



Operations Dashboard

# Extending the Operations Dashboard

Jay Chen

Tif Pun

Overview of Operations Dashboard Extensibility

# Agenda

Demo – building a map tool

Demo – building a widget

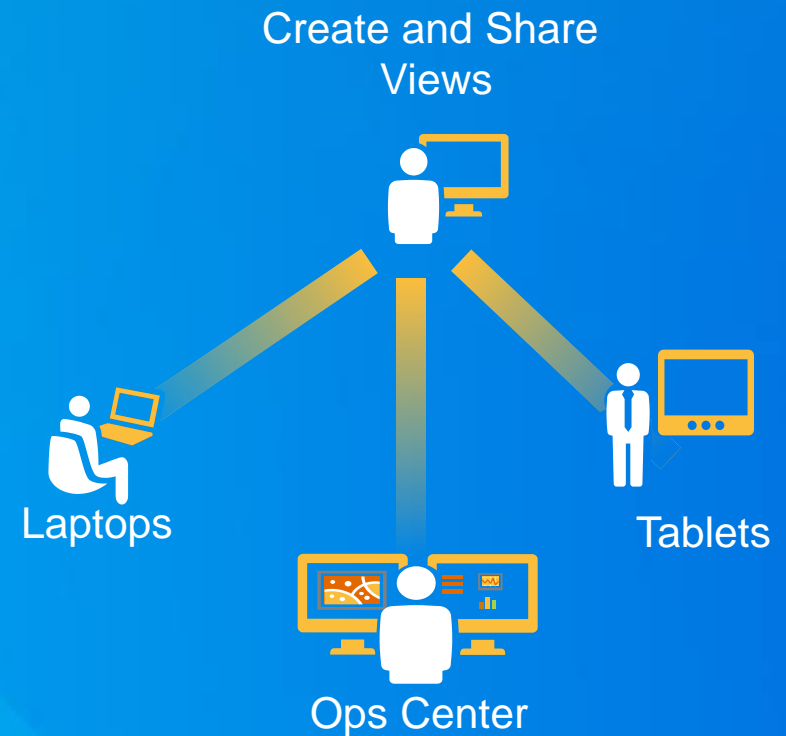
Demo - deployment

Roadmap for extensibility

Questions

# Operations Dashboard for ArcGIS

- **Monitor and manage your operations**
- **Ready to use widgets publishers can configure**
  - Maps, Lists
  - Bar Charts, Gauges
  - Histograms, Indicators
  - Details
- **Customer Use**
  - Monitor day to day operations
  - Manage planned and unplanned events
  - Visualize key performance indicators



# Creating and Using Operations Views

## Components

- Widgets
- Map Tools
- Feature Actions
- Data Sources

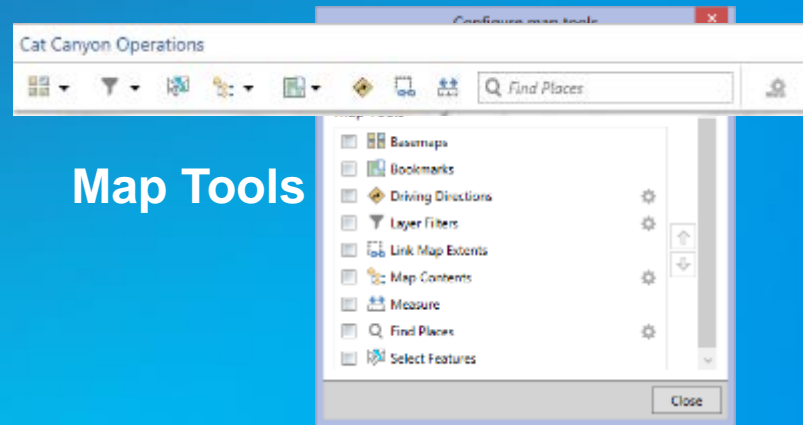
## Dynamic, Role Based Access

- Widgets are dynamic, interactive
- Maps provide spatial context
- Create and Share Focused views...

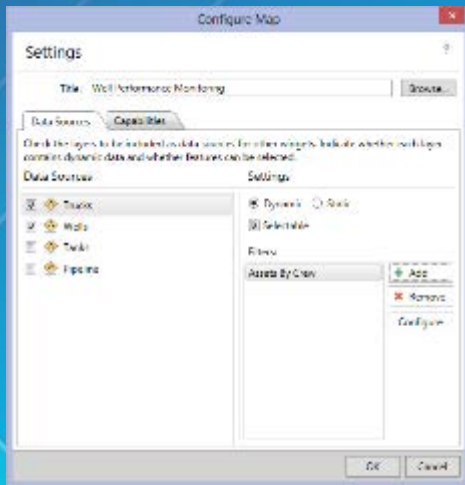
## Two Types of Views

- Single Display
- Multiple Display

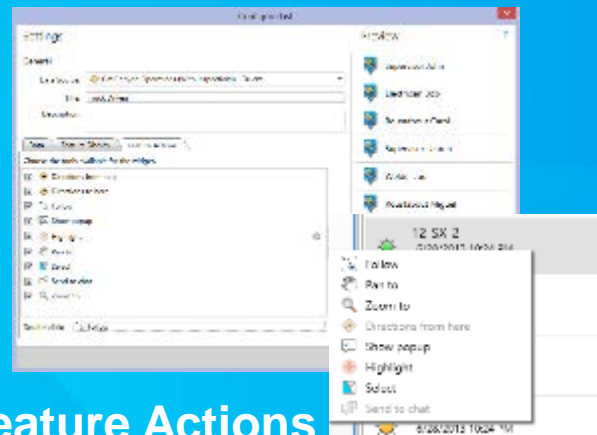
# Composition of an Operation View...



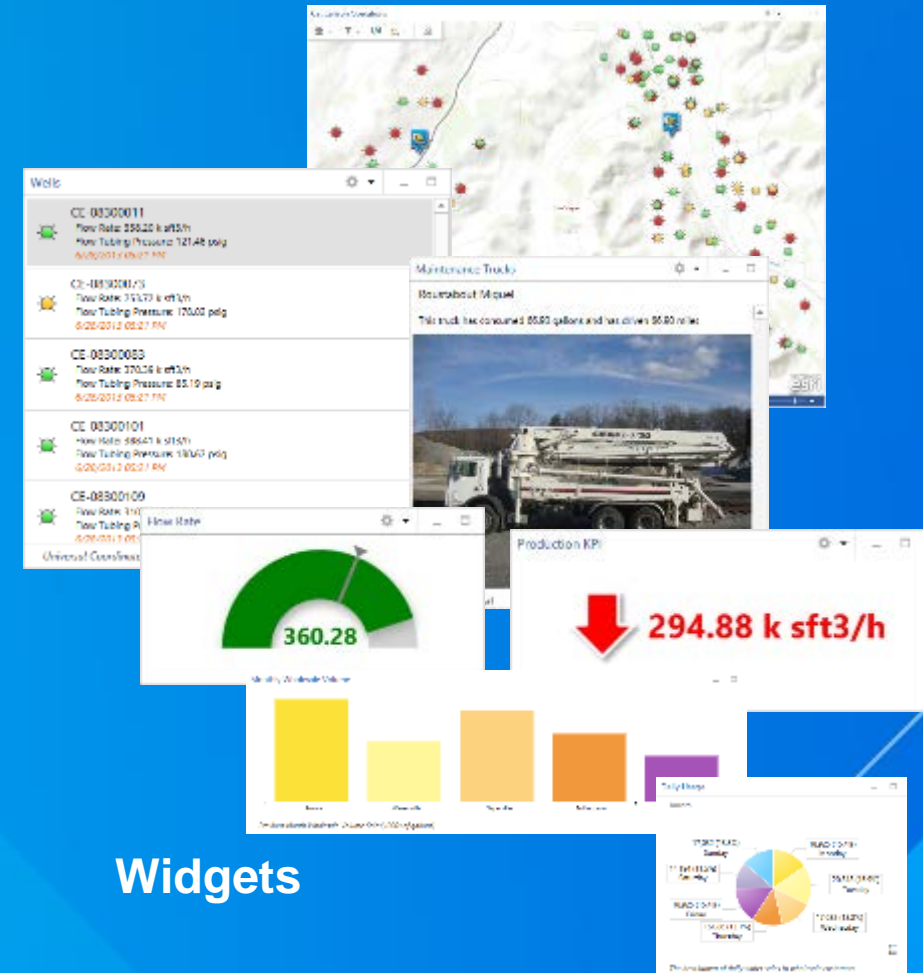
Map Tools



Data Sources

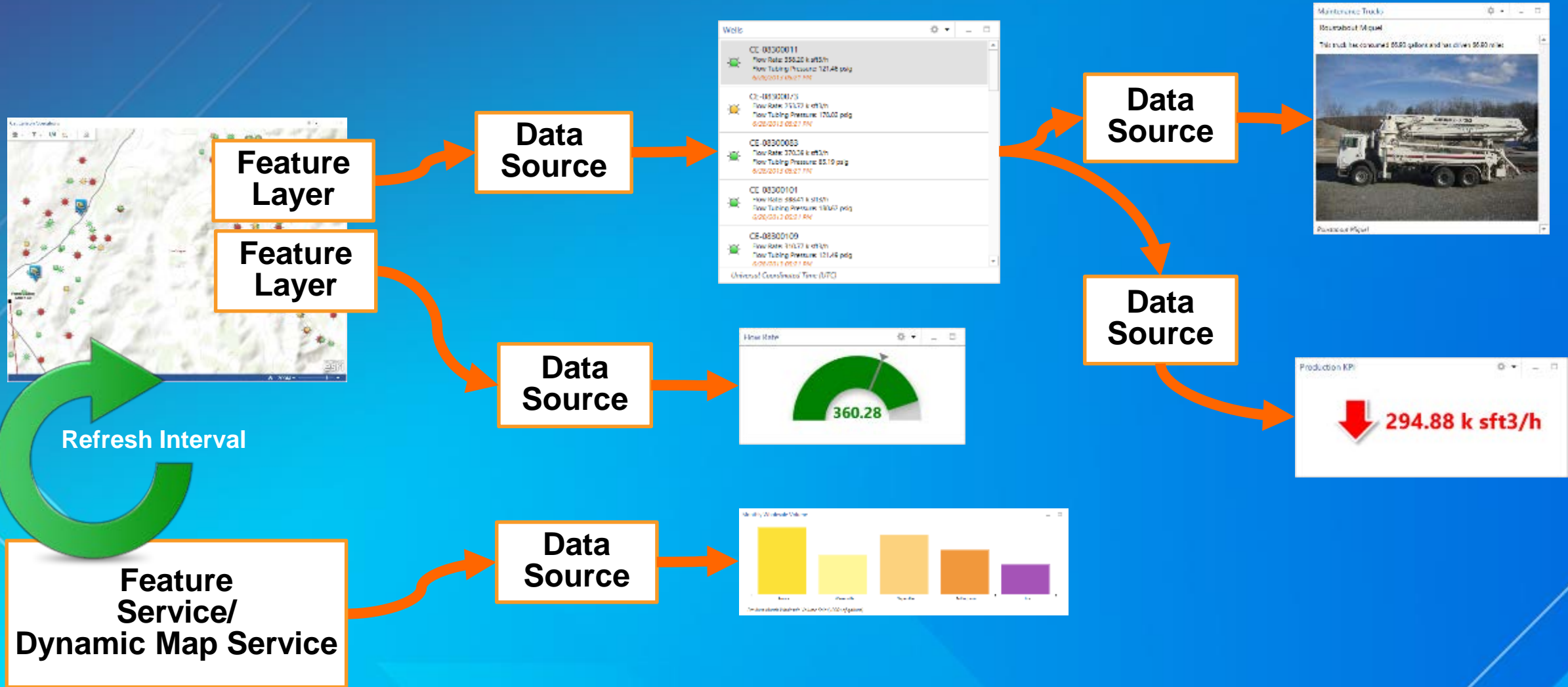


Feature Actions



Widgets

# Data Sources



# Configuring an Operation View...

The image shows two overlapping dialog boxes for configuring widgets. The top dialog is titled "Configure Gauge" and has a "Settings" tab. It includes a "Data Source" dropdown set to "Casco Bay Ferries Map-All Internal - Casco Bay", a "Title" field with "Average Passenger Capacity", and a "Description" field. The "Display Type" is set to "Statistic". The "Preview" pane shows two gauge charts: one for "Above Threshold" with a red needle and one for "Below Threshold" with a blue needle. The "Data" tab is selected, showing "Value Field" as "NumberPassengers" and "Operation" as "Average". The bottom dialog is a "Settings" window for "Casco Bay Ferries Map-All Internal - Casco Bay", with the "Data Sources" tab active. It lists several data sources with checkboxes: "Casco Bay Ferries", "Commercial Vessels", "Ferry Terminals", "Ferry Routes", and "Shipping Lanes".

Widgets

# Map Tools

The image shows two overlapping dialog boxes for configuring map tools. The top dialog is titled "Configure map tools" and has a "Settings" tab. It includes a search bar "Enter a text to filter" and a list of map tools with checkboxes: "Basemaps", "Bookmarks", "Layer Filters", "Map Contents", "Find Places", and "Select Features". The bottom dialog is titled "Configure List" and has a "Settings" tab. It includes a search bar "Filter" and a list of features with checkboxes: "Ridley Cove", "Maquoit II", "Autocisco III", "Autocisco IV", "Quahog Bay", "Malaga", "Broadcove", and "Machigonne II". The "Preview" pane shows a list of these features with their current capacity percentages. The "Feature Actions" tab is active, showing a list of actions with checkboxes: "Show pop-up", "Highlight", "Directions from here", "Directions to here", "Follow", "Pan to", and "Zoom to".

Feature Actions

# Extending Operations Dashboard

The screenshot displays the 'Casco Bay Ferry Operations (Dev Summit)' dashboard. On the left, a 'Ferries' widget lists various ferry routes with their current capacity percentages. Below this, a 'Details' section provides specific information for the 'Casco Sound' ferry, including the number of passengers, passenger capacity, and minutes behind schedule. At the bottom left, there is a photograph of a ferry. On the right side of the dashboard is a map of the Casco Bay area, showing ferry routes and various landmarks. A 'Map Tool' callout points to the map's search bar, and a 'Feature Action' callout points to a context menu overlaid on the map.

Ferry	Current Capacity
Casco Sound	88%
Merepoint	82%
Ridley Cove	79%
Maquoit II	79%
Autocisco III	71%

Details for Casco Sound Ferry:  
Time: 3/4/2015, 4:58:36 PM  
NumberPassengers: 229  
Passenger Capacity: 327  
Minutes Behind Schedule: 15

Map Tool: Place of Address

Feature Action: Show pop-up, Highlight, Follow, Pan to, Select, Zoom to

Widget

Map Tool

Feature Action

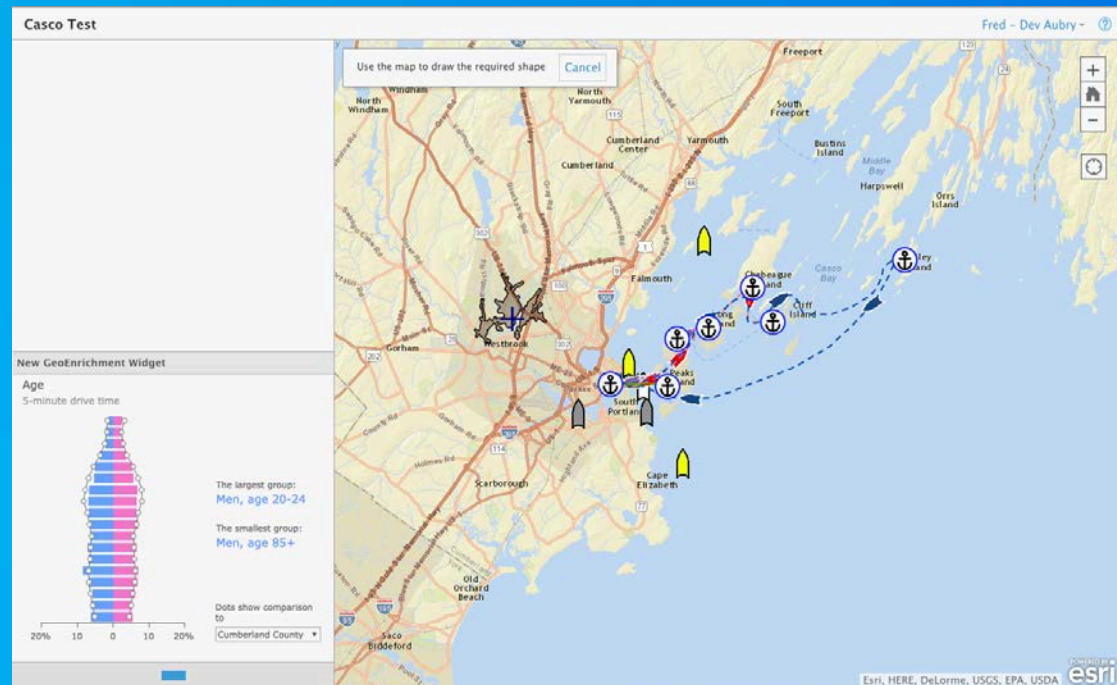
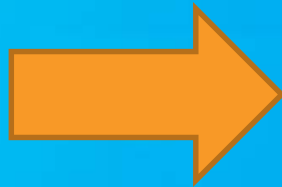


# An operations dashboard extension is...

- A web application loaded in
  - A web control in the Windows Operations Dashboard
  - A iframe in the Browser Operations Dashboard
- Code it once, run in both Windows and browser apps



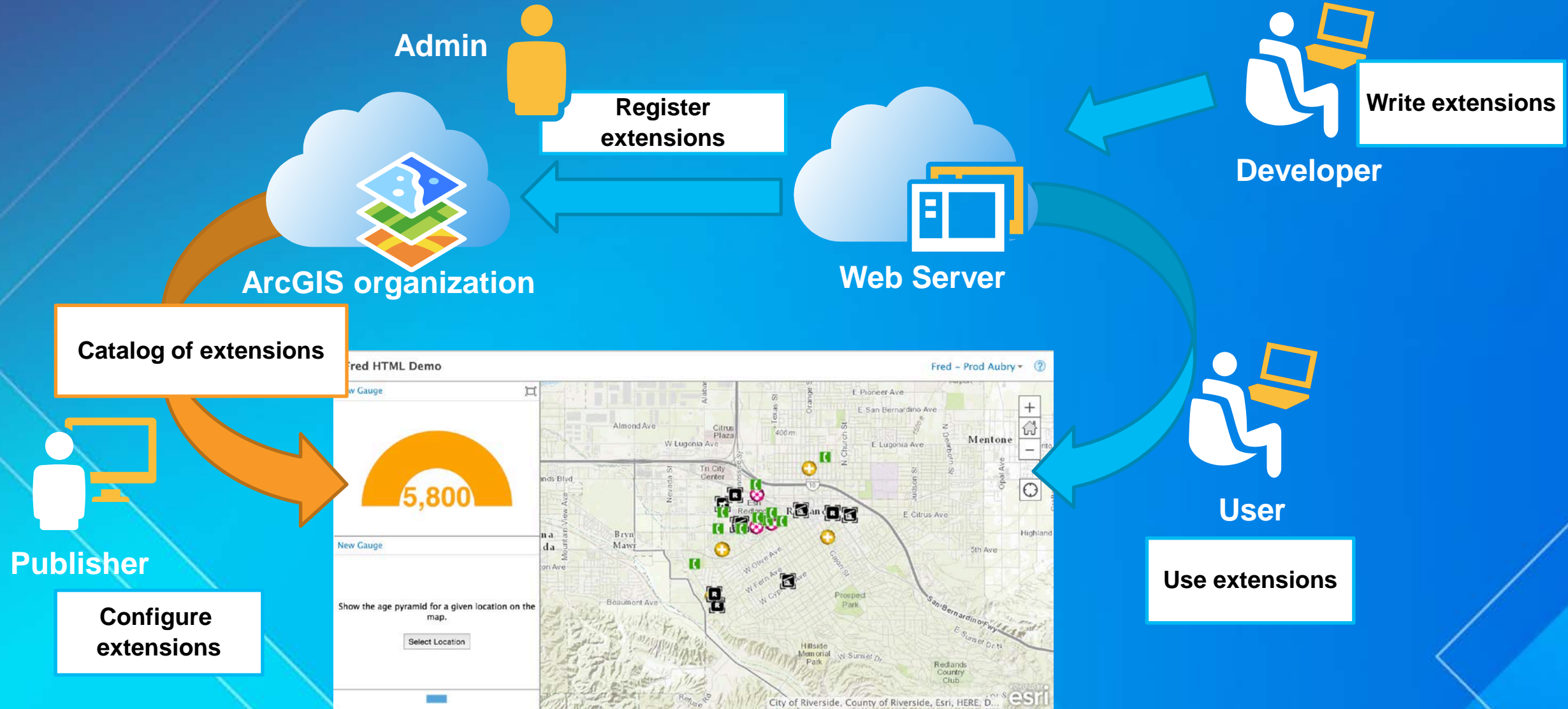
Web App  
HTML + JavaScript +  
manifest



# Making an extension available...



# Extending Operations Dashboard using Esri JavaScript API



## Custom Map Tools can...

- **Workflow in a map**
- **Only 1 tool at the time can be activated**
- **Associated UI is destroyed when deactivated (toolbar or dropdown)**
- **Great for temporary analysis/visualization**
  - **Geo-processing tool**
  - **Explore locations**
  - **Editing**

*Do something on the map...*

# Demo – Buffer Map Tool

The image displays a software interface for a Buffer Map Tool. It features a list of incidents on the left and a map on the right. The incident list includes a search filter and details for a specific event: "Spill of concentrated ammonia (Toxic!!)", which occurred on 7/22/2015 at 15:00:00 and was updated at 15:15:00. The map shows a street grid with a red warning icon at a specific location. Three concentric white circles represent buffers of 0.5km, 1km, and 2km. A tooltip above the map instructs the user to "Click on the map to create three buffers (0.5km, 1km, 2km) for the clicked location." and includes a "Done" button. The interface is titled "Toronto Traffic Cameras with Extensions - Tech session".

**Incidents**

Filter

Spill of concentrated ammonia (Toxic!!)  
Happened: 7/22/2015, 15:00:00  
Updated: 7/22/2015, 15:15:00

Click on the map to create three buffers (0.5km, 1km, 2km) for the clicked location. Done

Toronto Traffic Cameras with Extensions - Tech session

## Recap – Custom Map Tool

- An extension consists of a manifest, a HTML5 file, a HTML5 template, and optionally a JS file and a css file.
- Refresh the extension to get the changes in the HTML and JS files; Restart the app to get the changes in the manifest
- Life cycle of a custom map tool:
  - constructor → (postCreate) → hostReady → (main business logic) → deactivateMapTool
- Map tools are stateless
- Custom map tools usually work with geoprocessing services
- Custom map tools usually work with GraphicsLayerProxy
- It's possible to create configurable custom map tools

## Custom Widgets can...

- **Display additional data**
  - **Weather**
  - **Video**
  - **Related information found in a different data store**
- **Display data resulting from a geo-processing tool**
- **Keep a state**

*Widgets coexist with a map or its features...*

# Demo – Attribute table widget

The image displays a GIS application interface with two main components: a data table and a map. The top window, titled "Toronto Traffic Monitoring", shows a table with three rows of vehicle data. The bottom window, titled "Toronto Traffic Cameras with Extensions - Tech session", shows a similar table with three rows of camera data. Both windows include a map of Toronto with vehicle icons and traffic camera symbols.

**Toronto Traffic Monitoring**

Police has 3 features

GasLevel	Speed	Type	UnitName
53.4375	62.424	Unmarked Car	T456
57.76	51.5205	Patrol	D143
73.568	36.1	Pickup Truck	D152

**Toronto Traffic Cameras with Extensions - Tech session**

Police has 3 features

GasLevel	Speed	Type	UnitName
12.68096924	55.7006855	Unmarked Car	T456
14.64519457	57.40293596	Patrol	D143
38.82418166	44.569782	Pickup Truck	D152



## Recap – Custom Widgets

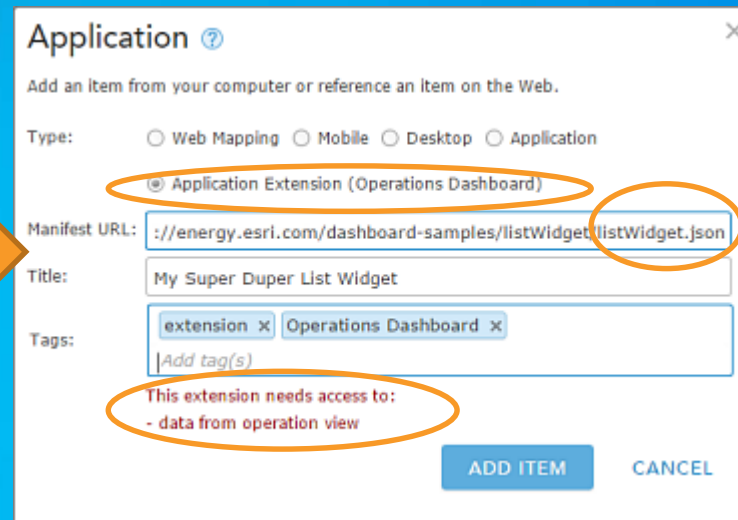
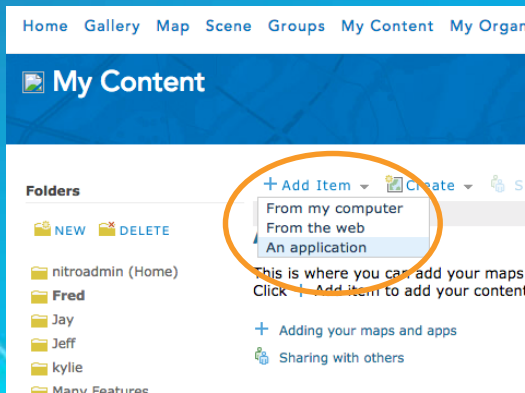
- To make a widget configurable, add a HTML5 file, a HTML5 template, (optionally) add a JS file, and update the manifest file
- Life cycle of a custom widget:
  - (constructor) → (postCreate) → hostReady → (main business logic)
- Configurations are stored in the `dataSourceConfig` object, which will then be written into the operation view when the view is saved.

## Deployment

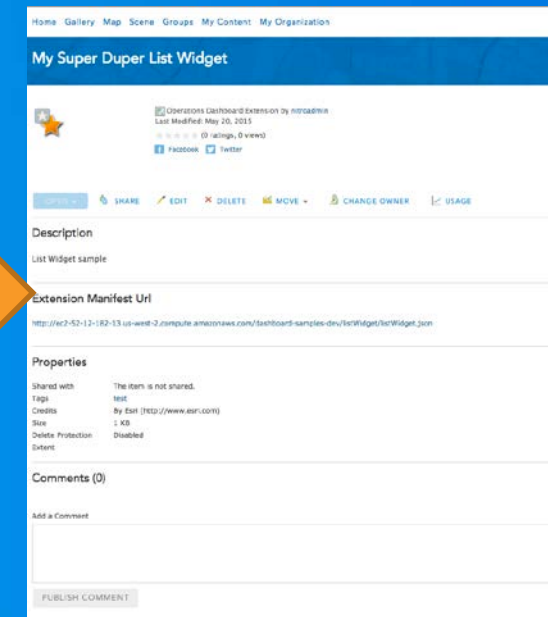
- **Requirements for web server**
  - Public facing
  - HTTPS/SSL/valid certificates
  - Enable CORS
  
- **Register and share extensions**

# Registering an extension in an organization

- The Admin for the organization will register the extension by creating an Operations Dashboard Extension Item.  
(reference the extension manifest <http://myDomain/myExtension.json>)

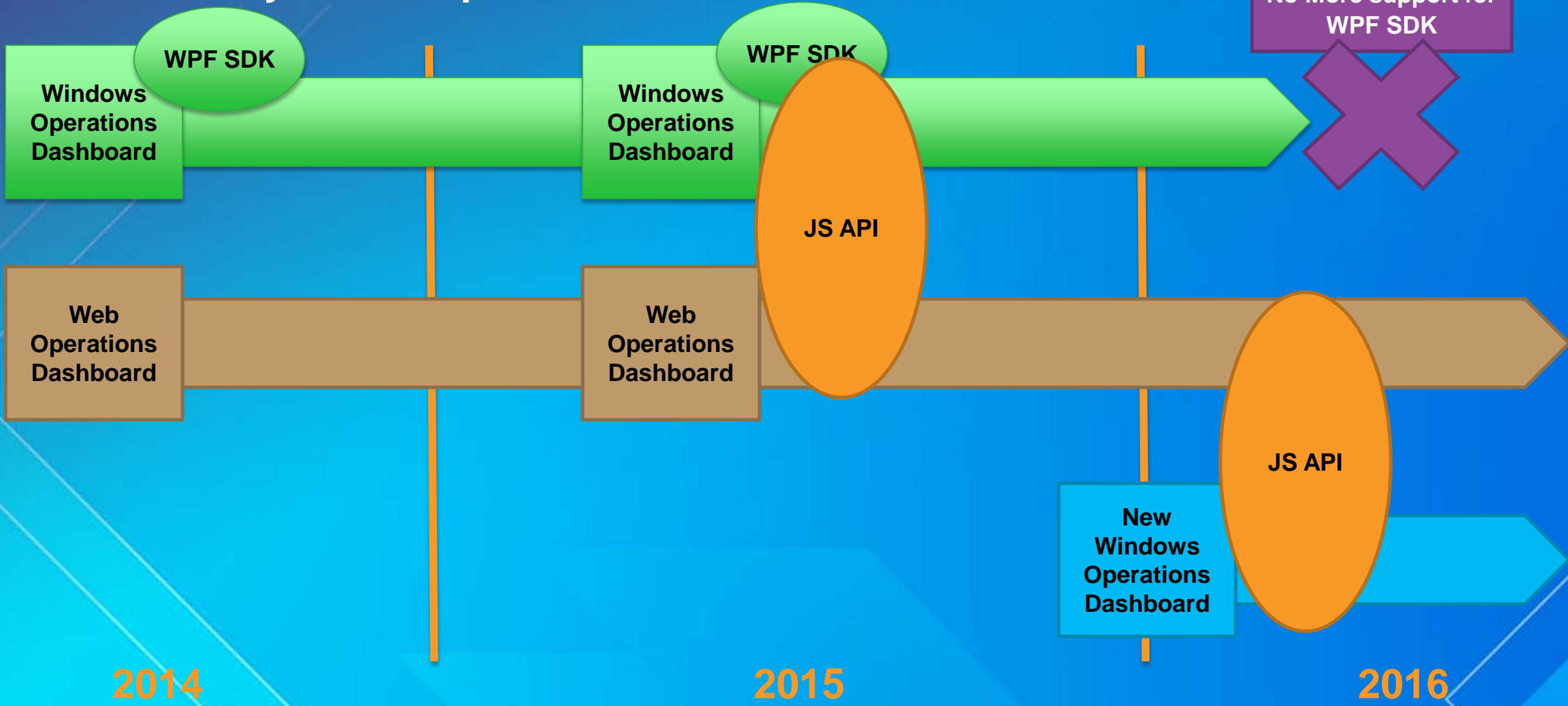


A screenshot of the 'Application' dialog box. The 'Type' section has 'Application Extension (Operations Dashboard)' selected. The 'Manifest URL' field contains '://energy.esri.com/dashboard-samples/listWidget/listWidget.json'. The 'Title' field contains 'My Super Duper List Widget'. The 'Tags' field contains 'extension' and 'Operations Dashboard'. A red circle highlights the 'Application Extension (Operations Dashboard)' option. Another red circle highlights the 'Manifest URL' field. A third red circle highlights the 'Tags' field with the text: 'This extension needs access to: - data from operation view'. The 'ADD ITEM' and 'CANCEL' buttons are at the bottom.



# Demo – register an extension

# Extensibility Roadmap



## Want to learn more?

- **ArcGIS API for JavaScript?**
  - <https://developers.arcgis.com/javascript/>
- **Doc?**
  - <https://developers.arcgis.com/javascript/jshelp/operations-dashboard-extensibility-overview.html>
- **Samples?**
  - <https://github.com/Esri/dashboard-samples>

# Upcoming Sessions for Operations Dashboard

- **WEDNESDAY**

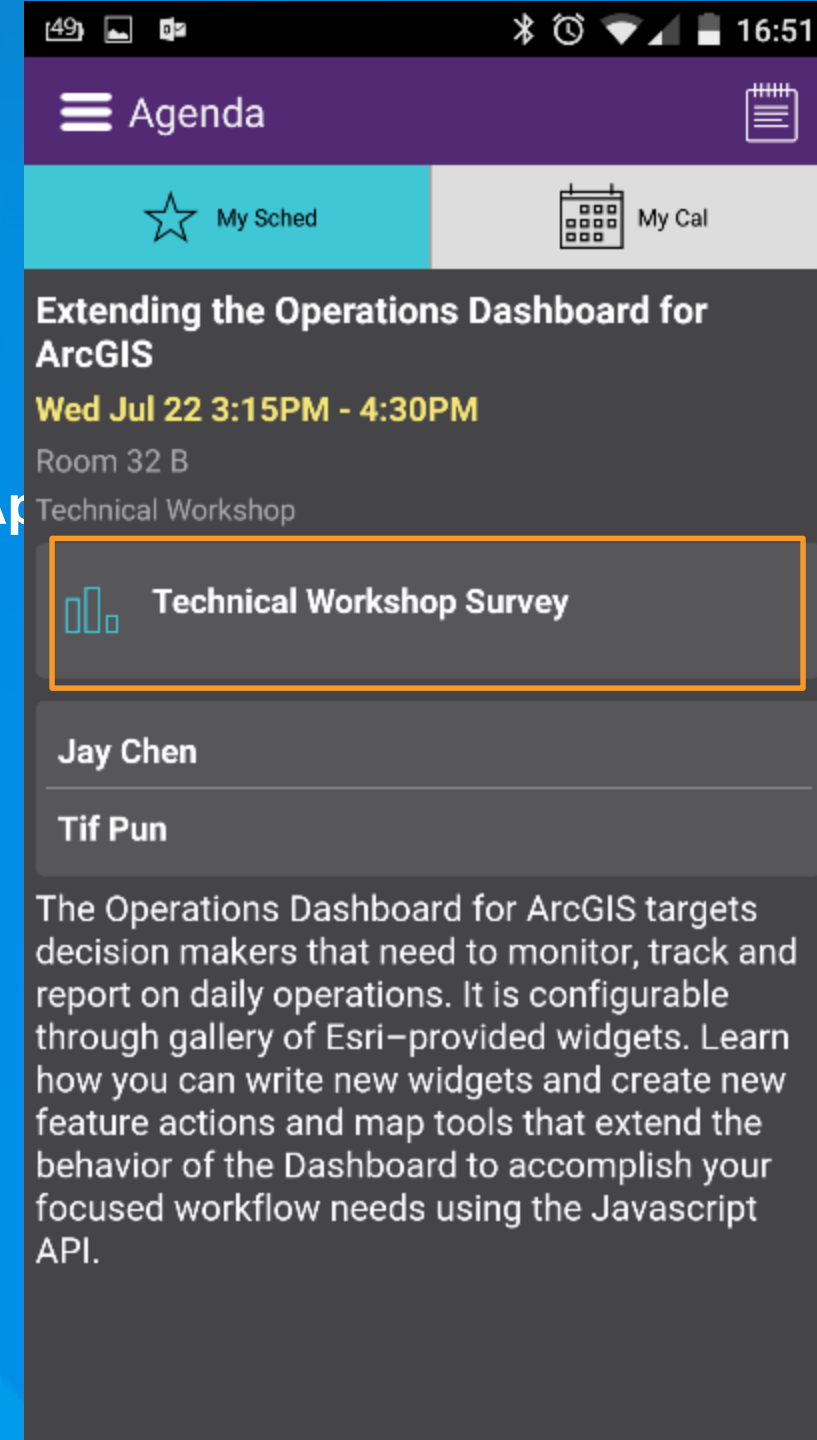
- 10:30 am – Demo Theater 10 – Extend Operations Dashboard with Custom Widgets
- 2:30 pm – Demo Theater 10 – Operations Dashboard: An Introduction
- 3:15 pm – Room 32 B – Extending Operations Dashboard (Tech Workshop)

- **THURSDAY**

- 3:15 pm – Room 32 A – Using Operations Dashboard (Tech Workshop)

# Thank you...

- Please fill out the session survey in your mobile app
- Select **Extending the Operations Dashboard** in the Mobile App
  - Use the Search Feature to quickly find this title
- Click “Technical Workshop Survey”
- Answer a few short questions and enter any comments







Understanding our world.