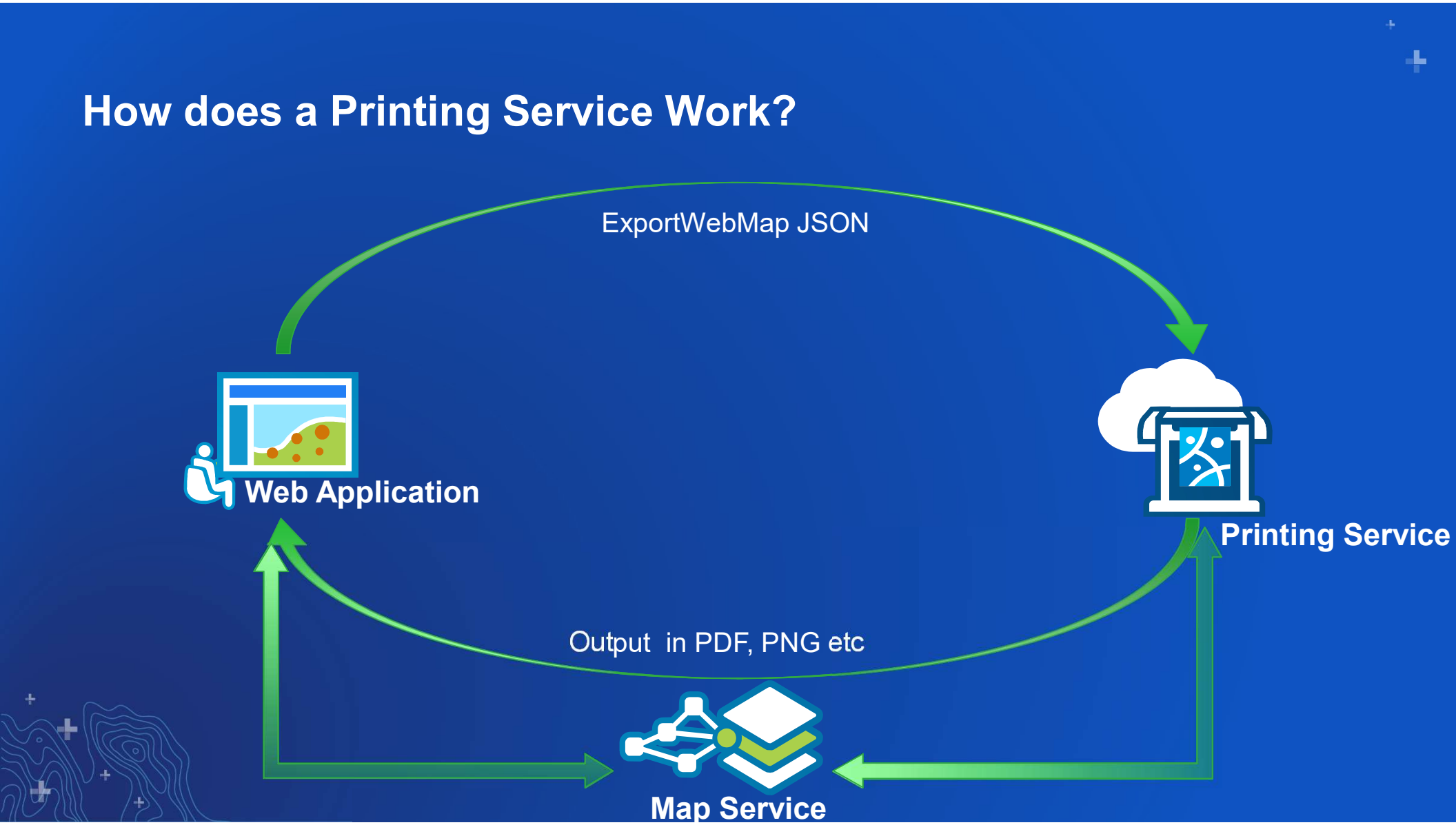
# Working with Print Service: Advanced Topics

Tanu Hoque and Jeff Moulds

SEE WHAT OTHERS CAN'T

# How does a Printing Service Work?



# Security

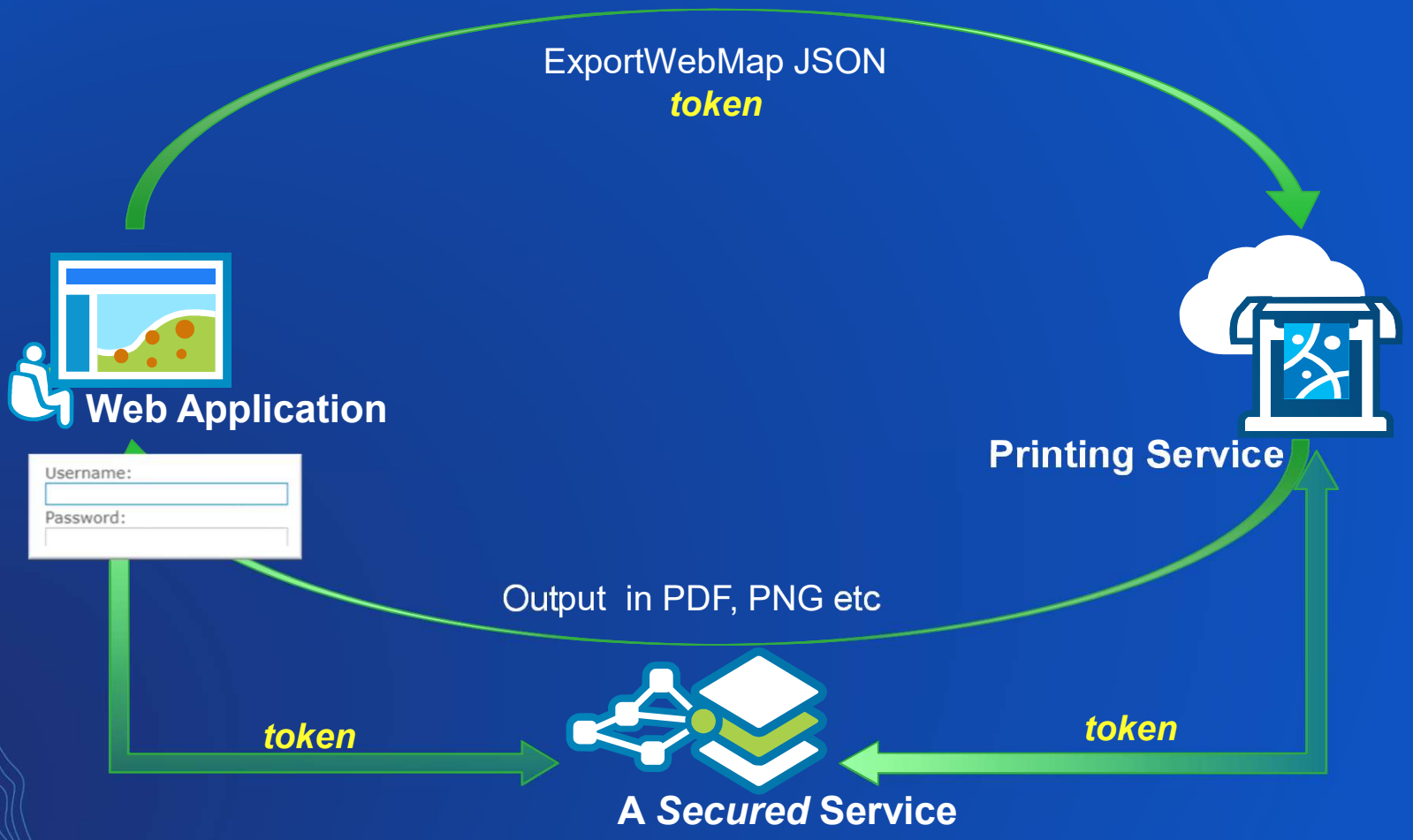
Print web map containing secured services

# Can I Use Secured Services with Printing Service?

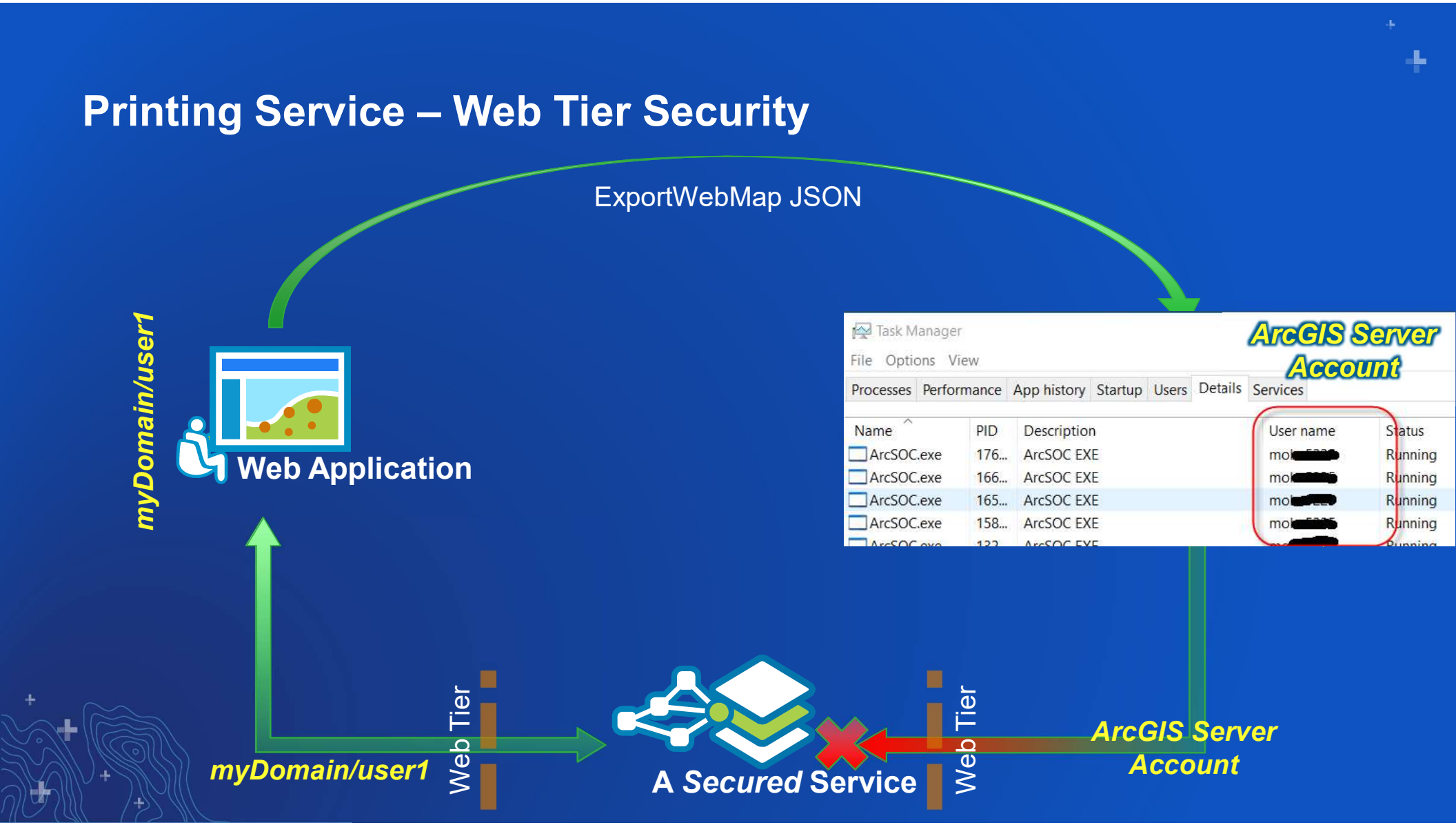
- Secured Services are supported
  - Short term **token** (optionally with **referrer**)
- **Web tier authentication** supported when
  - ArcGIS server is **federated** to ArcGIS Enterprise portal, and
  - only the **portal's ArcGIS Web Adaptor** uses web-tier authentication (IWA, PKI, Basic, and so on)
- Web tier authentication does **not** work, when
  - ArcGIS Server site's Web Adaptor uses web-tier authentication
- Embed credentials into a print service
  - Use this with cautions – not recommended!

**Help topic:** http://esriurl.com/15572

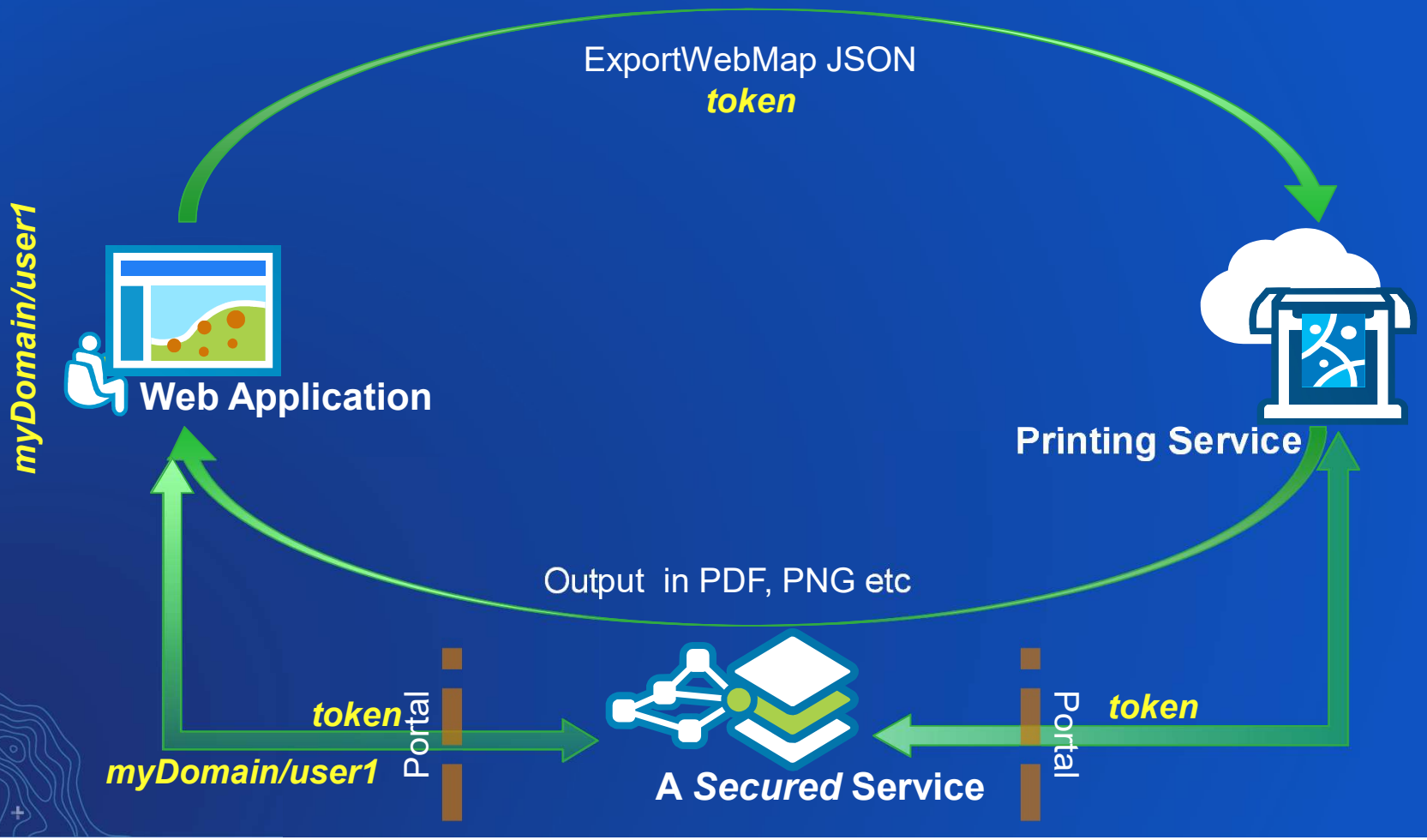
# Printing Service – how does it work with secured services?



# Printing Service – Web Tier Security



# Printing Service – Web Tier Security and Portal

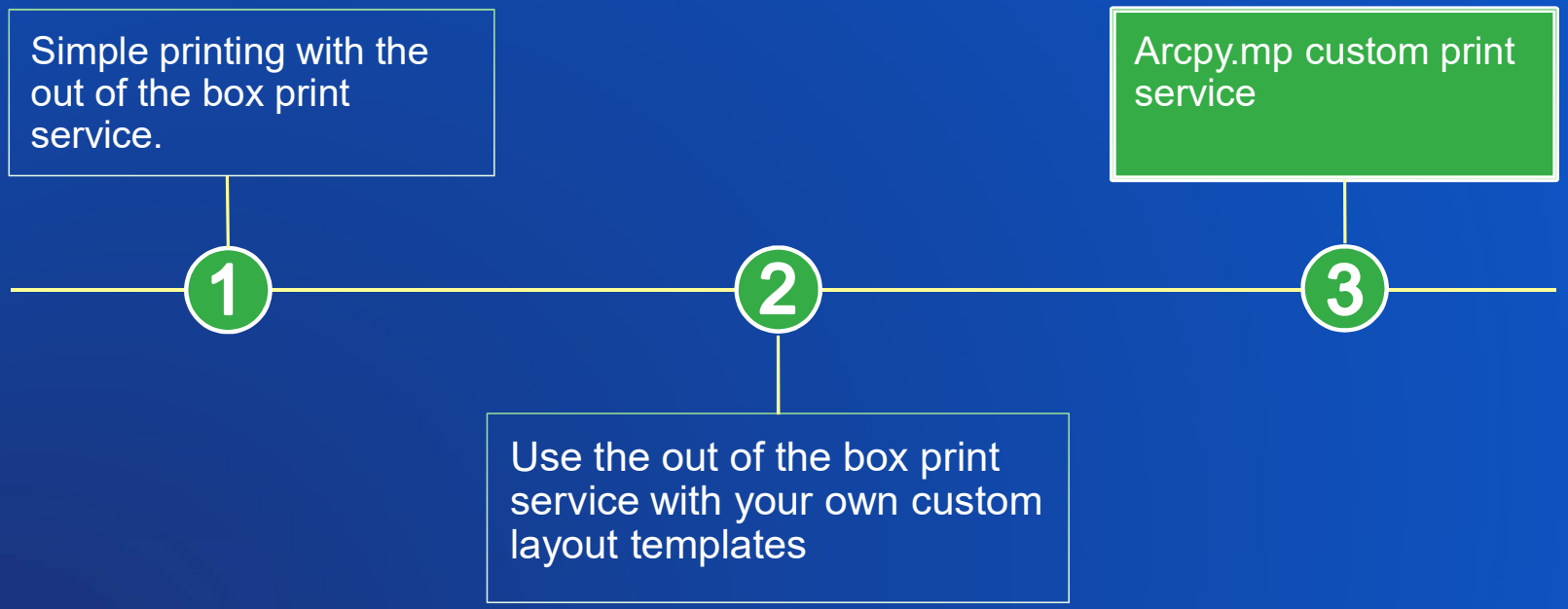


# Advanced Print Service Using arcpy

Jeff Moulds

# Three Web Map printing development paths

- **Create your web app using the JavaScript API or the Web AppBuilder**



# Advanced web map printing with arcpy.mp

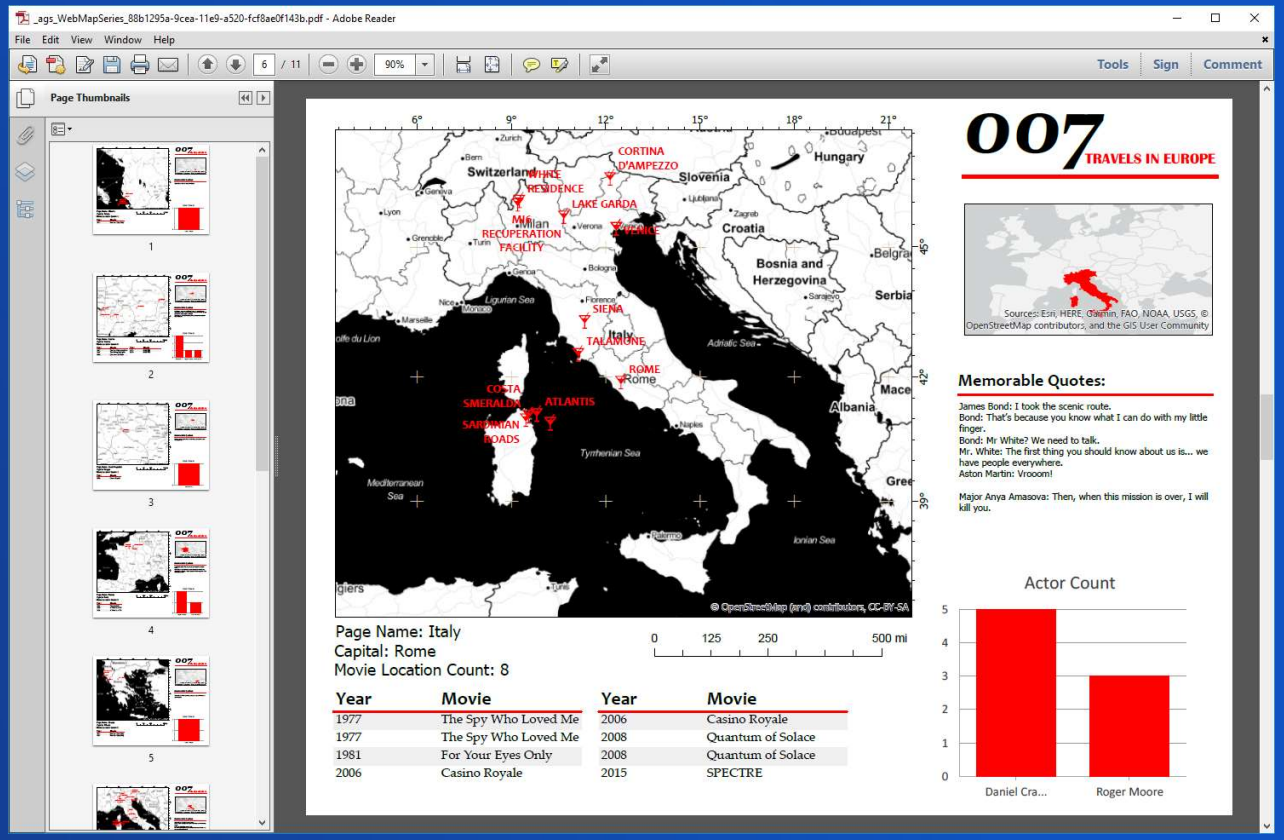
- Full capabilities of arcpy.mp:
  - Creating reports
  - Printing pop-ups (selected features) and feature attributes
  - Creating map books (map series)
  - Export to PDF and insert additional PDF pages (title page, reports, driving directions, street index, etc.)
  - Update layout elements (e.g. update a picture)
  - Export using advanced options (e.g. high DPI or embed feature attributes in PDF)
  - Swap out cached service layers for staged local vector data for vector PDF output
  - Etc.

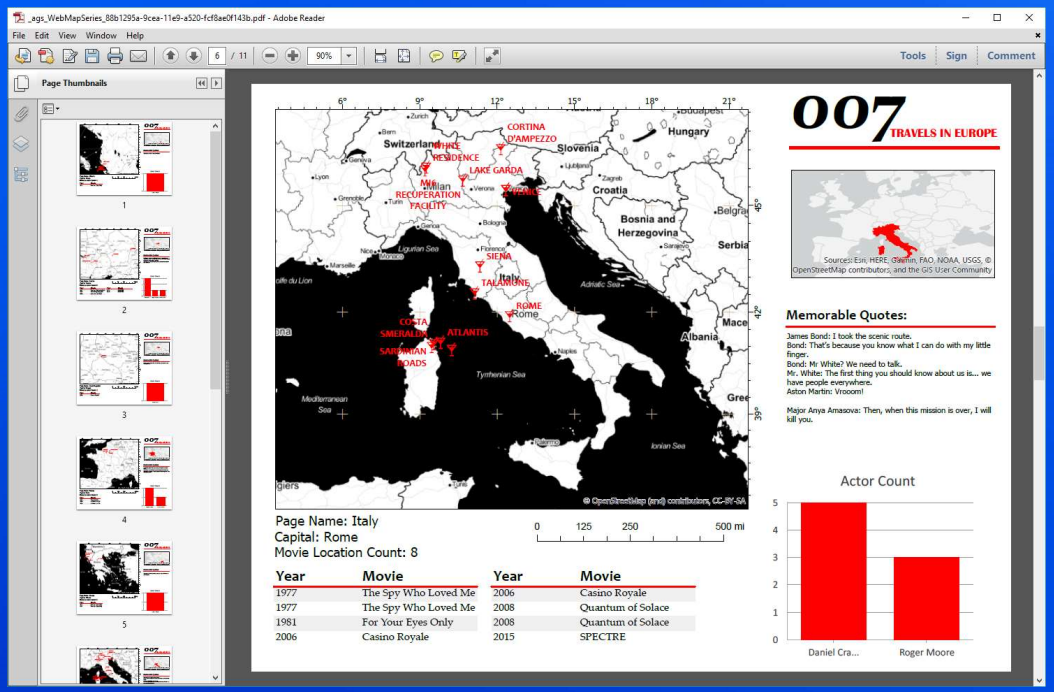
# Advanced web map printing with arcpy.mp


- The Print Widget in the web app will use your python script
- Arcpy.mp method for converting Web Maps to ArcGIS Pro Projects:
  - `ConvertWebMapToMapArcGISProject(webmap_json, {template_pagx}, {mapframe_name}, {notes_gdb})`
  - Return a printer-friendly output file (PDF, PNG, etc.)
  - Help topic - http://esriurl.com/14382

# Advanced web map printing with arcpy.mp - James Bond Demo

- arcpy.mp.MapSeries class
- Utilize dynamic layout elements:
  - Extent Indicators
  - Dynamic Grids
  - Dynamic Legends
  - Dynamic Tables
  - Dynamic Charts
  - Dynamic Text
    - Dynamic Table Attributes
    - Dynamic Table Statistics
    - Map Series Dynamic Text:
      - page X of Y
      - page name
      - page number






# James Bond Demo

Get web map JSON

```
# Input web map json
Web_Map_as_JSON = arcpy.GetParameterAsText(0)
```

Get output format

```
# Format for output
Format = arcpy.GetParameterAsText(1)
```

Get layout template

```
# Input Layout template
Layout_Template = arcpy.GetParameterAsText(2)

# The template location in the server data store
templatePath = 'C:/datastore/Bond'
templatePagx = os.path.join(templatePath, Layout_Template)
```

ConvertWebMapToArcGISProject

```
# Convert the web map to an ArcGIS Project
result = arcpy.mp.ConvertWebMapToArcGISProject(Web_Map_as_JSON, templatePagx, mapframe_name='Map Frame Extent')
aprx = result.ArcGISProject
```

Referencing variables

```
# Reference variables in the ArcGIS Project
layout = aprx.listLayouts('Layout')[0]
ms = layout.mapSeries
mfExtent = layout.listElements('MAPFRAME_ELEMENT', 'Map Frame Extent')[0]
mapSeriesLyr = mfExtent.map.listLayers("World Countries*")[0]
locationsExtentLyr = mfExtent.map.listLayers("Locations staged map series")[0]
```

Turn off web layers

```
# Turn off the service layer as the vector data is already staged in the pagx template file
mfExtent.map.listLayers("BondLocations")[0].visible = False
mfExtent.map.listLayers("Stamen")[0].visible = False
```

Select by location

```
# Select location points within the mapframe extent
extent = mfExtent.camera.getExtent().polygon
arcpy.SelectLayerByLocation_management(locationsExtentLyr, "INTERSECT", extent, selection_type="NEW_SELECTION")

# If there are no selected points, then dont export the map series.
if arcpy.Describe(locationsExtentLyr).FIDSet != '':
    # Select map series features that contain the selected location points
    arcpy.SelectLayerByLocation_management(mapSeriesLyr, "INTERSECT", locationsExtentLyr, selection_type="NEW_SELECTION")
    pageRange = ', '.join(str(e) for e in ms.selectedIndexFeatures)
    arcpy.AddMessage('Exporting the following page numbers: ' + pageRange)
```

Export to PDF

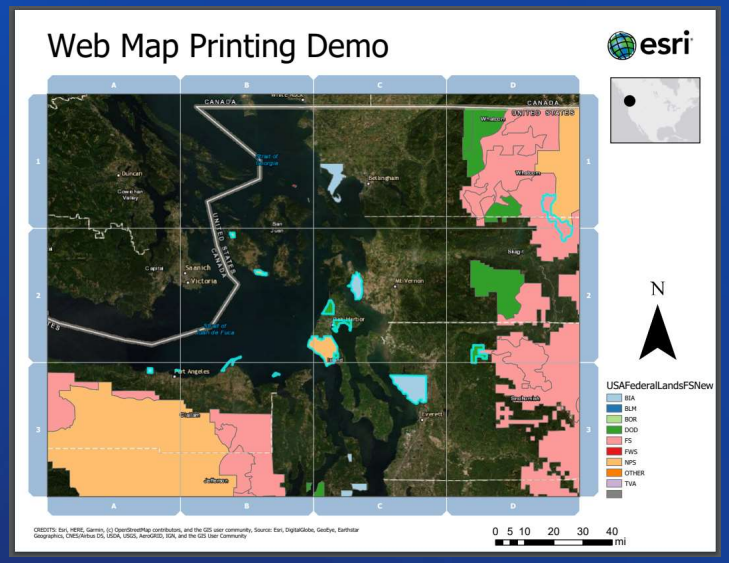
```
    # Define output PDF file name
    pdfMapSeries = os.path.join(arcpy.env.scratchFolder, 'WebMapSeries_{}.pdf'.format(str(uuid.uuid1())))

    # Export Map Series pages
    ms.exportToPDF(pdfMapSeries, 'RANGE', pageRange, layers_attributes='LAYERS_AND_ATTRIBUTES', georef_info=True)
```

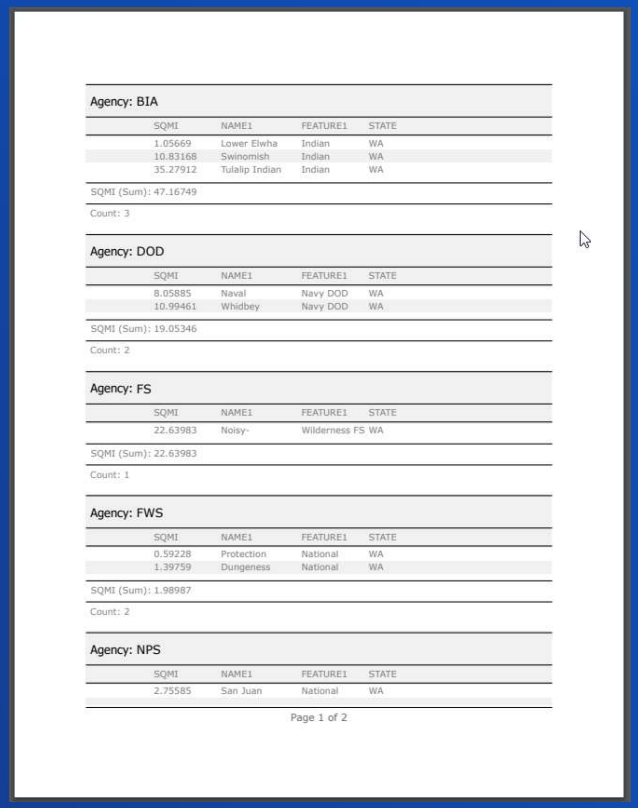
Output file for job

```
    # Set the output parameter to be the output file of the server job
    arcpy.SetParameterAsText(3, pdfMapSeries)
```

# Show selected features from Web App using Reports

- arcpy.mp.Report class
- aprx.importDocument(ReportTemplatePath)
  rpt = aprx.listReports("Fed Lands Report")[0]
  rpt.setReferenceDataSource(webLyr)
  rpt.exportToPDF(output_pdf)



Agency: BIA

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 1.05669 | Lower Elwha | Indian | WA |
| 10.83168 | Swinomish | Indian | WA |
| 35.27912 | Tulalip Indian | Indian | WA |

SQMI (Sum): 47.16749

Count: 3

Agency: DOD

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 8.05885 | Naval | Navy DOD | WA |
| 10.99461 | Whidbey | Navy DOD | WA |

SQMI (Sum): 19.05346

Count: 2

Agency: FS

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 22.63983 | Noisy- | Wilderness FS | WA |

SQMI (Sum): 22.63983

Count: 1

Agency: FWS

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 0.59228 | Protection | National | WA |
| 1.39759 | Dungeness | National | WA |

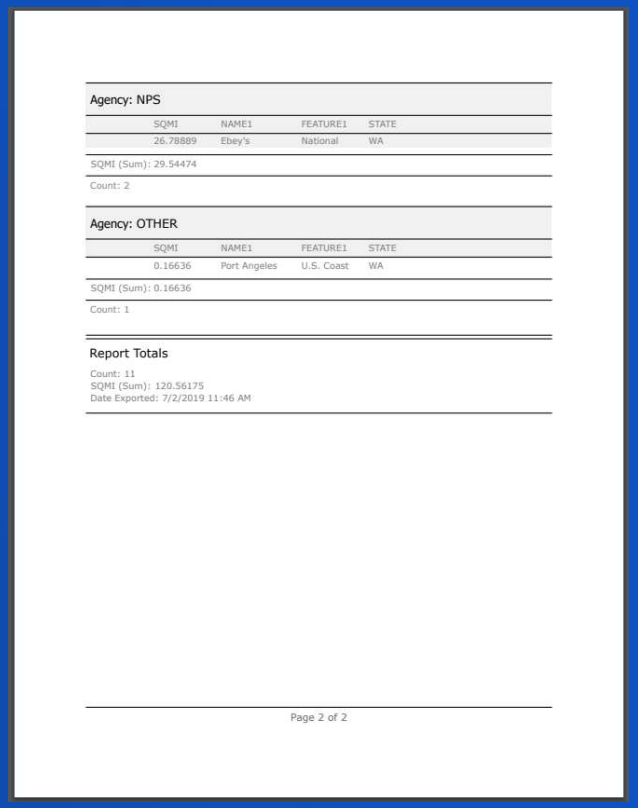
SQMI (Sum): 1.98987

Count: 2

Agency: NPS

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 2.75585 | San Juan | National | WA |

Page 1 of 2



Agency: NPS

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 26.78889 | Ebey's | National | WA |

SQMI (Sum): 29.54474

Count: 2

Agency: OTHER

| SQMI | NAME1 | FEATURE1 | STATE |
|---|---|---|---|
| 0.16636 | Port Angeles | U.S. Coast | WA |

SQMI (Sum): 0.16636

Count: 1

Report Totals

Count: 11
SQMI (Sum): 120.56175
Date Exported: 7/2/2019 11:46 AM

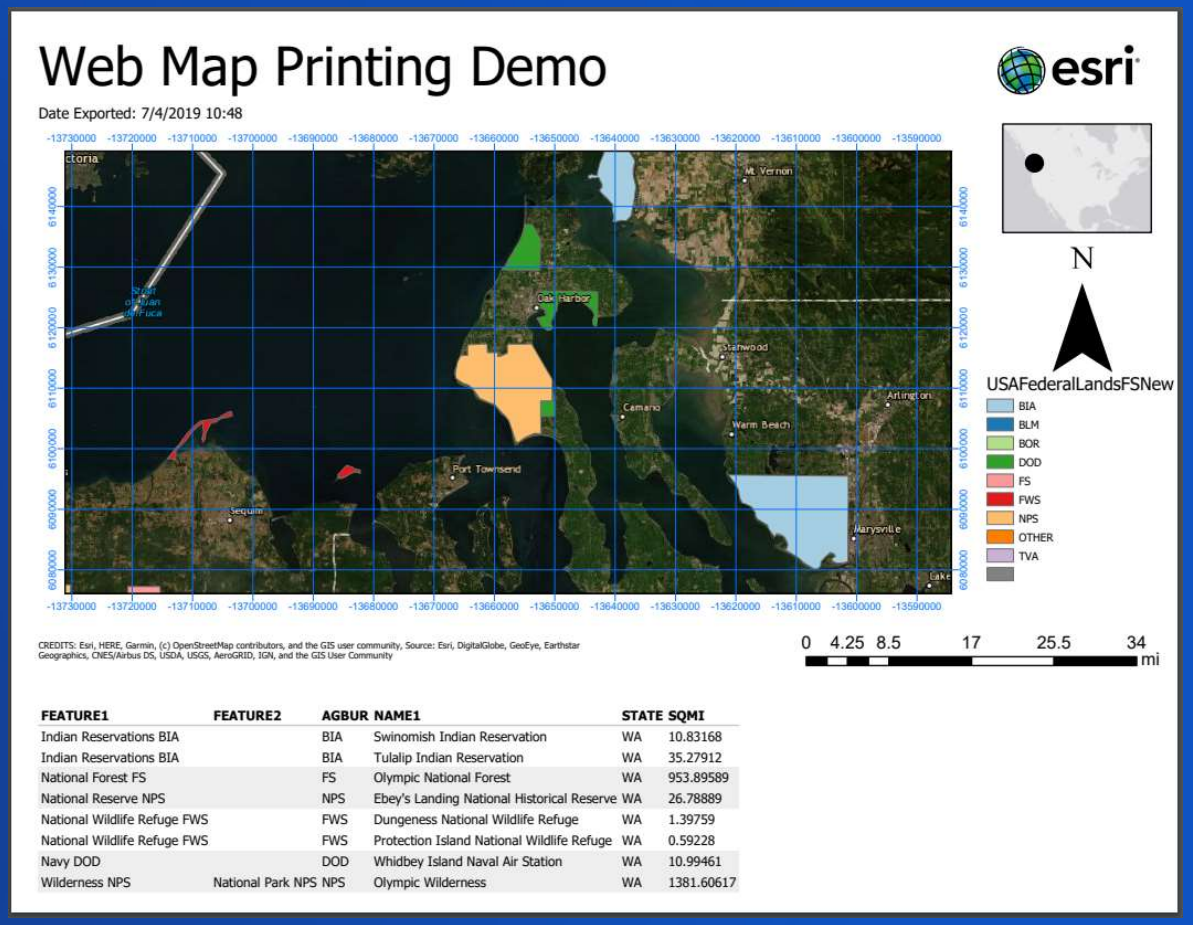
Page 2 of 2

- Blog post: http://esriurl.com/15573
- See code sample #7 in the ConvertWebMapToArcGISProject help topic: http://esriurl.com/15537

# Show selected features from web app – Table Frame Demo

- Use the CIM – new in ArcGIS Pro 2.4

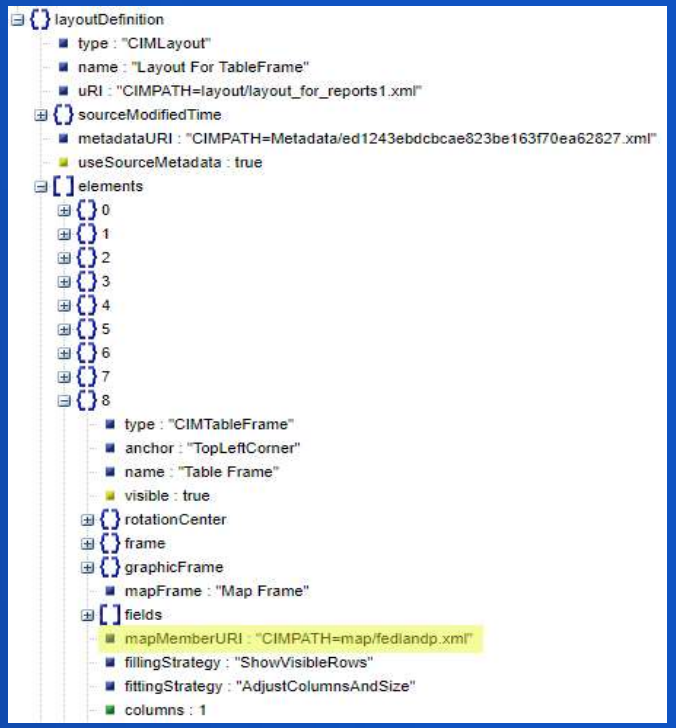
```
layCIM = layout.getDefinition("V2")
for elm in layCIM.elements:
    if elm.name == 'Table Frame':
        elm.mapMemberURI = lyrCIM.uRI
        layout.setDefinition(layCIM)
```




| FEATURE1 | FEATURE2 | AGBUR | NAME1 | STATE | SQMI |
|---|---|---|---|---|---|
| Indian Reservations BIA | | BIA | Swinomish Indian Reservation | WA | 10.83168 |
| Indian Reservations BIA | | BIA | Tulalip Indian Reservation | WA | 35.27912 |
| National Forest FS | | FS | Olympic National Forest | WA | 953.89589 |
| National Reserve NPS | | NPS | Ebey's Landing National Historical Reserve | WA | 26.78889 |
| National Wildlife Refuge FWS | | FWS | Dungeness National Wildlife Refuge | WA | 1.39759 |
| National Wildlife Refuge FWS | | FWS | Protection Island National Wildlife Refuge | WA | 0.59228 |
| Navy DOD | | DOD | Whidbey Island Naval Air Station | WA | 10.99461 |
| Wilderness NPS | National Park NPS | NPS | Olympic Wilderness | WA | 1381.60617 |

# Python CIM Access in Pro (released with Pro 2.4)

- CIM (Cartographic Information Model)
  - Specification for how project and document information is persisted and re-created
  - MAPX, PAGX, LYRX files are in JSON format
  - Structure is analogous to an object model diagram
- Arcpy.mp can navigate the CIM object model
  - Entry points are with the Map, Layer, Table or Layout objects
- Don't let the name fool you
  – you have access to so much more than cartography!




```
lyrCIM = newLyr.getDefinition("V2")
layCIM = layout.getDefinition("V2")
for elm in layCIM.elements:
    if elm.name == 'Table Frame':
        elm.mapMemberURI = lyrCIM.uRI
layout.setDefinition(layCIM)
```

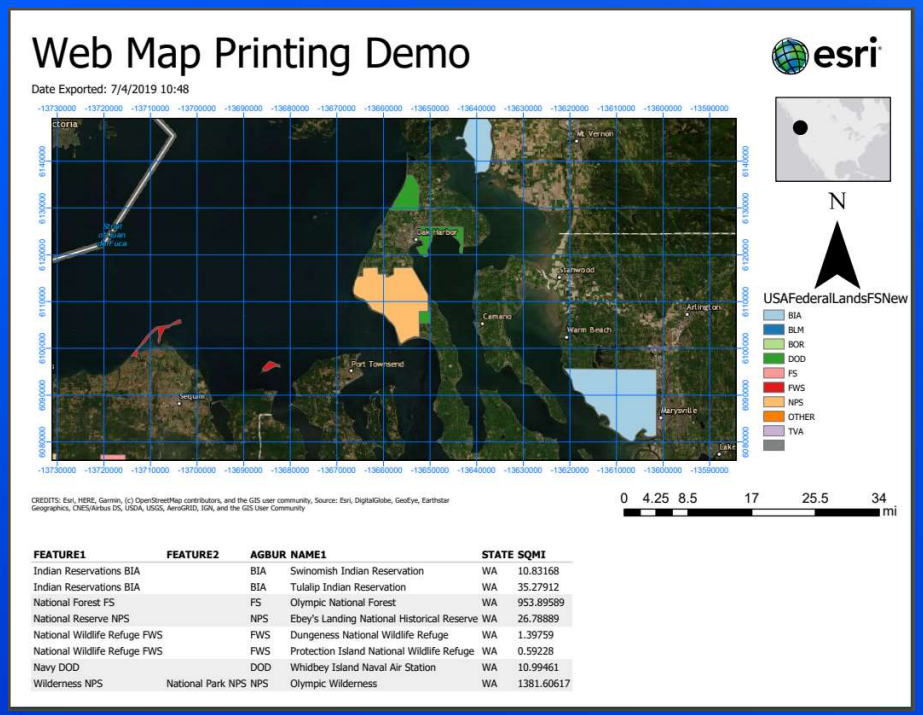
**Help topic:** http://esriurl.com/15489

**Video:** http://esriurl.com/15490

# Why CIM Access … and why not

- Provide finer grained access to project properties
- Pro is growing so rapidly, the CIM immediately exposes capabilities
- It has been available to the .NET SDK community since the beginning

- Why not to use the CIM
  - Doesn't provide access to everything
    - high level project properties, metadata, change spatial reference
  - Can't create new objects.  If it is not in the CIM, you can't access it.
  - No life guard (managed UI or API) on duty
    - You can break the behavior of the app
    - Test thoroughly




| FEATURE1 | FEATURE2 | AGBUR | NAME1 | STATE | SQMI |
|---|---|---|---|---|---|
| Indian Reservations BIA | | BIA | Swinomish Indian Reservation | WA | 10.83168 |
| Indian Reservations BIA | | BIA | Tulalip Indian Reservation | WA | 35.27912 |
| National Forest FS | | FS | Olympic National Forest | WA | 953.89589 |
| National Reserve NPS | | NPS | Ebey's Landing National Historical Reserve | WA | 26.78889 |
| National Wildlife Refuge FWS | | FWS | Dungeness National Wildlife Refuge | WA | 1.39759 |
| National Wildlife Refuge FWS | | FWS | Protection Island National Wildlife Refuge | WA | 0.59228 |
| Navy DOD | | DOD | Whidbey Island Naval Air Station | WA | 10.99461 |
| Wilderness NPS | National Park NPS | NPS | Olympic Wilderness | WA | 1381.60617 |

# Demo: Python CIM access to show features from a Web App in a Table Frame.

Get web map JSON

Get output format

Get layout template

Convert Web Map to ArcGIS Project

Referencing variables

Remove staged layer

Reference layer from web map

Get CIM definitions

Find Table Frame in CIM

Modify Table Frame in CIM & commit changes

Output file for job

```
# Input web map json
Web_Map_as_JSON = arcpy.GetParameterAsText(0)
```

```
# Format for output
Format = arcpy.GetParameterAsText(1)
```

```
# Input Layout template
Layout_Template = arcpy.GetParameterAsText(2)

# The template location in the server data store
basedir = 'C:/datastore/SelectedFeatures/SelectedFeaturesProject'
templatePagx = os.path.join(basedir, Layout_Template)
```

```
# Convert the web map to an ArcGIS Project
result = arcpy.mp.ConvertWebMapToArcGISProject(Web_Map_as_JSON, templatePagx, 'Map Frame')
aprx = result.ArcGISProject
```

```
# Reference variables in the ArcGIS Project
layout = aprx.listLayouts()[0]
m = aprx.listMaps('Web Map')[0]
```

```
# Remove the layer in the layout template that was used to create the table
m.removeLayer(m.listLayers("SeedLayerForFields")[0])
```

```
# Reference the first layer from the web map json
lyr = m.listLayers()[1]
f = arcpy.ListFields(lyr)
```

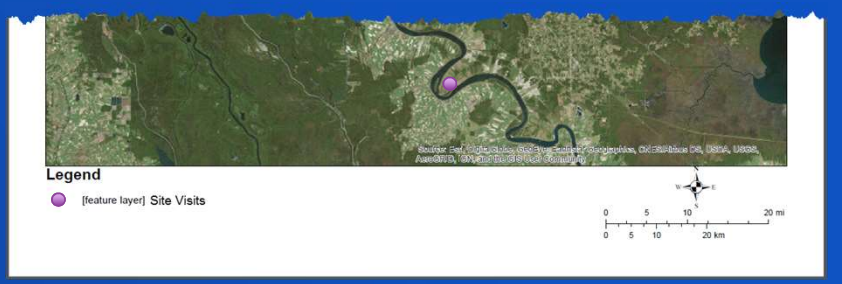
```
# Get CIM definition for the web map layer and the layout
lyrCIM = lyr.getDefinition("V2")
layCIM = layout.getDefinition("V2")
```

```
# Get the table frame from the CIM
for elm in layCIM.elements:
    if elm.name == 'Table Frame':
        # Point the Table Frame to the layer from the Web Map
        elm.mapMemberURI = lyrCIM.uRI
        # Use the field names from the layer from the Web Map in the Table Frame
        for x in range(len(elm.fields) - 1):
            elm.fields[x].name = f[x + 1].name
        # Set Layout CIM definition
        layout.setDefinition(layCIM)
```

```
# Export the layout to PDF and set the output parameter to be the output PDF
LayoutPDF = os.path.join(arcpy.env.scratchFolder, 'WebMapLayout_{}.pdf'.format(str(uuid.uuid1())))
layout.exportToPDF(LayoutPDF)
arcpy.SetParameterAsText(3, LayoutPDF)
```

# Advanced web map printing with arcpy.mp

- Printing geodatabase attachments (e.g. pictures)




```
import requests, json

# Use the json module to get the selected features
# from the web map json
js = json.loads(Web_Map_as_JSON)
```

```
# send a url request to retrieve the attachments
attchInfosUrl = '{0}/{1}/attachments?f=json'.format(url, oid)
r = requests.get(attchInfosUrl)
j = r.json()
attchInfos = j['attachmentInfos']
```

```
# reference empty staged picture element on the layout
picElm = layout.listElements("PICTURE_ELEMENT")[0]
```

```
# update the picture's source image with the url of the attachment
picElm.sourceImage = '{0}/{1}/attachments/{2}'.format(url, oid, attchInfos['id'])
```

**Community Site Visit**

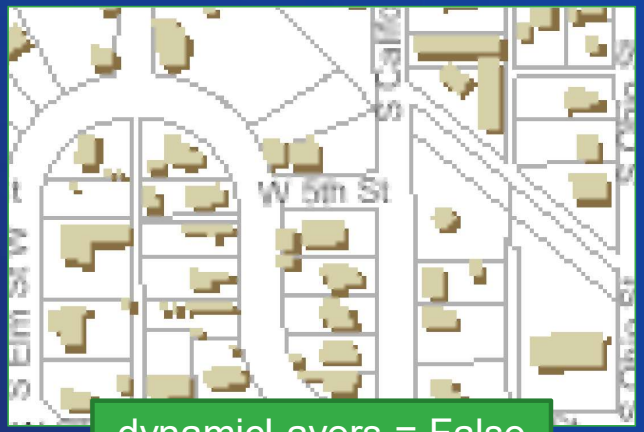
| | |
|---|---|
| **ID Number:** | 65489798654 |
| **Specialist:** | 13 |
| **Address:** | 2846 East 18th Ave |
| **City:** | Victoria |
| **Development Type:** | New Construction |
| **Structure Type:** | Mobile Home |
| **Machine Elevated:** | No |
| **Enclosure:** | N/A |
| **Comments:** | N/A |

# Tips

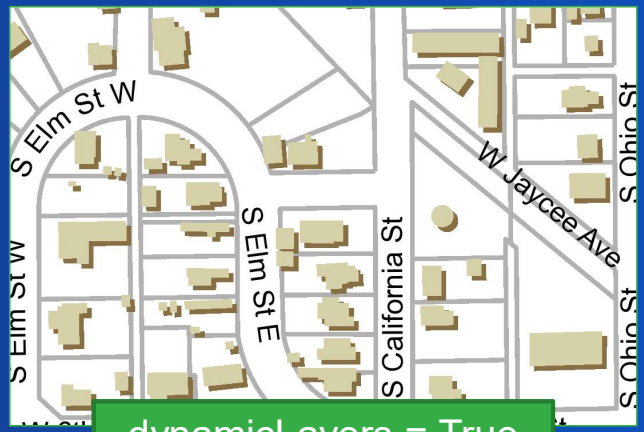
Tanu Hoque

# Printing Cached Services

- ***Raster tiles typically created at 96 DPI***
- Dynamic layer support can be enabled for Map Services
  - Allows for printing at higher DPI than cached images
  - You must have Publisher or Admin privileges to update the service



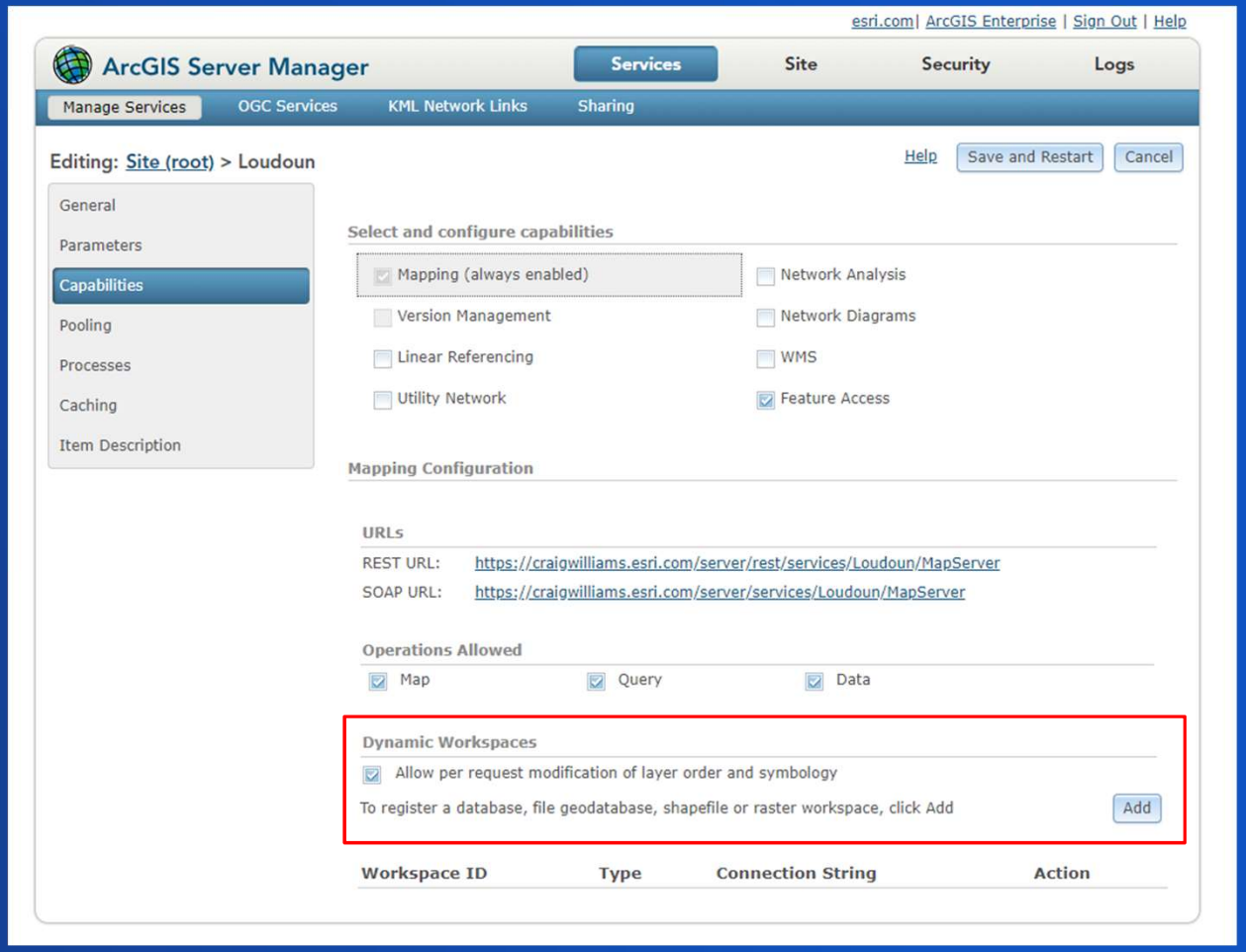

dynamicLayers = False
(600 DPI)




dynamicLayers = True
(600 DPI)

# Enabling Dynamic Layers

Option in properties of a cached service



# Recap

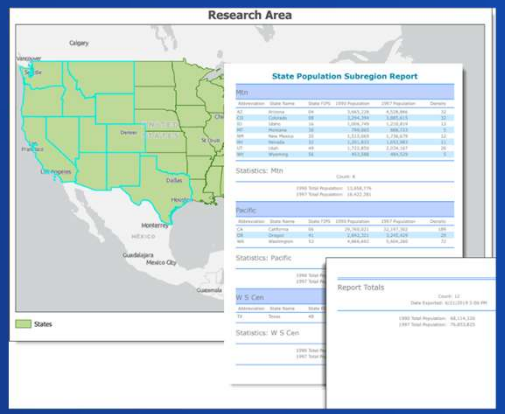
- 3 tier approach for on-premises print service
  - Out of the box
  - Your own print service with custom templates
  - Arcpy based advanced print service
- WebMap_As_JSON
- Print service must have access to all services
- Arcpy to solve advanced use cases:
  - Reports
  - Map series
  - Tables, Charts etc.
- Security

# Advanced web map printing with arcpy.mp – additional resources

- Introduction to arcpy.mp - http://esriurl.com/14447
- Make a layout tutorial - http://esriurl.com/14457

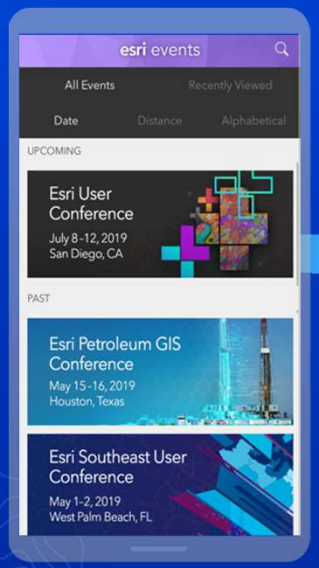
# Help links

- Printing in web application http://esriurl.com/15563
- Share your arcpy script as web tool http://esriurl.com/15564
- Print maps that contain secured services http://esriurl.com/15572
- Blog post on print reports http://esriurl.com/15561

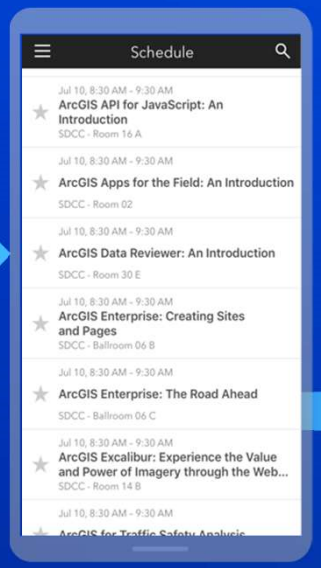


# Please Share Your Feedback in the App

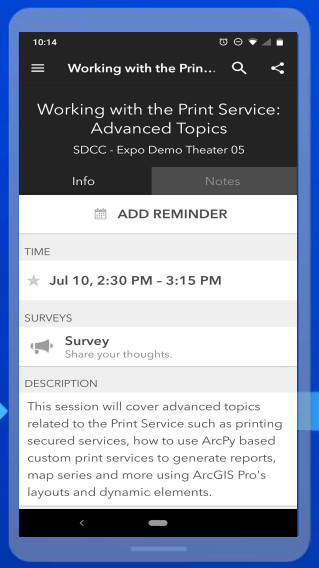
Download the Esri Events app and find your event



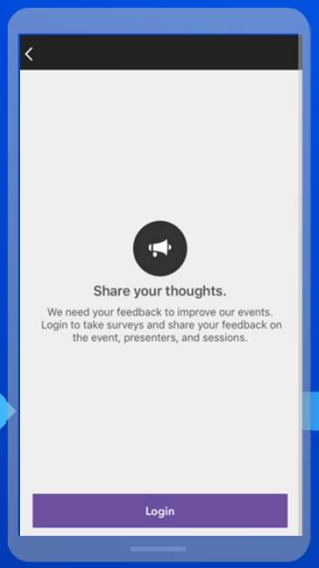

Select the session you attended




Scroll down to "Survey"




Log in to access the survey




Complete the survey and select "Submit"