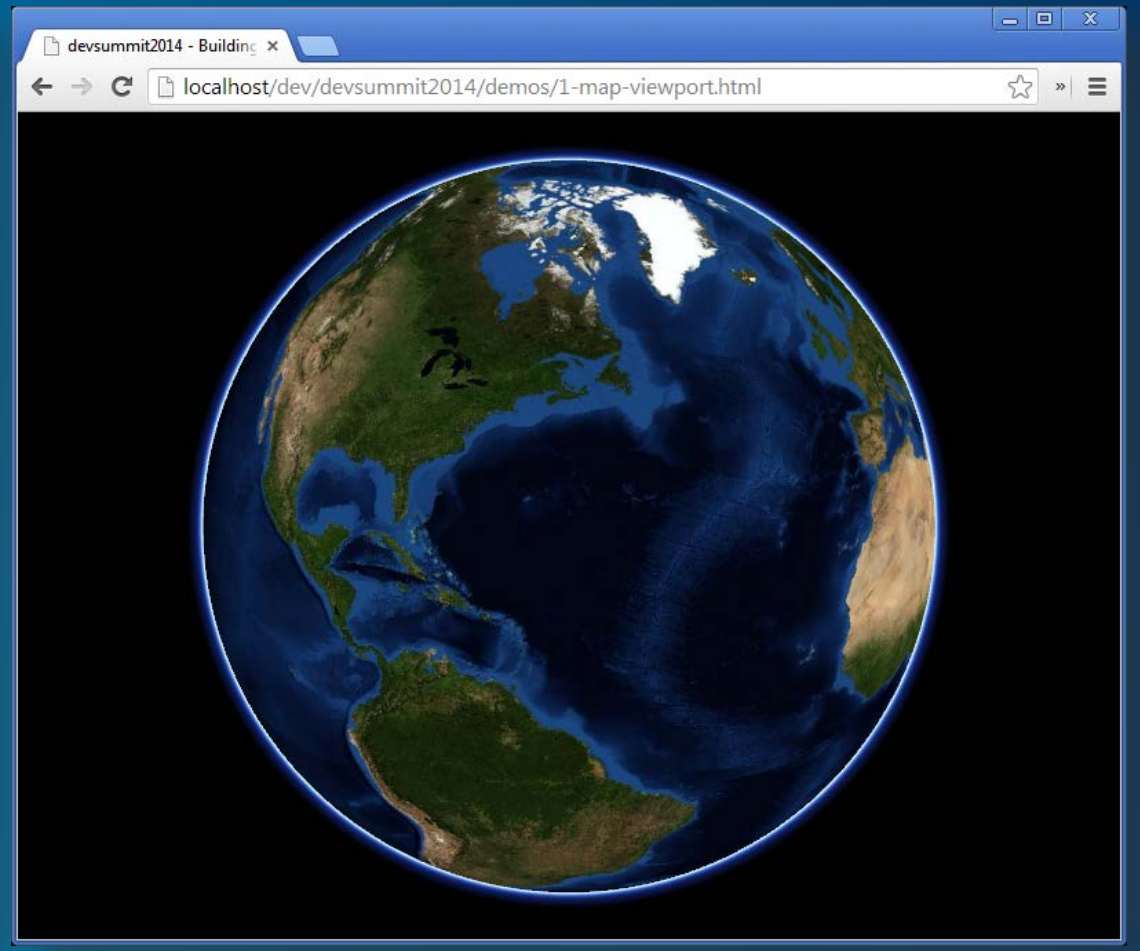
Application Development (1) Full app on API (2) Widget/Theme for Web AppBuilder for ArcGIS

Build 3D Web App

with JavaScript API

Map and Viewport





Create a Map

```
require([
  "esri/Map"
],function(
  Map) {

  // setup the map
  // when creating a map, the option "viewportType"
  // indicates which kind of map to create;
  // the possible values are:
  // "2d" - 2D map
  // "3d-canvas" - WebGL based 3D map
  // "3d-plugin" - Runtime core based 3D map

  var map = new Map("mapDiv", {
    viewportType: "3d-canvas"
  });
});
```

Add a Basemap

```
require([
  "esri/Map",
  "esri/layers/ArcGISTiledMapServiceLayer"
],function(
  Map, ArcGISTiledMapServiceLayer) {

  // setup the map
  var map = new Map("mapDiv", {
    viewportType: "3d-canvas"
  });

  // create tiled layer
  var layer = new ArcGISTiledMapServiceLayer("http:/

  // add layer to map
  map.addLayer(layer);
});
```

Build Web 3D App

with Web AppBuilder


A menu of tools
Interactive content
Shortcut items
Map, of cause
The “player”
ArcGIS Web App
Log In
Query
CLEAR RESULTS
Stem App
CANADA
UNITED STATES
MEXICO
BRAZIL
Atlantic Ocean
Pacific Ocean

Web App Builder for ArcGIS Components

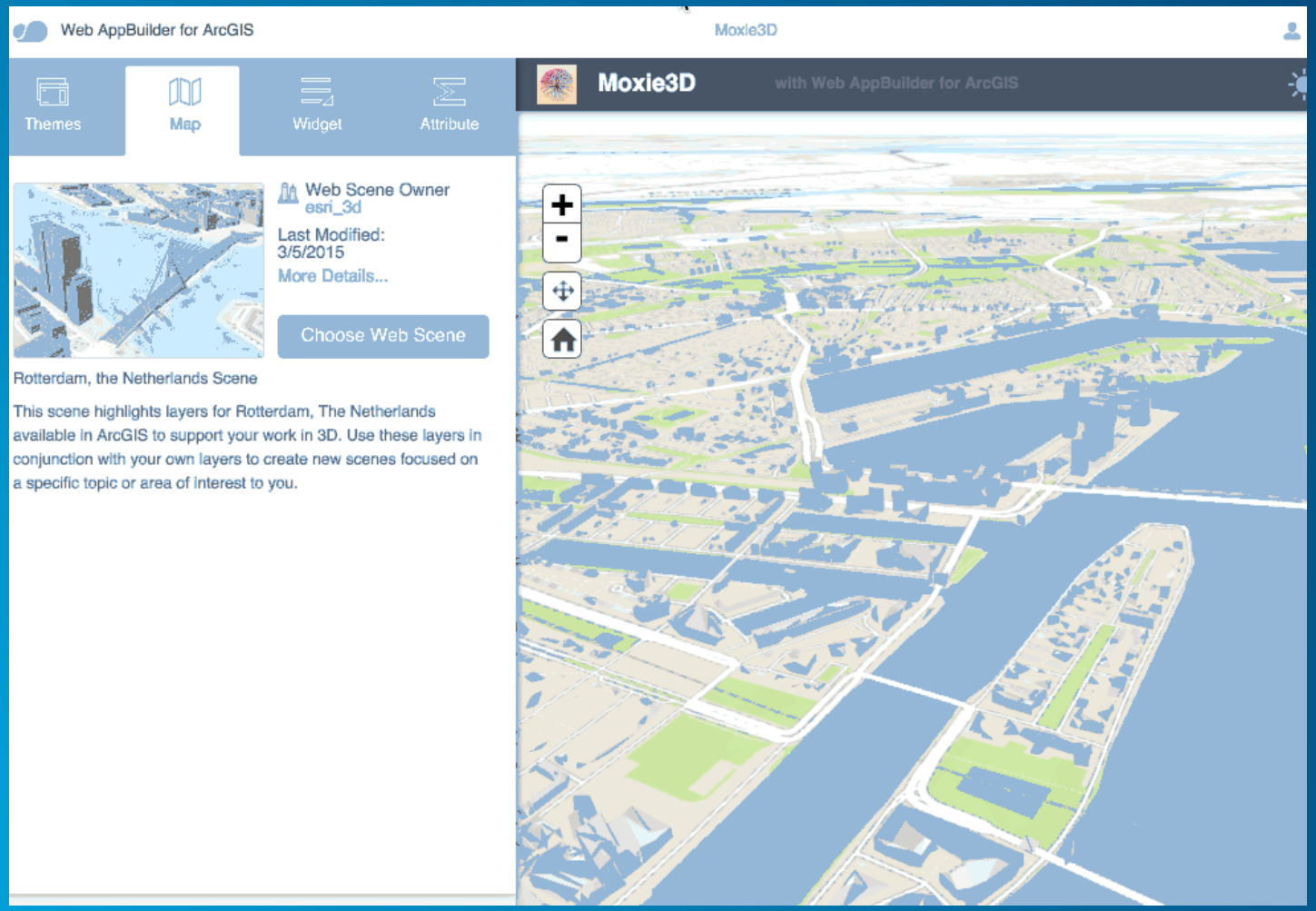
Widget

- HTML/JavaScript/CSS
- Specific task
- Configuration in JSON
- NLS support
- Builder config UI

Theme

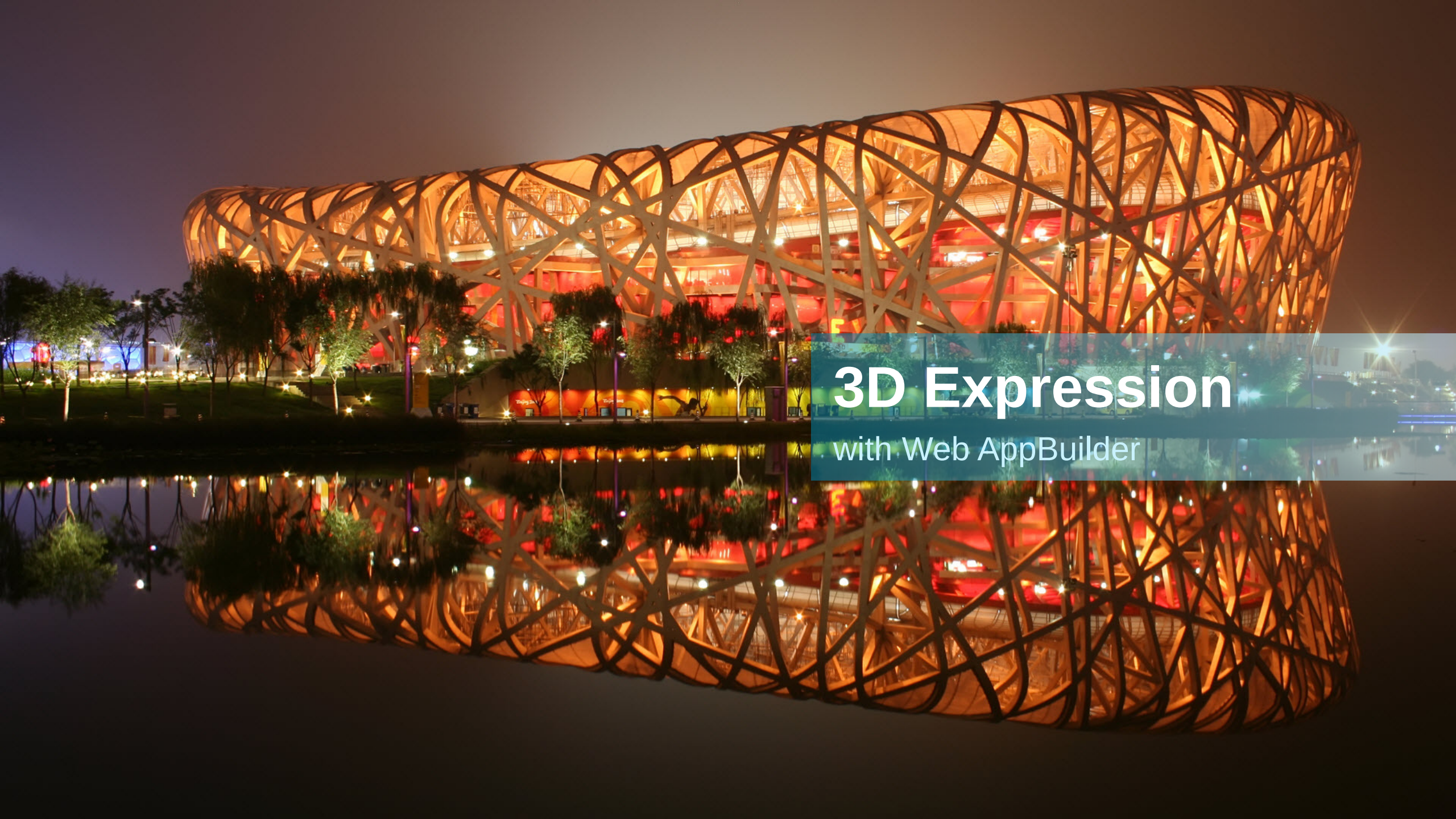
- HTML/JavaScript/CSS
- Layout
- Branding
- Widget panel and behavior
- Style (Color, etc.)
- Default widgets

Demo: Configure a Web 3D App



Code: Create a widget for 3D

```
16
17  define([
18      'dojo/_base/declare',
19      'jimu/BaseWidget',
20      'dojo/query',
21      'dojo/_base/html',
22      'dojo/_base/array',
23      'dojo/_base/lang',
24      'esri/Viewpoint',
25      'esri/viewports/viewport3d/canvas3d/webgl-engine/lib/gl-matrix'
26    ], function(declare, BaseWidget, query, html, array, lang, Viewpoin
27      var clazz = declare([BaseWidget], {
28
29        name: 'Slide',
30        baseClass: 'jimu-widget-slide',
31
32        postCreate: function(){
33          this.inherited(arguments);
34          this._createSlides();
35        },
36
37        _createSlides: function(){
38          var slides = this.map.itemData.presentation && this.map.itemD
39          if(slides && slides.length > 0){
40            array.forEach(slides, lang.hitch(this, function(slide){
41              var str = '<li class="slide">' +
```


3D Expression
with Web AppBuilder

What’s 3D Expression

A set of dynamic effects that visualize data in 3D space

Integrate with Web Scene

Integrate with Web 3D symbolization

A way to better visualize and understand data.

Demo: 3D Visualization

with 3D Expression

Why 3D Visualization

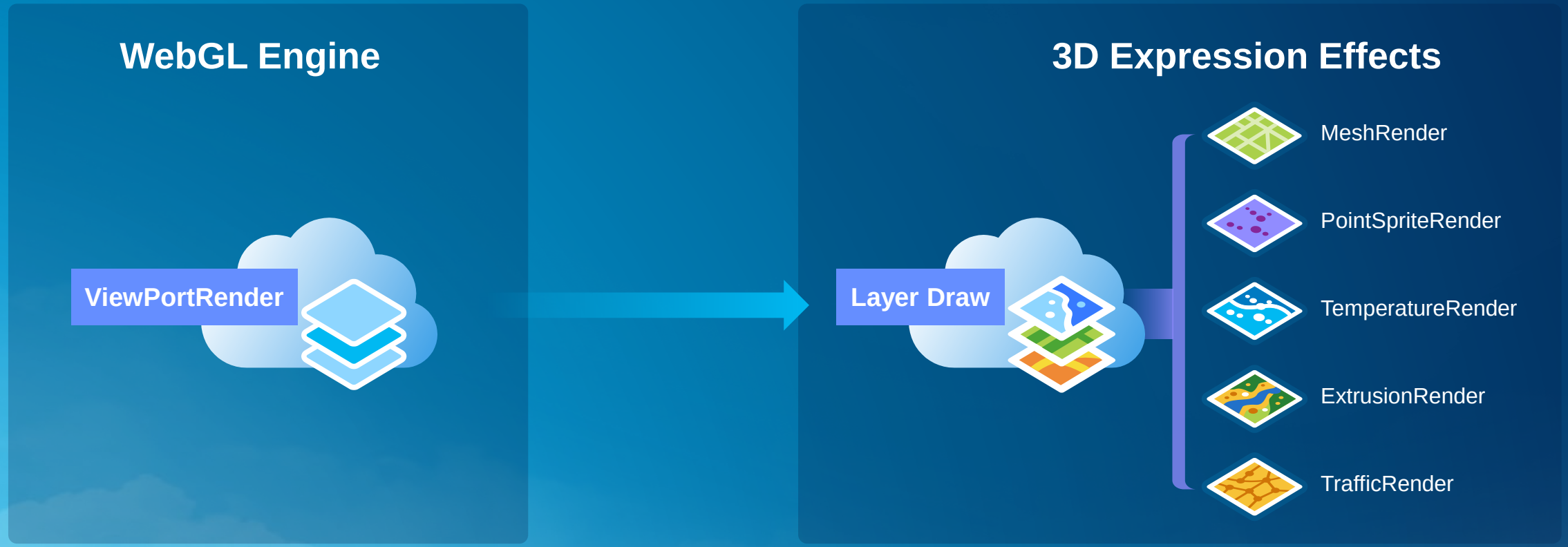
3D adds one additional dimension to data visualization.

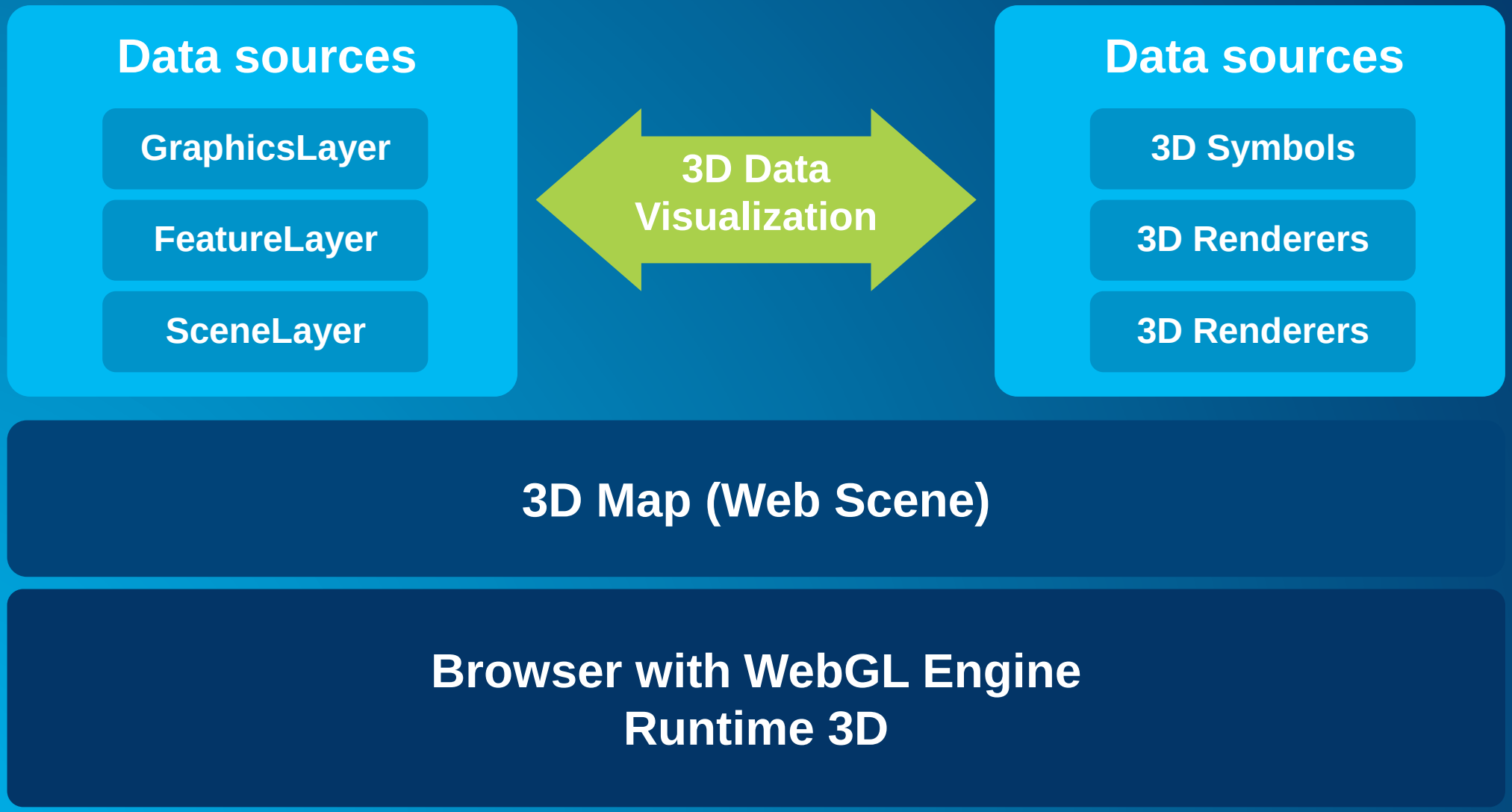
People tend to learn and understand from visual representations much easily than from textual representations.

3D visualization is more in-depth and more intuitional than 2D visualization.

3D visualization can better handle time-relevant data by supporting animated effects.

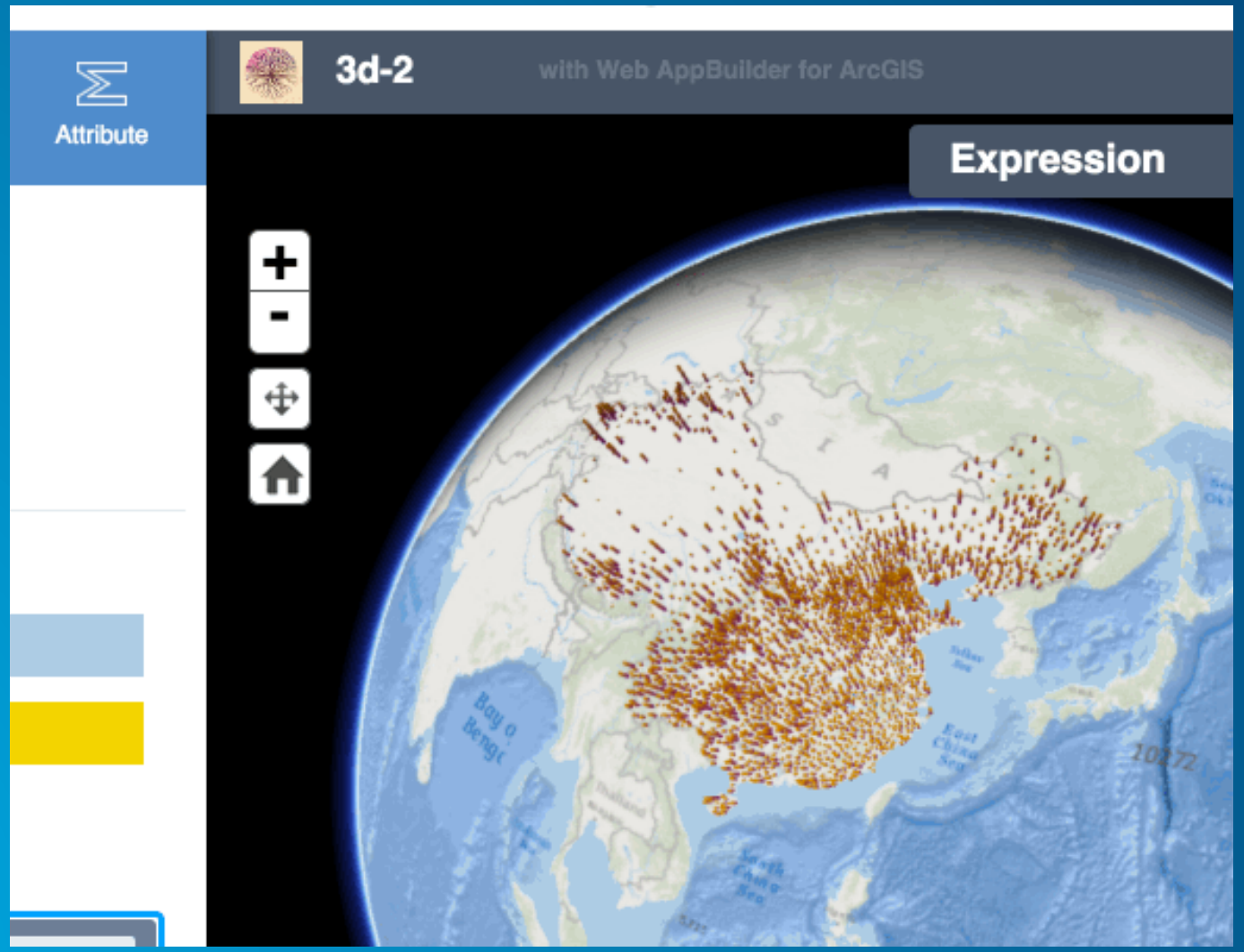
3D Expression Architecture




Data sources
GraphicsLayer
FeatureLayer
SceneLayer
3D Data Visualization
Data sources
3D Symbols
3D Renderers
3D Renderers
3D Map (Web Scene)
Browser with WebGL Engine
Runtime 3D


```
// featurelayer contains data of polygons
var featurelayer = ...
var defaultSymbol = new PolygonExtrusionSymbol({
            "height": ...,
            "color": ...
});

var renderer = new ClassBreaksRenderer(defaultSymbol, "value")

renderer.addClass(100, 200, new PolygonExtrusionSymbol({
    height: ...,
    color: ...
}));

renderer.addClass(201, 300, new PolygonExtrusionSymbol({
    height: ...,
    color: ...
}));

renderer.addClass(301, 400, new PolygonExtrusionSymbol({
    height: ...,
    color: ...
}));

map.addLayer(layer);
layer.setRenderer(renderer);
```

Demo: 3D Visualization with Web AppBuilder

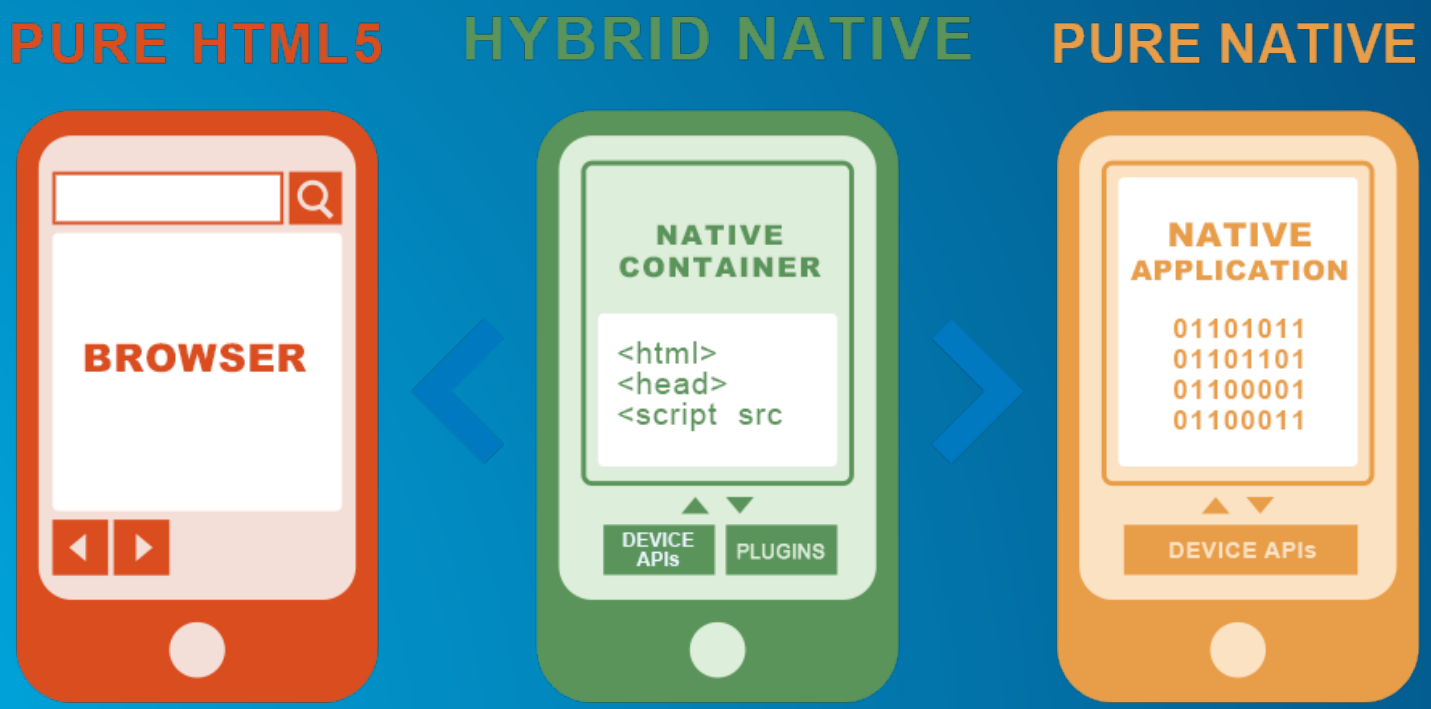


Create Native Mobile App

with Web AppBuilder

Mobile app development starts with many options

…and there are different approaches to build an app



Beyond Native Apps

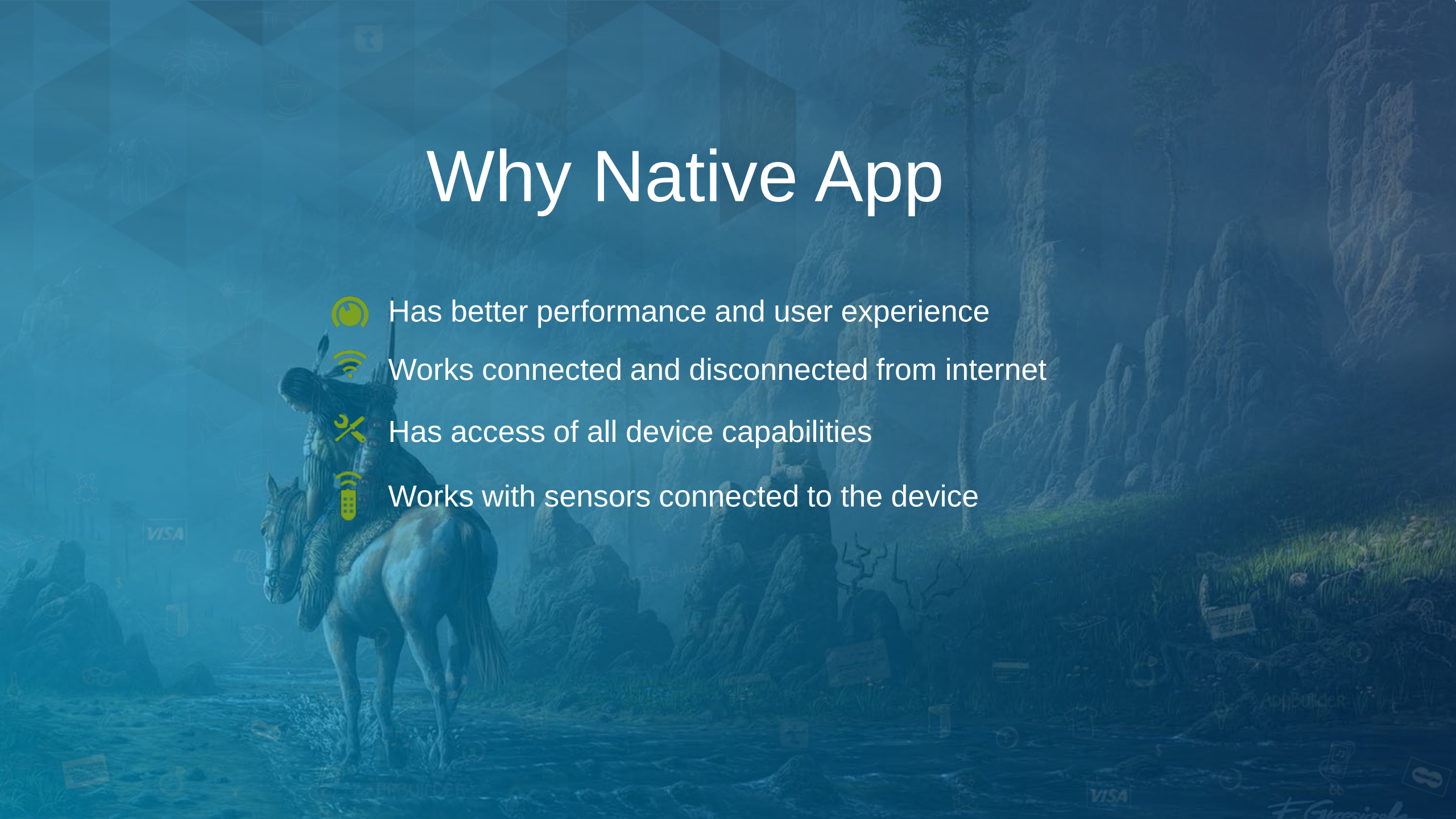
Create native iOS, Android, Mac and Windows apps in C#. Anything you can do in Objective-C, Swift or Java, you can do in C#.

Develop in Lua, use the same code-base and go cross-platform.

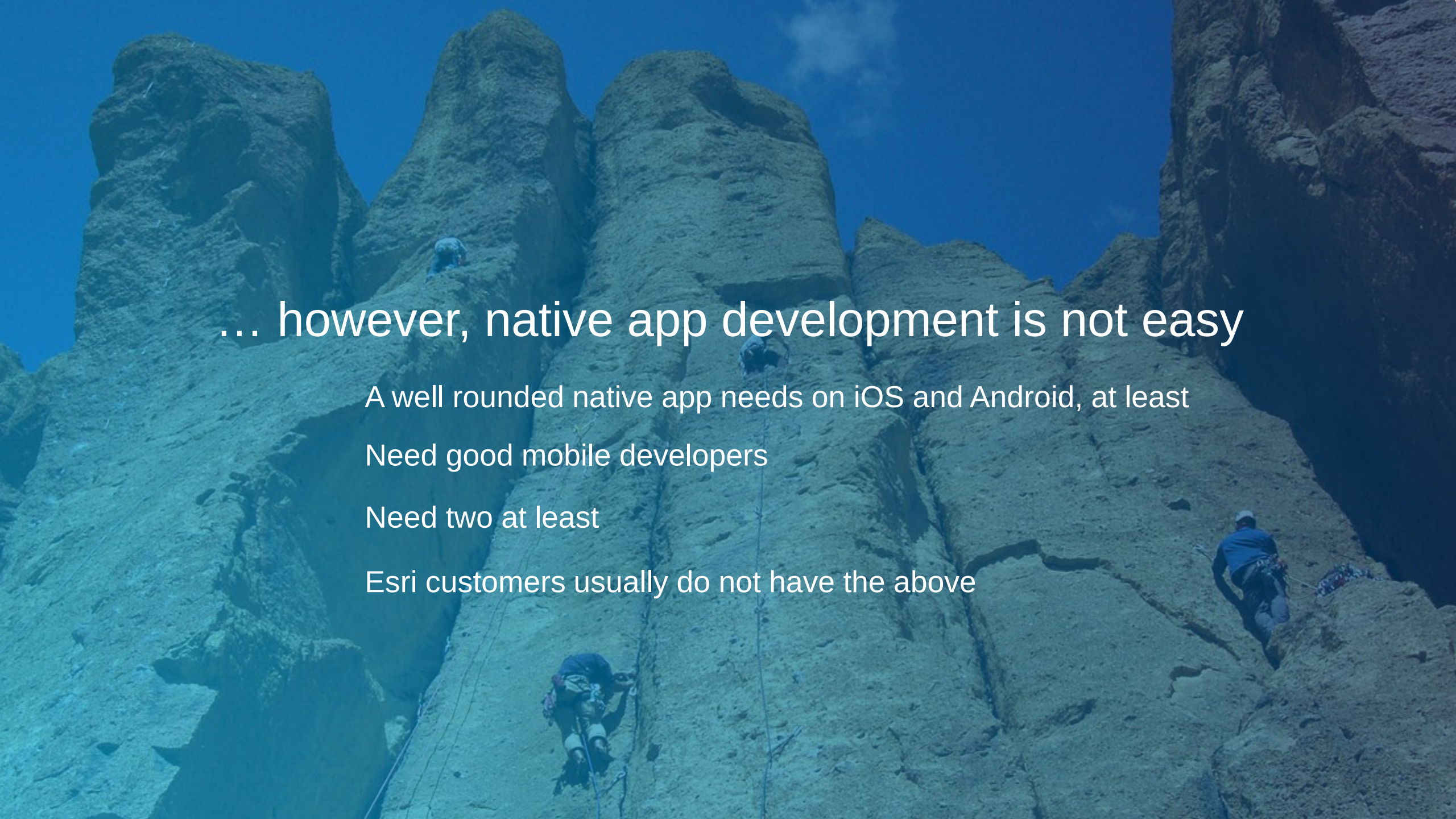
Deploying HTML5, native, or hybrid mobile apps using open and standards-based `tools.

- **Build native apps for iOS and Android using JavaScript**
- **Native UI, push, analytics, login modules "out-of-the box"**
- **Cloud build service, no need to setup Eclipse / Xcode**
- **Update your app without re-submitting to the App Store**
- **Open platform, extend with hooks, write native modules**

Why Native App



- Has better performance and user experience
- Works connected and disconnected from internet
- Has access of all device capabilities
- Works with sensors connected to the device



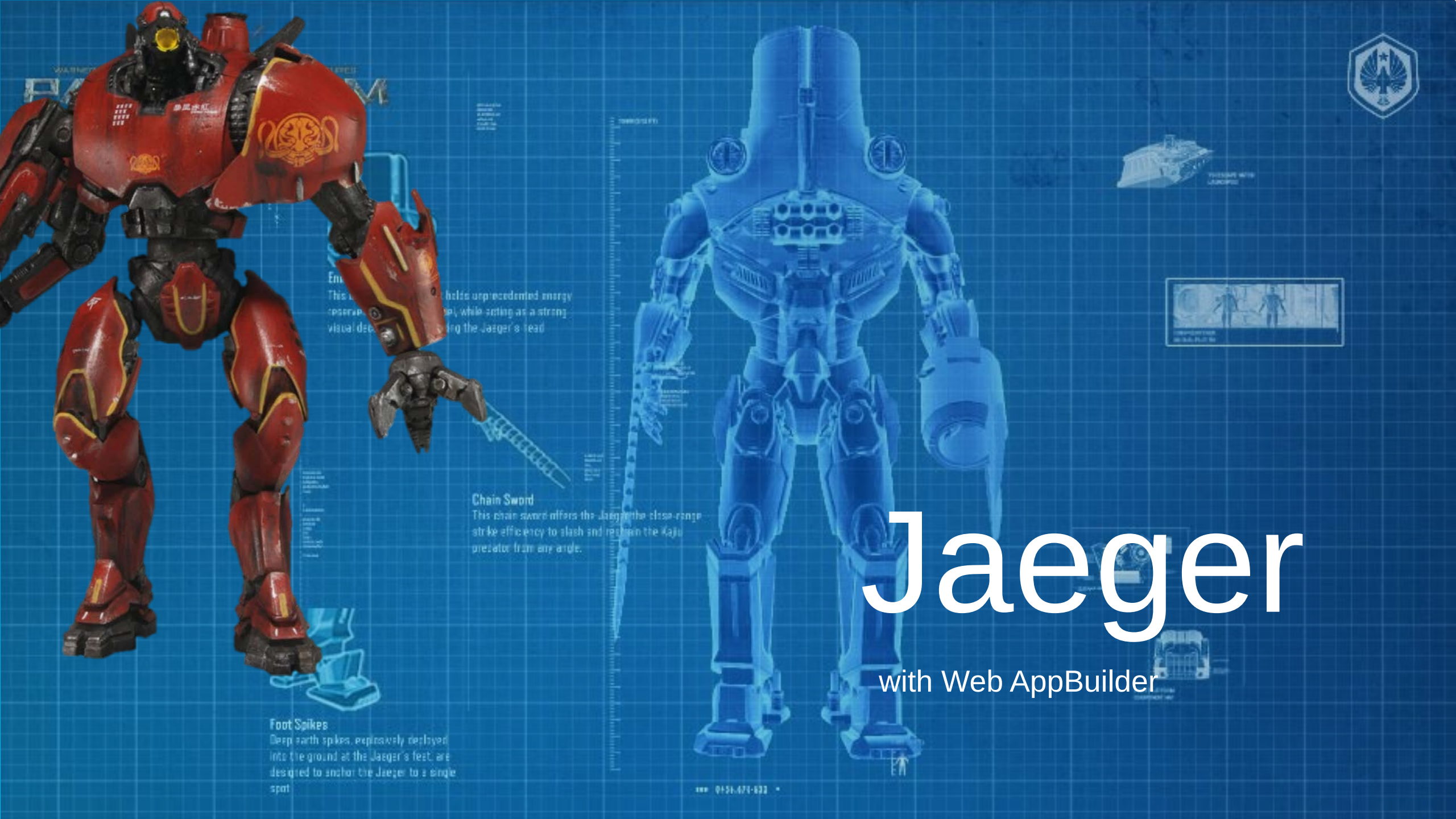
… however, native app development is not easy

A well rounded native app needs on iOS and Android, at least

Need good mobile developers

Need two at least

Esri customers usually do not have the above



Jaeger

with Web AppBuilder


Jaeger
Rapid mobile application development
A Jaeger app runs in pure native Runtime SDK includes all interactive and UI
A Jaeger app is built via configuration using Web AppBuilder for ArcGIS
Developers extend Jaeger by creating widgets using ArcGIS Runtime SDK

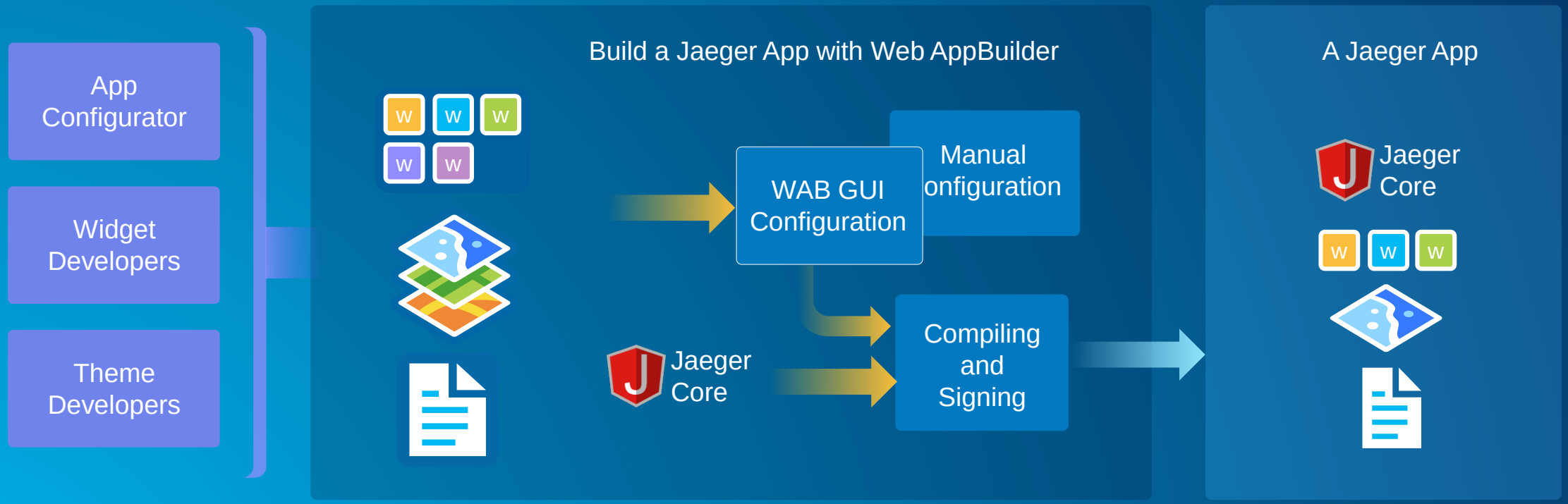
A Jaeger App

Panels with UI elements
As part of the Jaeger theme, panels and a fix number of UI elements are provided

Widgets
Functional modules that contain only programming logics without dealing with user interactive and visualization

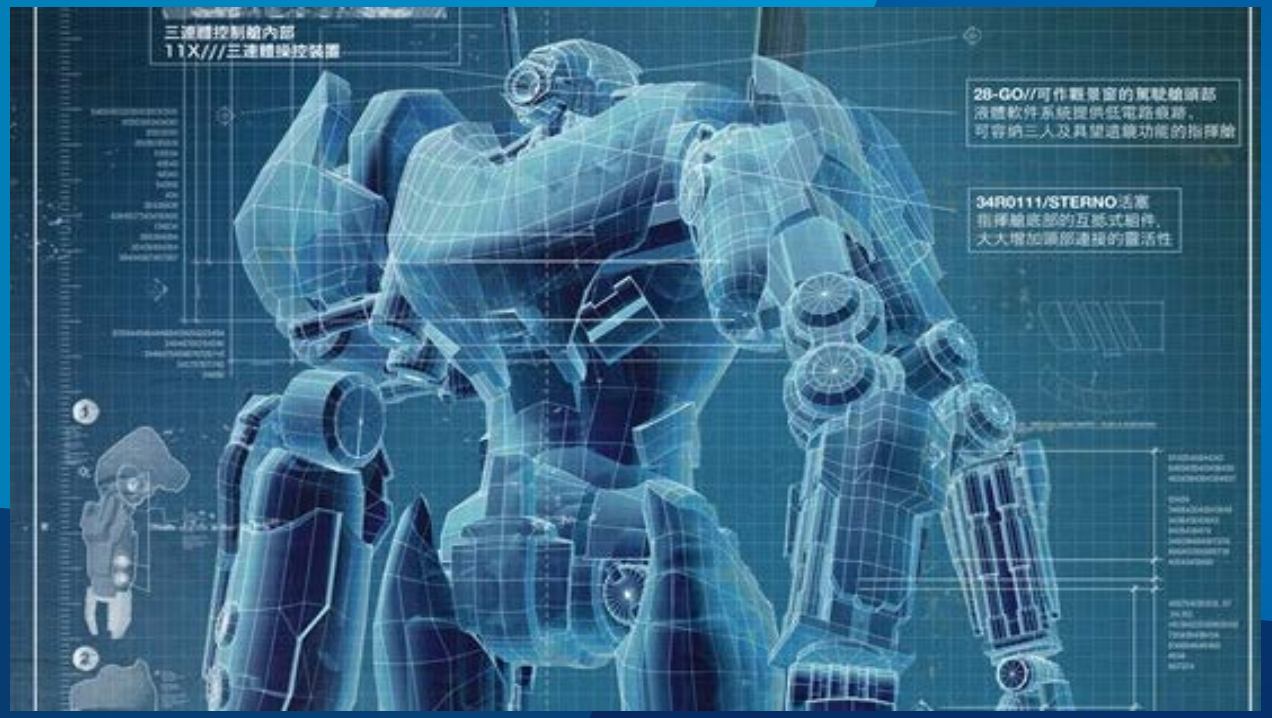
Map and events
All map, events handling and visual aspect of Jaeger are handled in native core. No performance loss

How a Jaeger App is Built



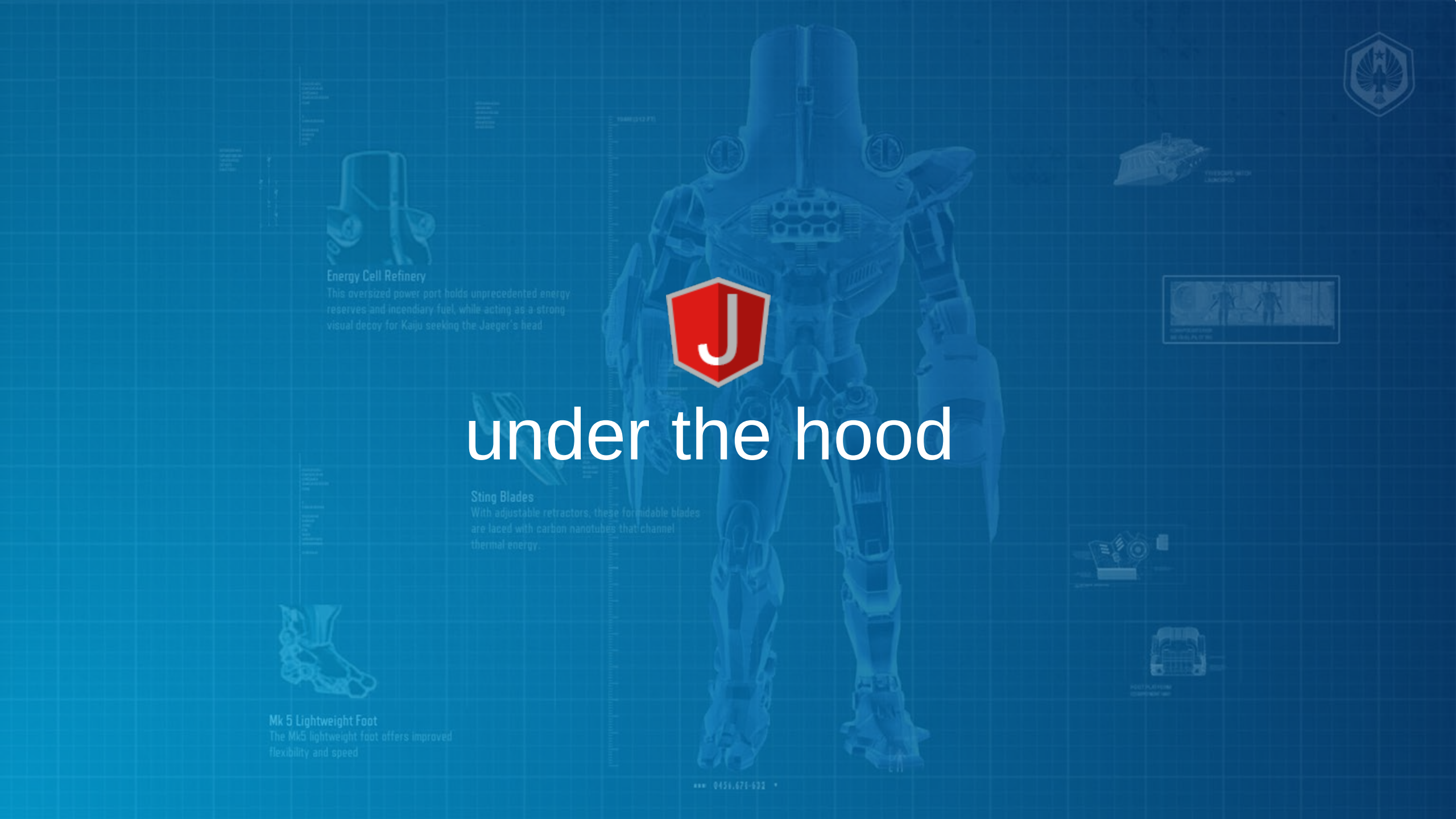
Jaeger for iOS

Jaeger for Android



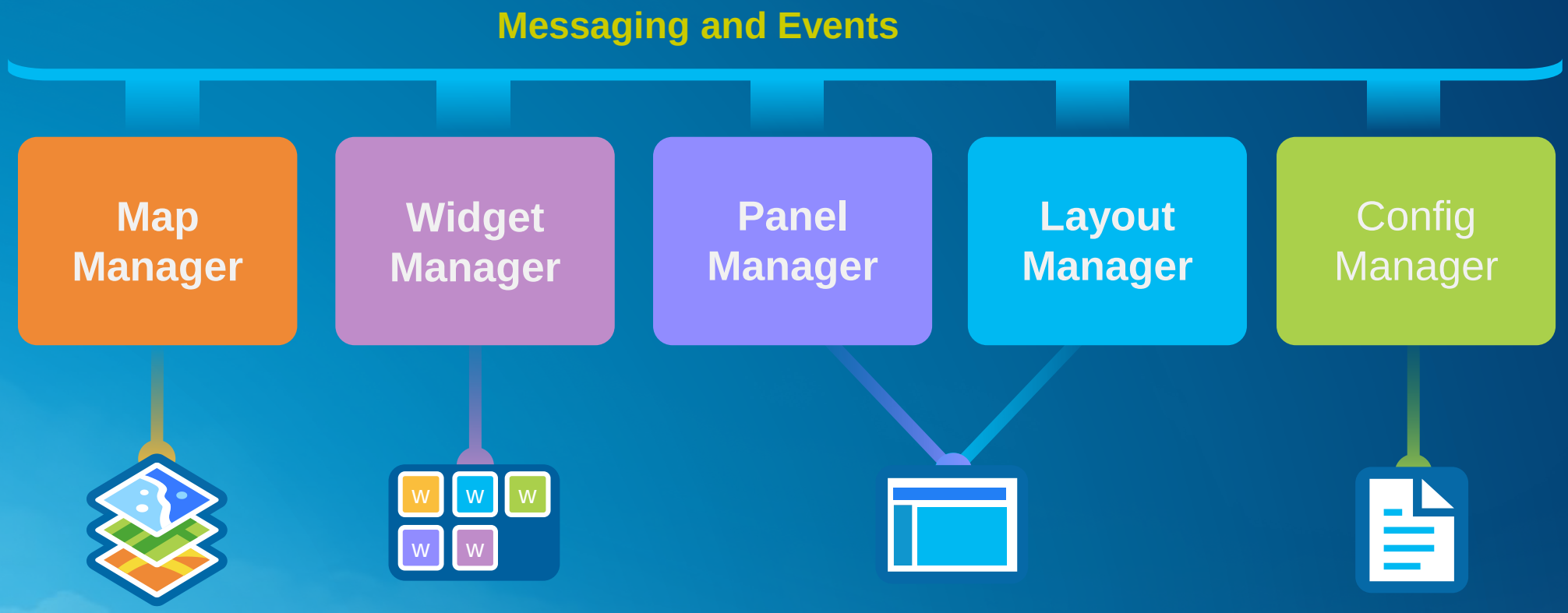
Create a Mobile App

for iOS and Android devices



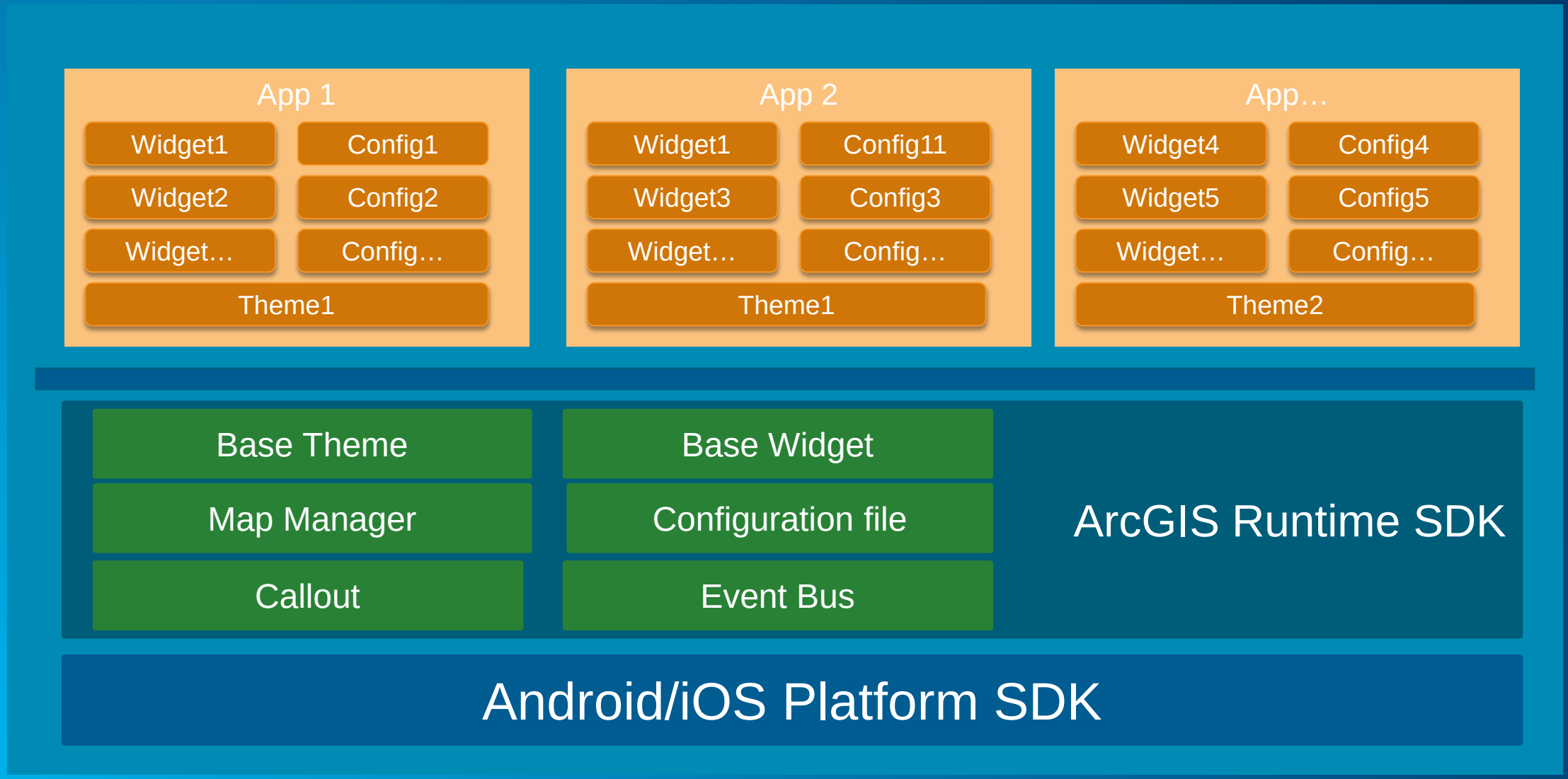
under the hood

Inside Jaeger App

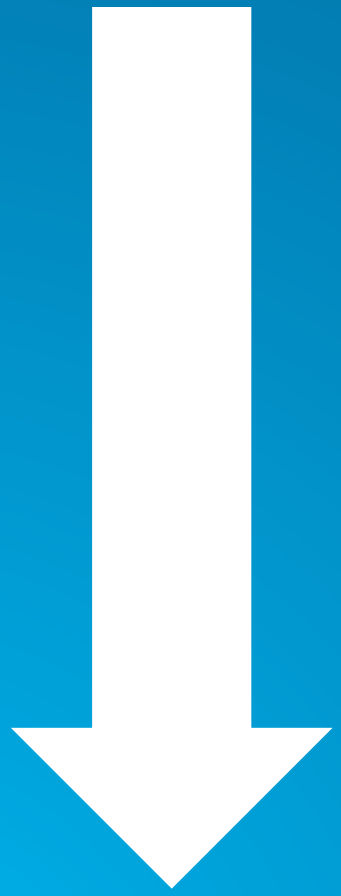



Jaeger
for
developers
Develop Widgets with ArcGIS Runtime SDK

Jaeger Development Architecture



Widget Lifecycle



App Start		`Widget.onCreate()`
Two Layouts	Screen	`Customized Action`
	Pool	`Widget.onSelected()`
App Exit		`Widget.onDestroy()`

Demo: Widget Development

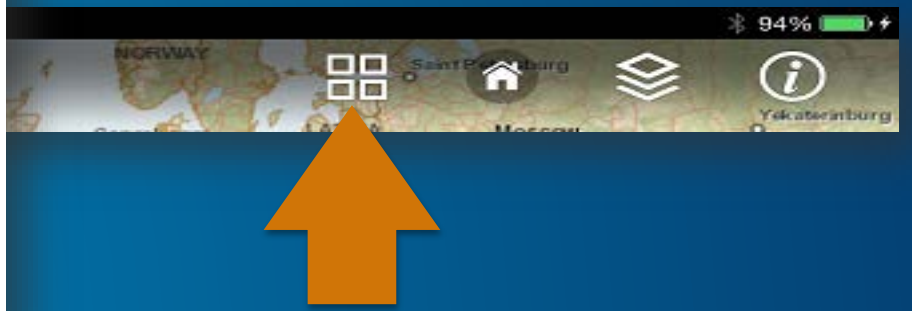
```
8
9   #import "BaseWidget.h"
10  @interface ZoomIn : BaseWidget
11
12  @end
13
```

ZoomIn.h

```
11
12  @interface ZoomIn()
13  @end
14  @implementation ZoomIn
15  -(void)selected
16  {
17      [super selected];
18      [self.mapView zoomIn:YES];
19  }
20  @end
21
```

ZoomIn.m

```
"widgetPool": {
  "widgets": [
    {
      "label": "Zoom In",
      "uri": "ZoomIn"
    },
```

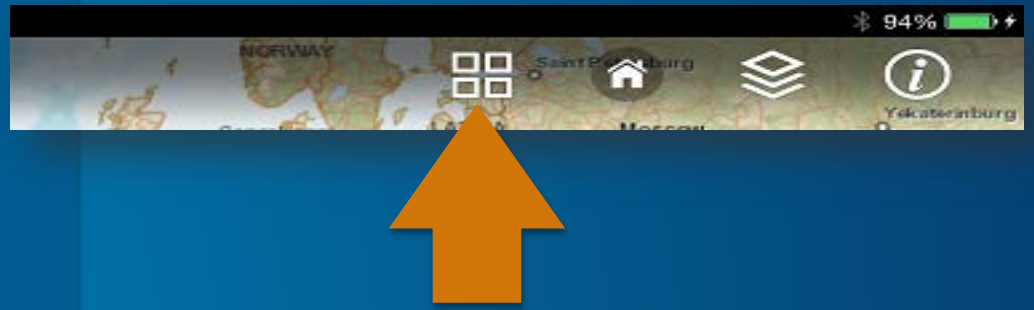


Widget Development

```
//ZoomIn.h
@interface ZoomIn : BaseWidget
@end
```

```
//ZoomIn.m
@implementation ZoomIn
-(void)selected
{
    [super selected];
    [self.mapView zoomIn:YES];
}
```

```
"widgetPool": {
  "widgets": [
    {
      "label": "Zoom In",
      "uri": "ZoomIn"
    },
```



Hello World

```
11
12  @interface HelloWorld()
13  @end
14  @implementation HelloWorld
15  -(void)create
16  {
17      [super create];
18      UIButton * button = [UIButton buttonWithType:UIButtonTypeCustom];
19      [button setFrame:CGRectMake(40, 20, 200, 44)];
20      [button setTitle:@"HelloWorld" forState:UIControlStateNormal];
21
22      [button setBackgroundColor:[UIColor grayColor]];
23      [button addTarget:self action:@selector(helloWorld:) forControlEvents:
            (UIControlEventTouchDown)];
24      [self.dataPanelView addSubview:button];
25
26  }
27  -(void)selected
28  {
29      [super selected];
30      [self showDataPanel];
31  }
32  -(void)helloWorld:(id)sender
33  {
34      UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@""
35                                                      message:@"Hello World!"
36                                                     delegate:self
37                                            cancelButtonTitle:@"Cancel"
38                                            otherButtonTitles:nil];
39      [alert show];
40  }
41  @end
42
```

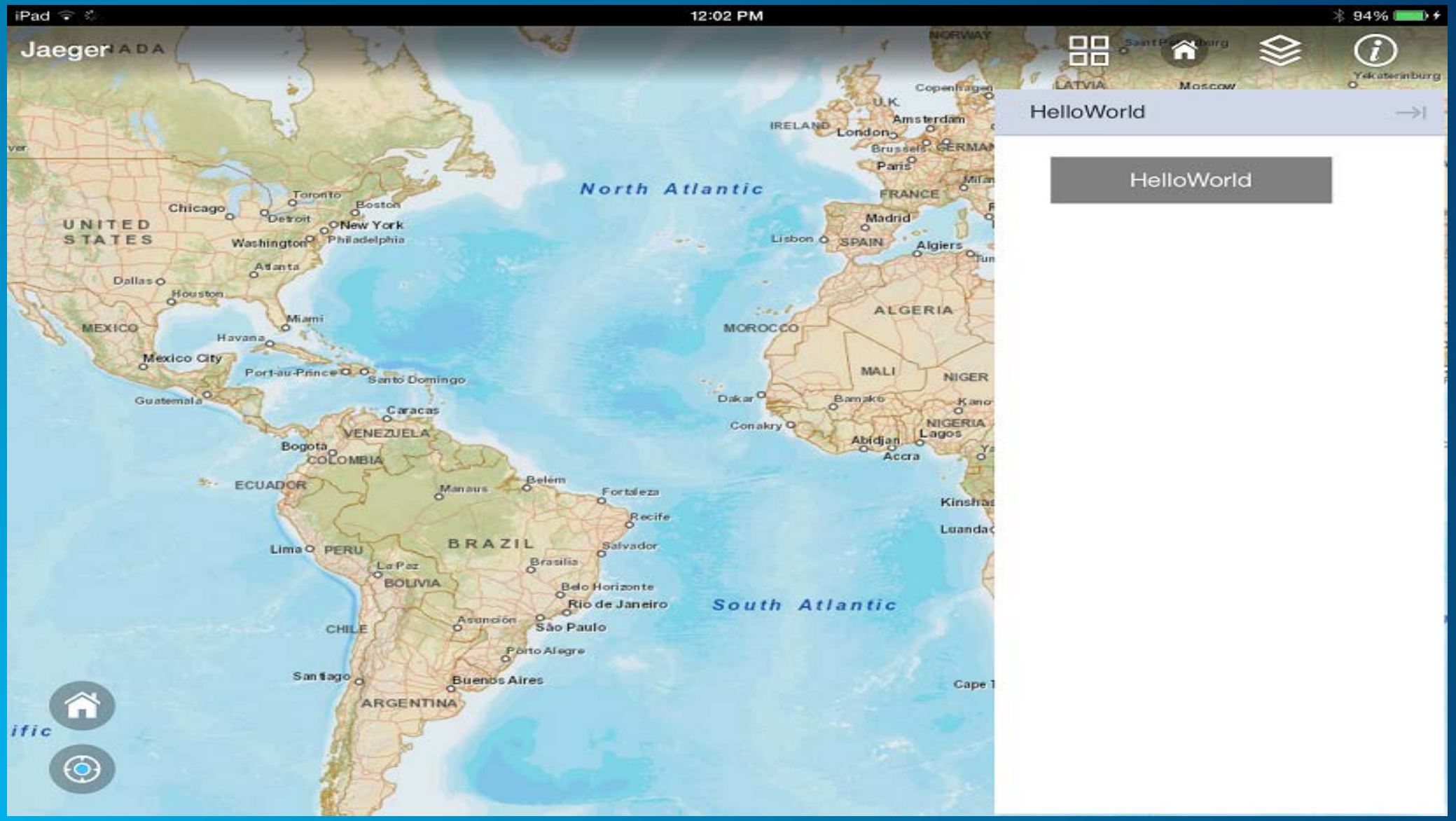
```
"widgetPool": {
  "widgets": [
    {
      "label": "HelloWorld",
      "uri": "HelloWorld"
    },
```

Hello World

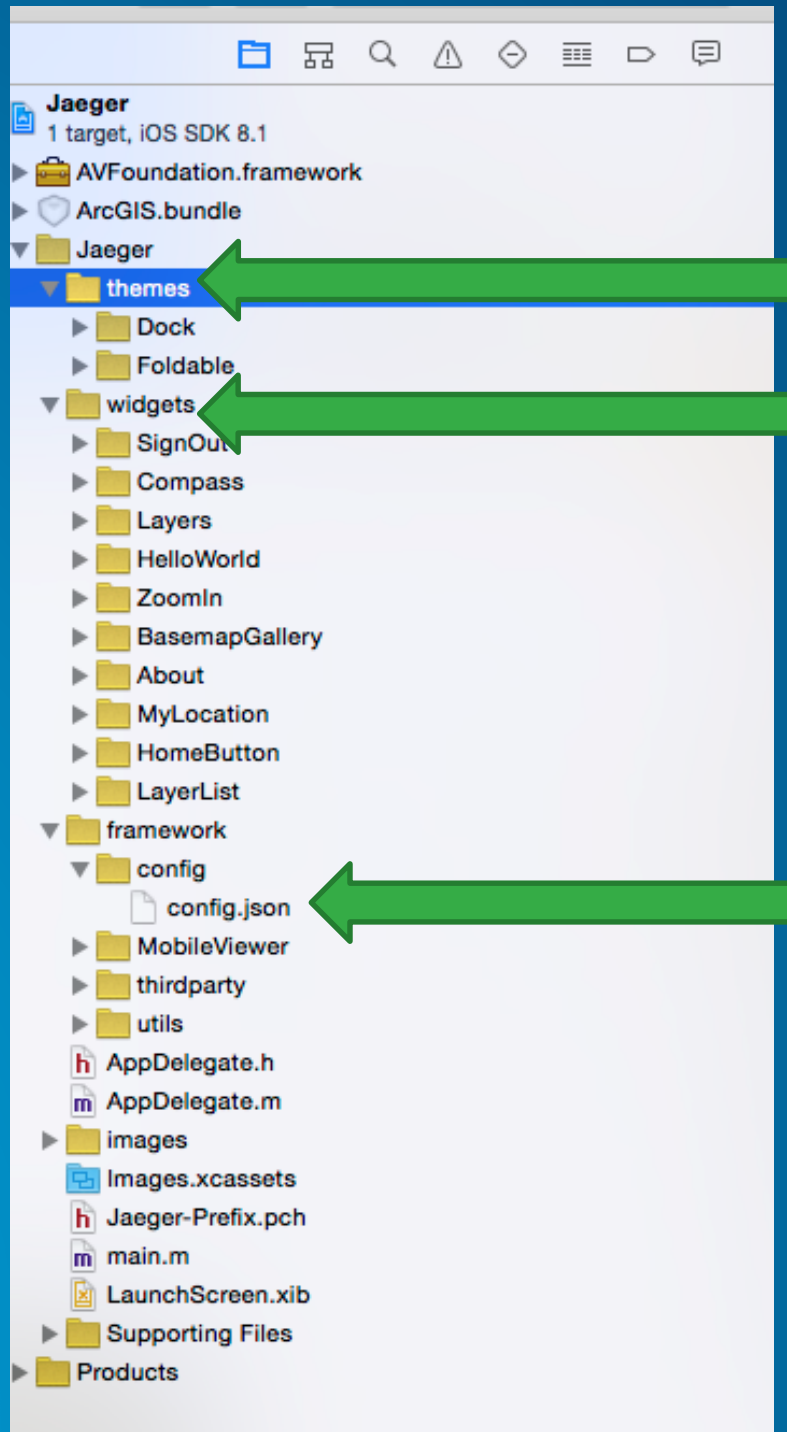
```
//HelloWorld.h
@interface HelloWorld : BaseWidget
@end

//HelloWorld.m
@implementation HelloWorld
-(void)create
{
    [super create];
    UIButton * button = [UIButton buttonWithType:UIButtonTypeCustom];
    [button setFrame:CGRectMake(40, 20, 200, 44)];
    [button setTitle:@"HelloWorld" forState:UIControlStateNormal];

    [button setBackgroundColor:[UIColor grayColor]];
    [button addTarget:self action:@selector(helloWorld:) forControlEvents:(UIControlEventTouchDown)];
    [self.dataPanelView addSubview:button];
}
-(void)selected
{
    [super selected];
    [self showDataPanel];
}
-(void)helloWorld:(id)sender
{
    [self.mapView zoomIn:YES];
}
```

```
"widgetPool": {
  "widgets": [
    {
      "label": "HelloWorld",
      "uri": "HelloWorld"
    },
```

Hello World


iPad
12:02 PM
94%
Jaeger
HelloWorld
HelloWorld
North Atlantic
South Atlantic
UNITED STATES
MEXICO
Chicago
Toronto
Boston
Detroit
New York
Washington
Philadelphia
Atlanta
Dallas
Houston
Miami
Havana
Mexico City
Port-au-Prince
Santo Domingo
Guatemala
Caracas
VENEZUELA
Bogota
COLOMBIA
ECUADOR
Lima
PERU
BOLIVIA
CHILE
Santiago
Buenos Aires
ARGENTINA
BRAZIL
Manaus
Belém
Fortaleza
Recife
Salvador
Brasilia
Belo Horizonte
Rio de Janeiro
São Paulo
Asunción
Porto Alegre
La Paz
IRELAND
U.K.
London
Amsterdam
Brussels
Paris
FRANCE
Madrid
Lisbon
SPAIN
Algiers
MOROCCO
ALGERIA
MALI
NIGER
Dakar
Bamako
Conakry
NIGERIA
Lagos
Abidjan
Accra
Moscow
Yekaterinburg
LATVIA
Copenhagen

Share the Widgets and Themes for Mobile



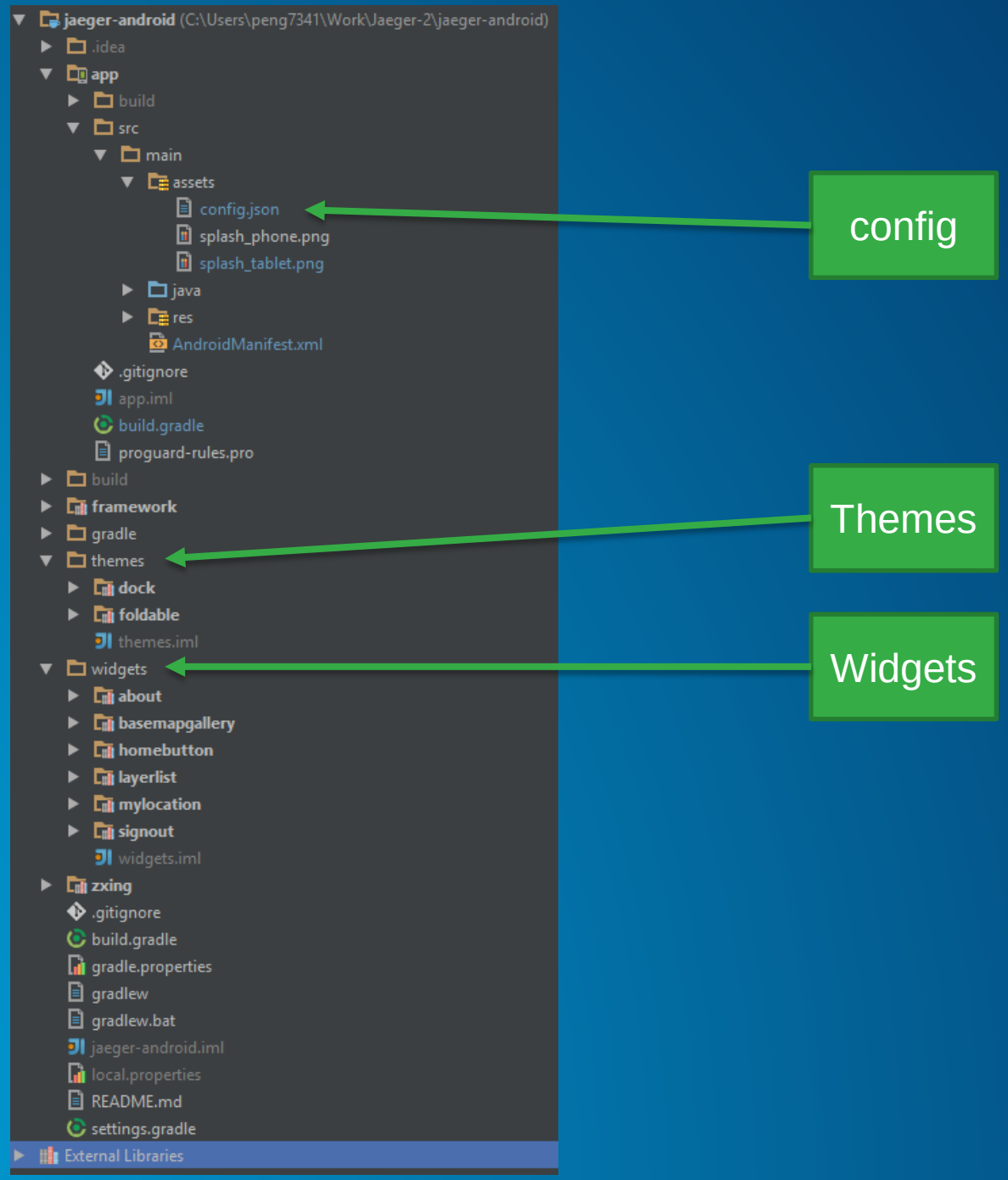
Themes

Widgets

config

Share Widget & Theme (iOS)

- Each Widget/Theme is a standalone library
- All the modules have their own resources & configuration files
- To share Widget/Theme, just share the library and resources


jaeger-android (C:\Users\peng7341\Work\Jaeger-2\jaeger-android)
.idea
app
build
src
main
assets
config.json
splash_phone.png
splash_tablet.png
java
res
AndroidManifest.xml
.gitignore
app.iml
build.gradle
proguard-rules.pro
build
framework
gradle
themes
dock
foldable
themes.iml
widgets
about
basemapgallery
homebutton
layerlist
mylocation
signout
widgets.iml
zxing
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
jaeger-android.iml
local.properties
README.md
settings.gradle
External Libraries
config
Themes
Widgets

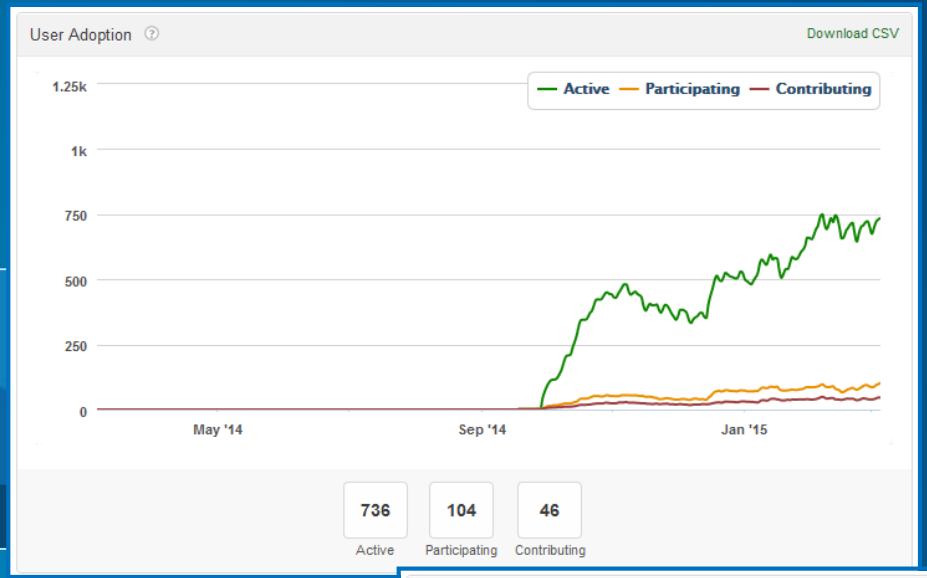
Share Widget & Theme (Android)

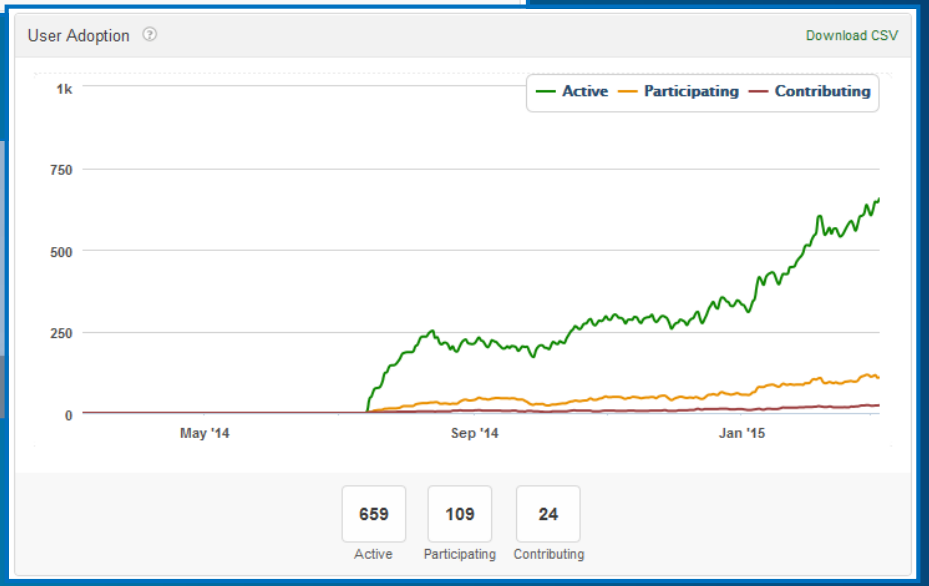
- Each Widget/Theme is a standalone module
- All the modules have their own resources & configuration files
- To share Widget/Theme, just export the module using IDE

Web AppBuilder Community

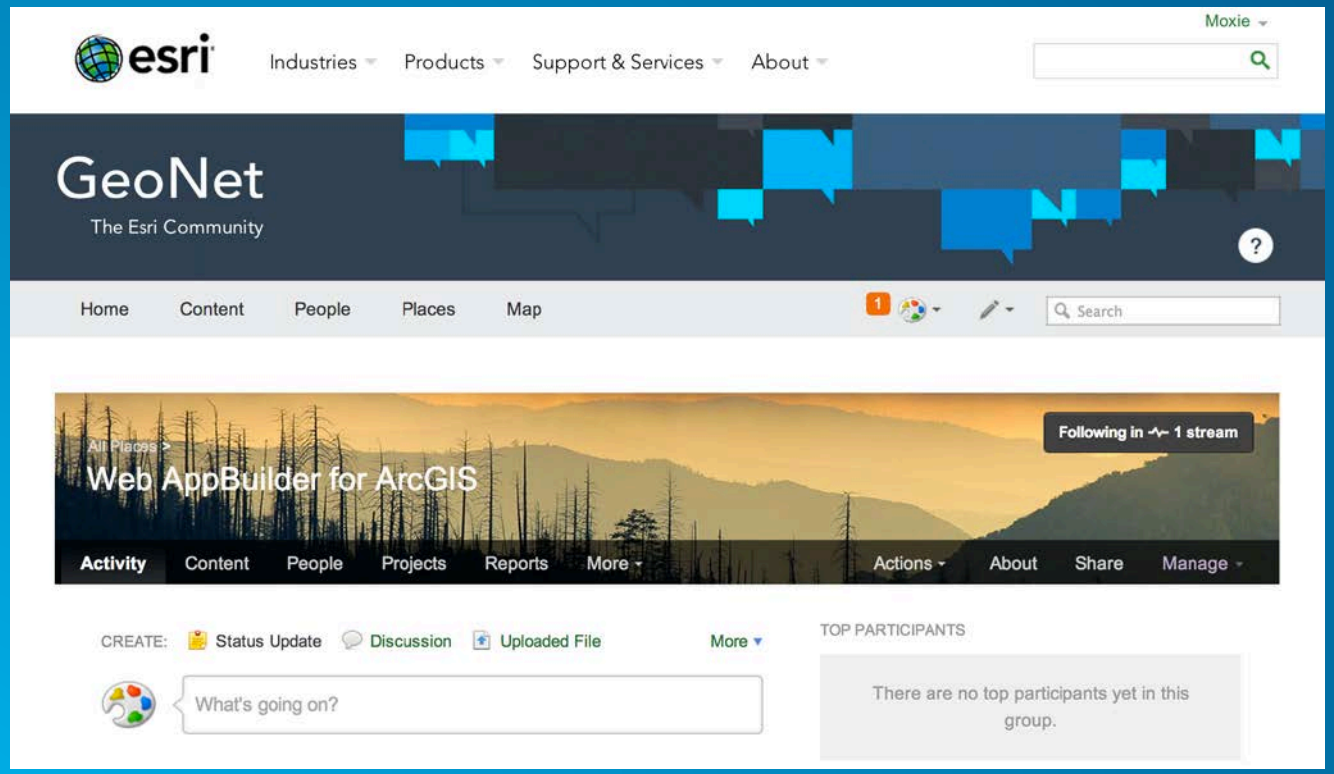
on GeoNet

GeoNet – Esri Community



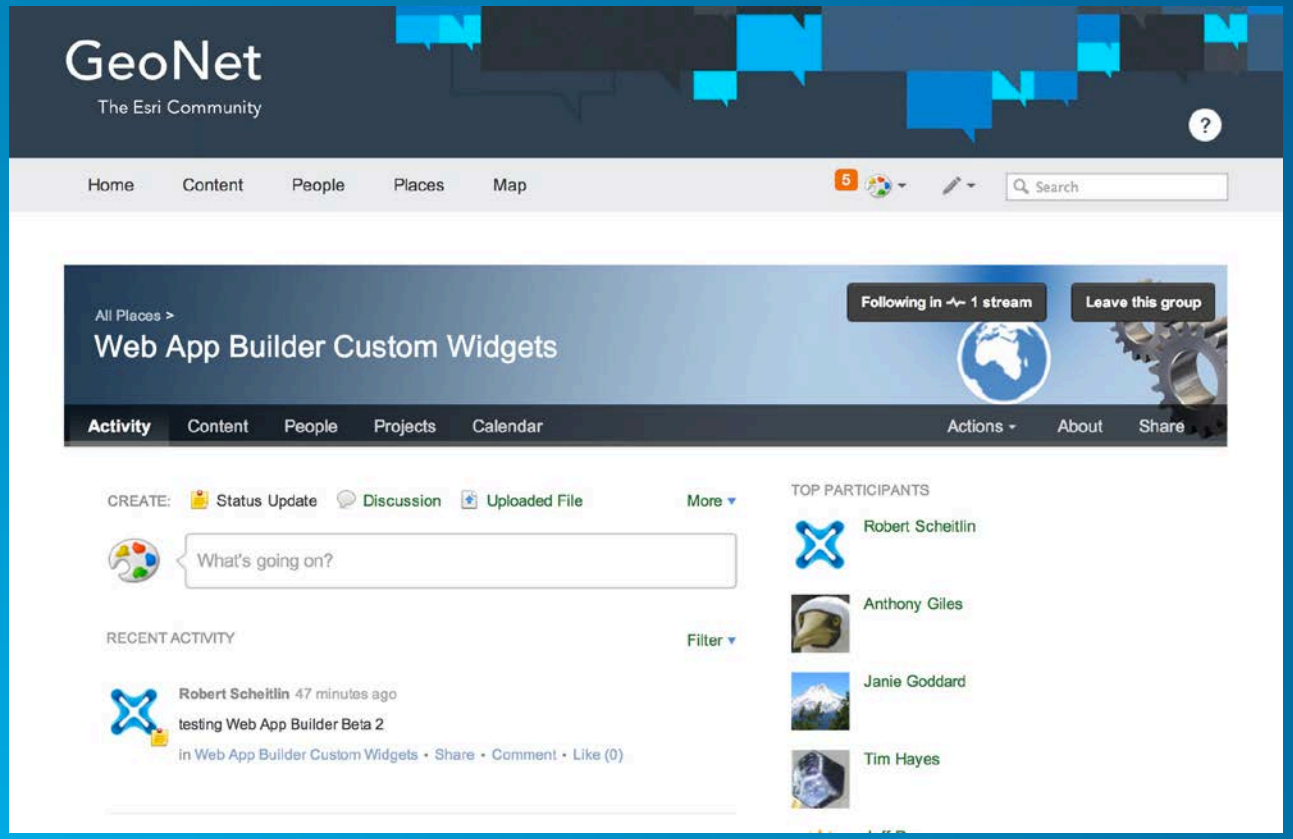


Esri GeoNet Group Web AppBuilder
https://geonet.esri.com/groups/web-appbuilder



GeoNet Group Web AppBuilder Custom Widgets

https://geonet.esri.com/groups/web-app-builder-custom-widgets



Rate This Session

www.esri.com/RateMyDevSummitSession