ArcGIS Enterprise: Using Webhooks

Cédric Despierre Corporon

Matthias Schenker

ESRI EUROPEAN DEVELOPER SUMMIT

esri

Agenda

- What are webhooks?
  - Advantages of implementing
- Key terms
- Demo
- Additional use cases
- Administration Tips
- Resources

Introducing webhooks

Automate notifications based on portal actions

An action occurs.

- Something is published
- A user creates a new account
- An item is deleted

The webhook is triggered.

- Delivers JSON description of event
- Relevant info fields


```
  "info": {
    "webhookName": "Group monitoring",
    "webhookId": "72fed926aeb74c9ca8a22aacddc6725a",
    "portalURL": "https://machineURL/portal/",
    "when": 1543192196521
  },
  "events": [{
    "username": "administrator",
    "userId": "173dd04b69134bdf99c5000aad0b6298",
    "when": 1543192196521,
    "operation": "update",
    "source": "group",
    "id": "173dd04b69134bdf99c5000aad0b6298",
    "properties": {}
  }]
}
```


Your platform of choice kicks in.

- IFTTT
- Integromat
- Microsoft Flow

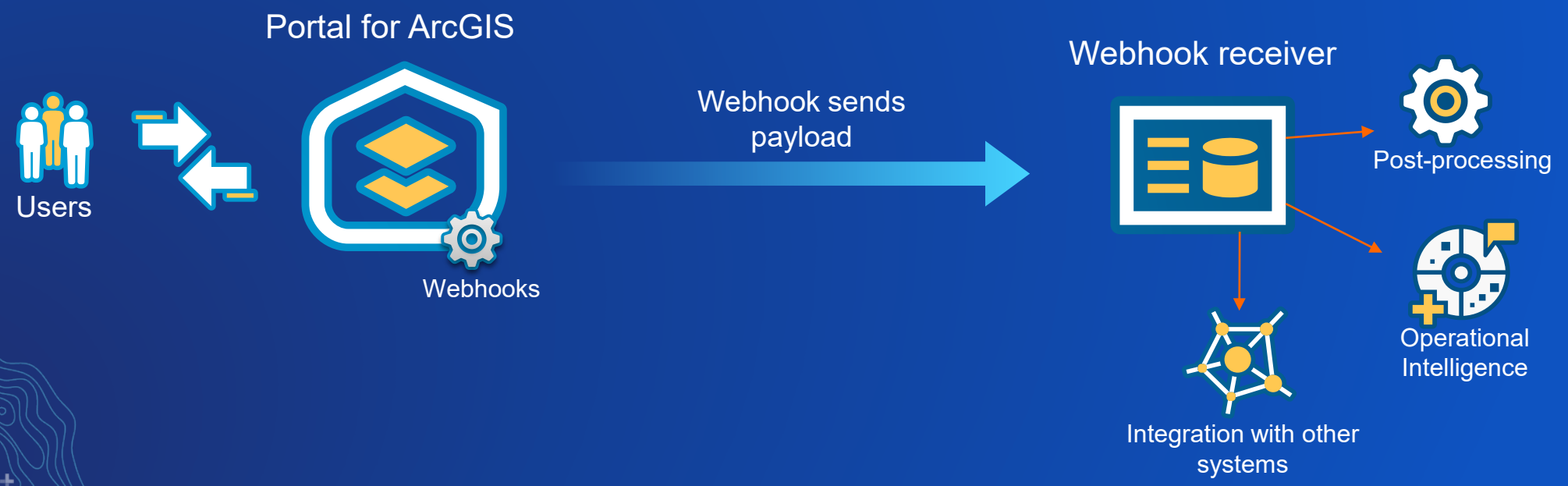
It performs an action.

- Emails you
- Posts in Slack
- Alexa announces it at the dinner table (maybe not)

!

What is a webhook?

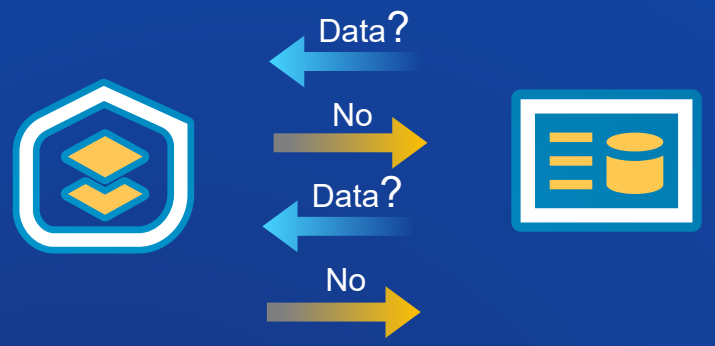
- A webhook is a new ArcGIS Enterprise capability that will automatically provides other applications with event-driven information, delivered as an HTTPS request (POST)
- They can be used to create automated and integrative workflows, adding new extensibility to ArcGIS Enterprise.



Advantages of using webhooks

- **More efficient than polling**

Polling:



Webhook:

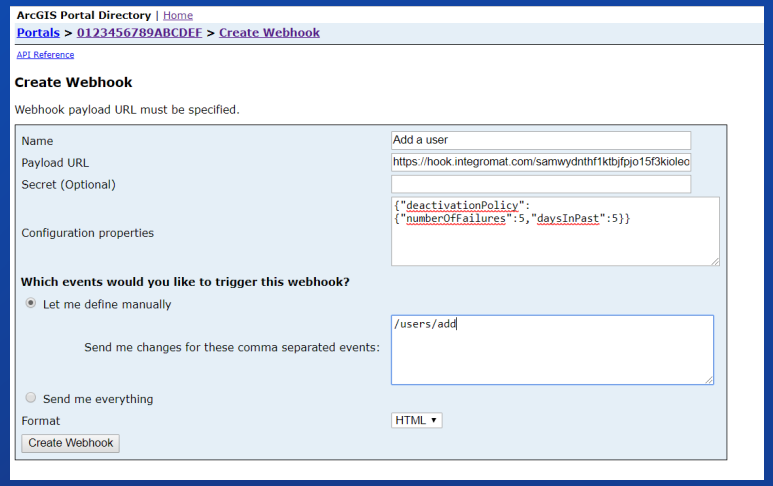
- **Allow users to create custom event-driven workflows, that can be integrated across multiple systems.**
- **Creates new and more efficient opportunities for automation.**

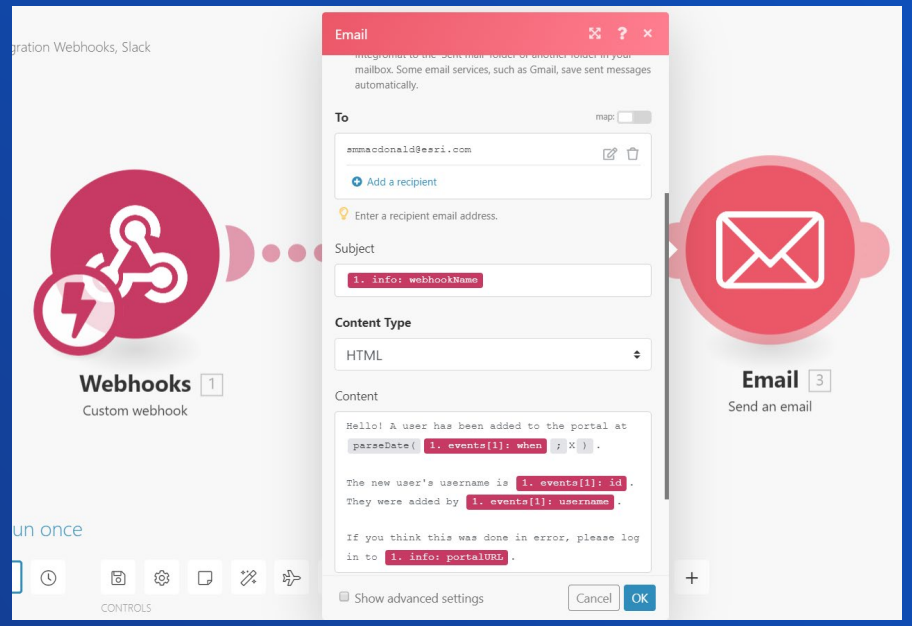
Scope

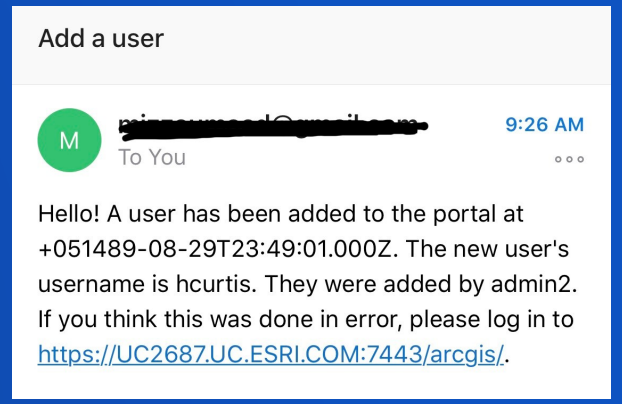
- Webhooks can only be created, updated, and deleted by portal admins via the ArcGIS Portal directory (Sharing API)
  - Registration and management of webhooks are also supported via the ArcGIS API for Python.

- Webhooks can be created to subscribe to events pertaining to items, users and groups in Portal for ArcGIS.
  - For 10.7, these events are primarily *create, update, and delete* –type events.
  - The REST API doc was updated to document this resource

How to: create webhooks







1

- Determine which client to use
- Set up webhook in Portal API
- Add payload URL from client

2

- Build webhook sequence
- Include variables from webhook JSON message

3

- Turn sequence on
- Receive messages each time a user is added

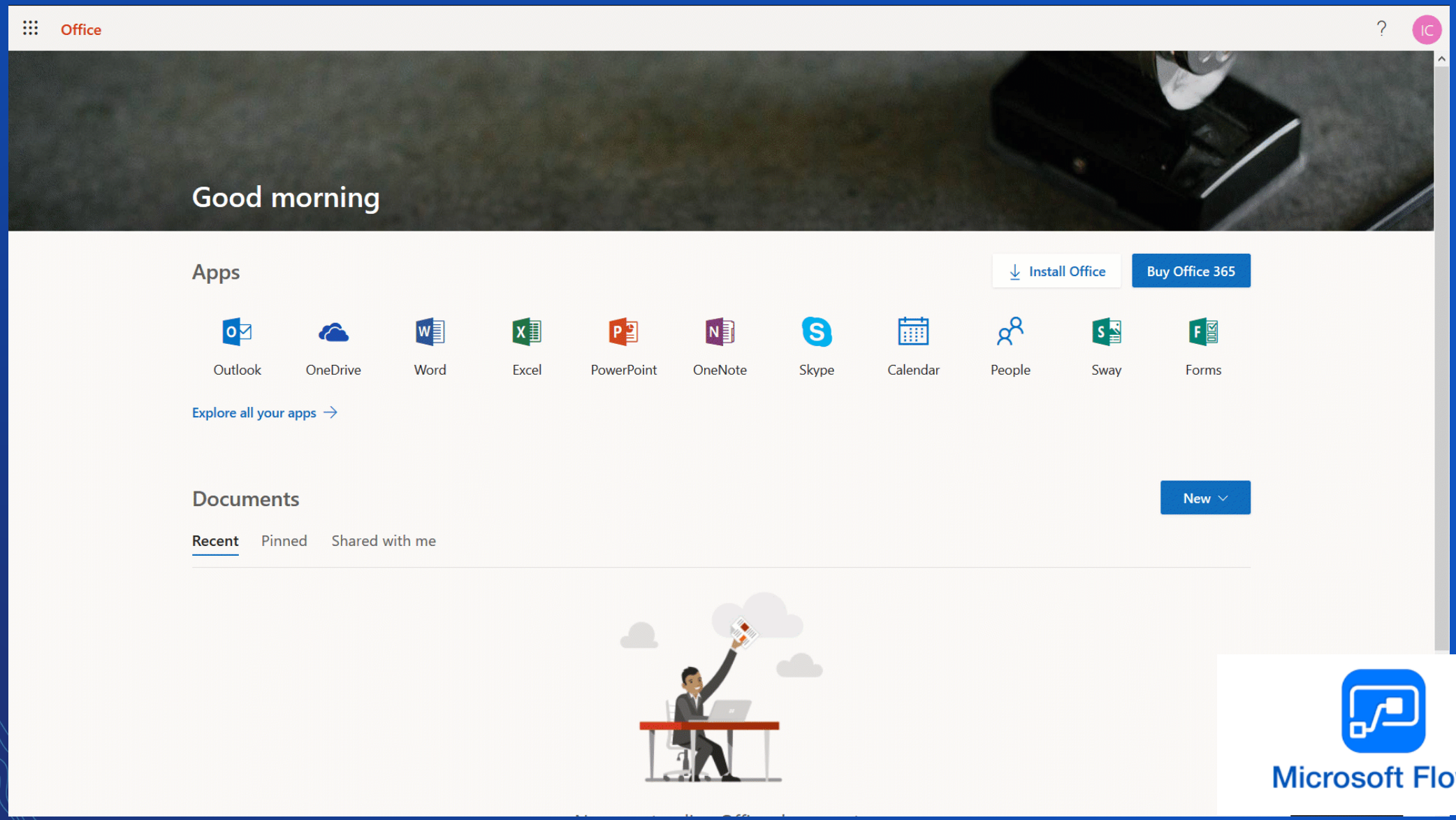
3rd Party Workflow Automation Software

- Visually create workflows that integrate ArcGIS Enterprise with other apps.
- Leverage the out of the box integrations for:

Integromat
Automate your workflow

Microsoft Flow

Microsoft Flow is part of Office 365



Microsoft Flow

Flow

When a survey response is submitted -> Send email

Save | Flow Checker | Test

- Home
- Approvals
- My flows
- Templates
- Connectors
- Data
- Solutions

When a survey response is submitted (Preview)

Send email

Choose an action

Search connectors and actions

All | Built-in | Standard | Premium | Custom

Control | Notifications | Office 365 Outlook | OneDrive for Business | SharePoint | Twitter | 10to8 Appointment

Triggers | Actions

Microsoft Flow

Automating e-mail notifications

A quick demonstration with ArcGIS Enterprise
and Microsoft Flow

Webhooks

3 example use cases

- **/groups:** When someone shares an item to a collaboration group, notify group members through Slack.
- **/items:** If the sharing settings are changed for a layer that contains confidential information, email administrators.
- **/users:** Once a user is deleted, notify your admin via text.

Key Terms: Trigger event

- **This is the operation you set to trigger your webhook. For example, you can configure your webhook to be triggered when a specific item is updated in your organization, or when an item is shared. A webhook can have more than one trigger event.**

Trigger event	URI example
All trigger events for all items	/items
Add item to the portal	/items/add
All trigger events for a specific item	/items/<itemID>
Delete a specific item	/items/<itemID>/delete
Update a specific item's properties	/items/<itemID>/update
Move an item or changing ownership of the item	/items/<itemID>/move
Publish a specific item	/items/<itemID>/publish
Share a specific item	/items/<itemID>/share
Unshare a specific item	/items/<itemID>/unshare

Key Terms: Trigger event

Trigger events for Portal users:

Trigger event	URI example
All trigger events for all users in the portal	/users
Add user to the org	/users/add
All trigger events associated with a specific user	/users/<username>
Delete a specific user	/users/<username>/delete
Update a specific user's profile	/users/<username>/update
Disable a specific user's account	/users/<username>/disable
Enable a specific user's account	/users/<username>/enable
Specific user signs in	/users/<username>/signIn
Specific user signs out	/users/<username>/signOut

Trigger events for Portal groups:

Trigger event	URI example
All trigger events for all groups	/groups
Add group	/groups/add
All trigger events for a specific group	/groups/<groupID>
Update a specific group	/groups/<groupID>/update
Delete a specific group	/groups/<groupID>/delete
Enable **Delete Protection** for a specific group	/groups/<groupID>/protect
Disable **Delete Protection** for a specific group	/groups/<groupID>/unprotect
Invite a user to a specific group	/groups/<groupID>/invite
Add a user to a specific group	/groups/<groupID>/addUsers

Key Terms: Payload

- Once a webhook is triggered, a payload is delivered to the specified payload URL in JSON format. Each event follows a similar JSON schema with information that is relevant to the event


```
{
  "info" : {
        "webhookId" : "b1280bce338e4a96a981b09c50ee409c",
        "webhookName" : "Myhook",
        "portalURL" : "https://dev004029.esri.com/portal/",
        "when" : 1547056298561
  },
  "events" : [{
              "userId" : "4ee8b3d66a2e4da3a56f8d256f1df7ce",
              "username" : "admin",
              "when" : 1547056298554,
              "operation" : "share",
              "source" : "item",
              "id" : "186bad9e3dd847ab9a601b505879f86d",
              "properties" : {
                    "sharedToGroups" : [“Everyone”, “4adc30bb03054812a846fa592de105de”]
}}]}
```

```
{
 "info" : {
     "webhookId" : "b1280bce338e4a96a981b09c50ee409c",
     "webhookName" : "Myhook",
     "portalURL" : "https://FQDNm/portal/",
     "when" : 1547056298561
 },
 "events" : [{
         "userId" : "4ee8b3d66a2e4da3a56f8d256f1df7ce",
         "username" : "admin",
         "when" : 1547056298554,
         "operation" : "share",
         "source" : "item",
         "id" : "186bad9e3dd847ab9a601b505879f86d",
         "properties" : {
             "sharedToGroups" : [“Everyone”, “4adc30bb03054812a846fa592de105de”]
}}]}
```

Key Terms: Payload URL

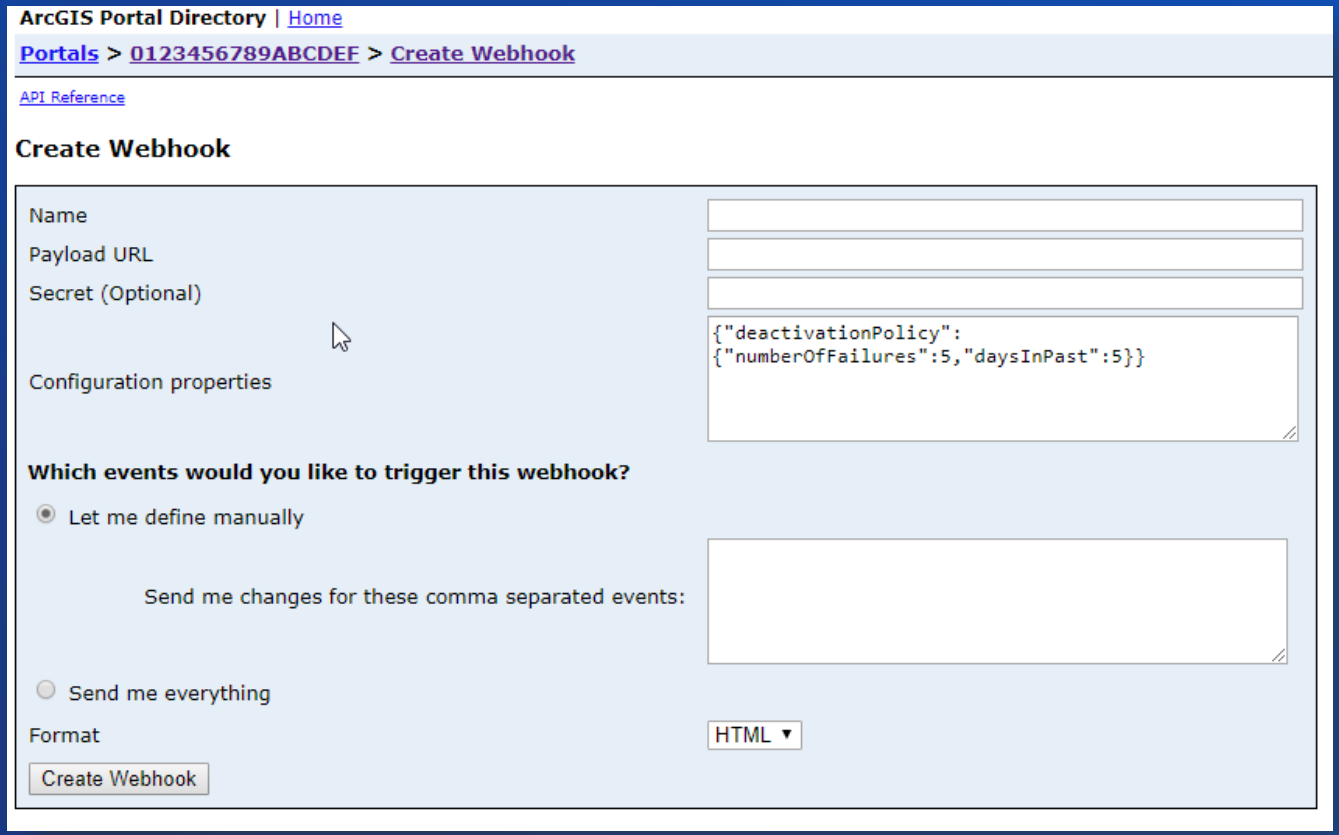
- A payload URL must be provided when creating a webhook; this defines where the payload will be delivered.

- Since all payloads are delivered through an HTTPS POST request, the webhook receiver must be configured to communicate over HTTPS and be reachable by the Portal.

- Number of options for how and where a webhook is received:
  - Enterprise SDK sample (Java servlet),
  - Third party services: Microsoft Flow, Zapier, IFTTT, Integromat
  - Custom web servers: Node JS, Python Flask

How to manage webhooks ?

- We've seen the Web Interface



How to manage webhooks ?

- **Every action is available as REST API calls**
  - **So you can manage webhooks deployment using scripts**

Portals - Webhooks

- Webhooks
- Activate Webhook
- Create Webhook
- Deactivate Webhook
- Delete Webhook
- List a Specific Webhook
- Notification Status
- Update Settings
- Update Webhook
- Webhook Settings

URL:	https://<portal-url>/webhooks
Operations:	Create Webhook
Child Resources:	List a Specific Webhook, Settings
Version Introduced:	10.7

Example usage

```
https://webadaptor.domain.com/webadaptorname/sharing/rest/portals/<org Id>/webhooks
```

JSON response example


```
{
  "webhooks": [
    {
      "id": "7c001f61f51e440baa81ab0616dc4143",
      "accountId": "0123456789ABCDEF",
      "payloadUrl": "https://machineName:8000",
      "secret": "",
      "isActive": true,
      "name": "All Updates",
      "config": {
        "deactivationPolicy": {
          "numberOfFailures": 5,
          "daysInPast": 5
        }
      },
      "ownerId": "86ccfe86be2d490a8507eaea82749dc6",
      "created": 1548281793118,
      "modified": 1548281793118,
      "events": [
        "/"
      ]
    }]
}
```

How to manage webhooks ?

- Also possible to use the ArcGIS API for Python
  - The library to manage your geospatial platform
- By using the Webhook manager and Webhook class

```
In [11]: from arcgis.gis import GIS

In [12]: gis = GIS("https://cedessltest.westeurope.cloudapp.azure.com/portal","myuser","mysecretpas

In [21]: whManager = gis.admin.webhooks
```

```
In [22]: whManager.list()

Out[22]: [<WebHook @ all items>,
          <WebHook @ allItems-otherone>,
          <WebHook @ allItems-new>,
          <WebHook @ allItems>,
          <WebHook @ allItems-Other>]
```

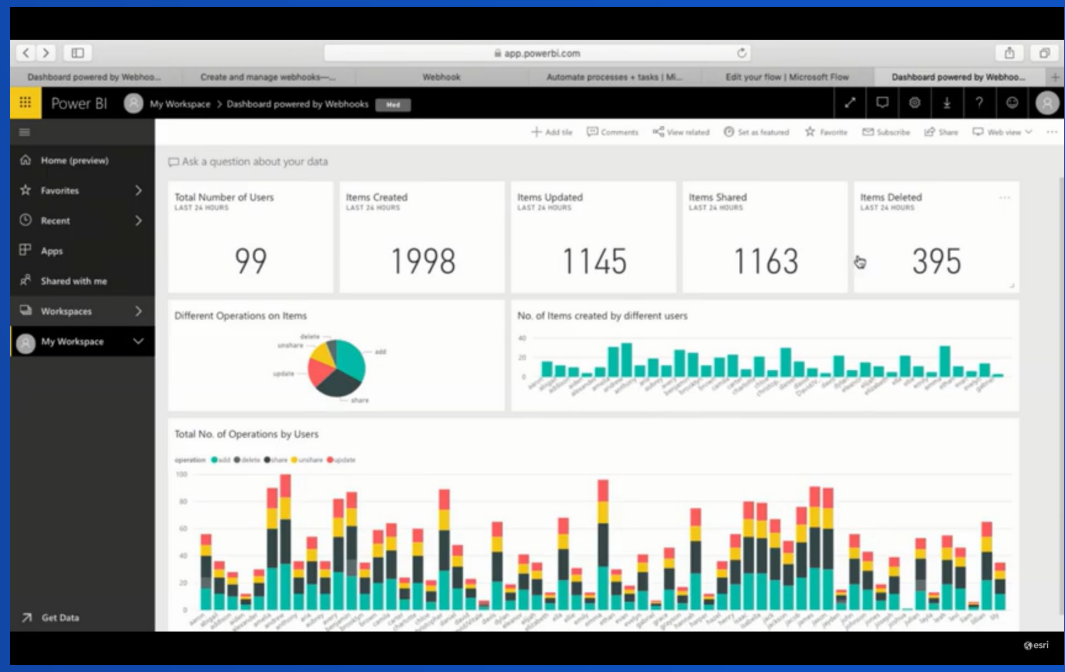
```
In [15]: whManager.list()[0].properties

Out[15]: {
           "id": "1f0b8ed2c9844e77a5500bbe1d028958",
           "accountId": "0123456789ABCDEF",
           "payloadUrl": "https://prod-114.westeurope.logic.azure.co
         c7edcbbb/triggers/manual/paths/invoke?api-version=2016-06-0
         ig=WeHVAgYfiCQG0-dgxOavjhi4mFbFgioZmVwYTyx14-U",
           "secret": "",
           "isActive": true,
           "name": "all items",
           "config": {
             "deactivationPolicy": {
               "numberOfFailures": 5,
               "daysInPast": 5
             }
           },
           "ownerId": "6f3cac6213ffceee27cc85414f458caa",
           "created": 1572272043299,
           "modified": 1572272043299,
           "events": [
```

```
In [20]: for webhook in whManager.list():
             #webhook.desactivate()
             webhook.activate()
```

Okay so what can I do with this?

Get new insights into your GIS usage

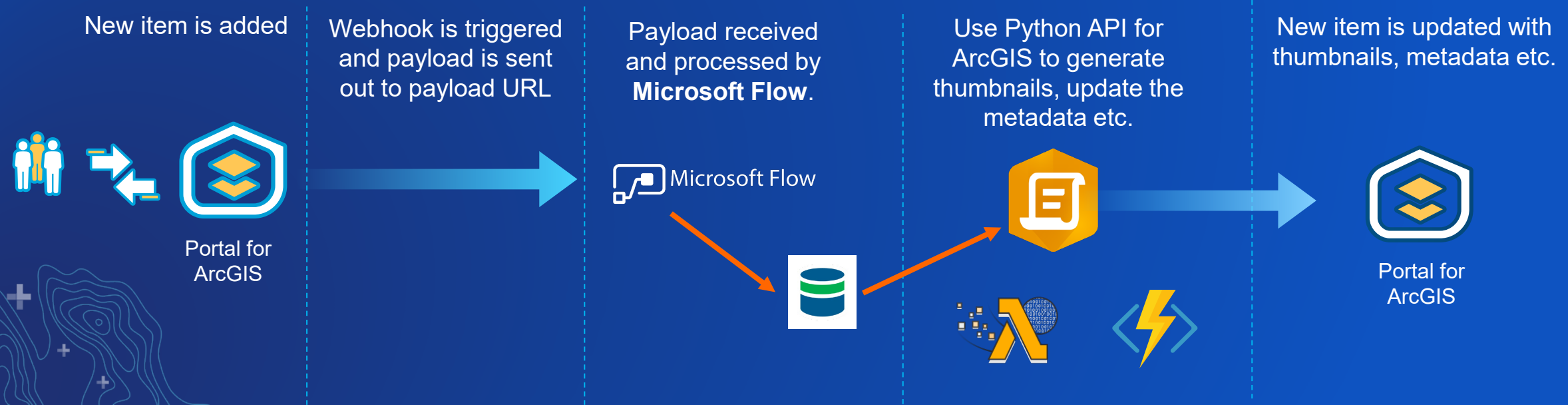
- By defining a webhook on all events
  - Who
  - What
  - When
- A lot of external connectors to save this info
  - Push into Excel, Google sheet, …
  - Push into SQL Server, Oracle, PostgreSQL, …
  - Push into a PowerBI Datasource



- With data, it's not complex to create a usage dashboard
  - Check out the Blog Post : How we did it: Webhooks at Dev Summit 2019

Additional use cases

- **Group-based workflows**
  - **Trigger webhook when an item has been shared to a specific group, or if an unauthorized user has been added to a secured group.**
- **Processing new items added to ArcGIS Enterprise portal**
  - **When a new item has been added generate thumbnails, update the metadata, share to groups of users etc.**

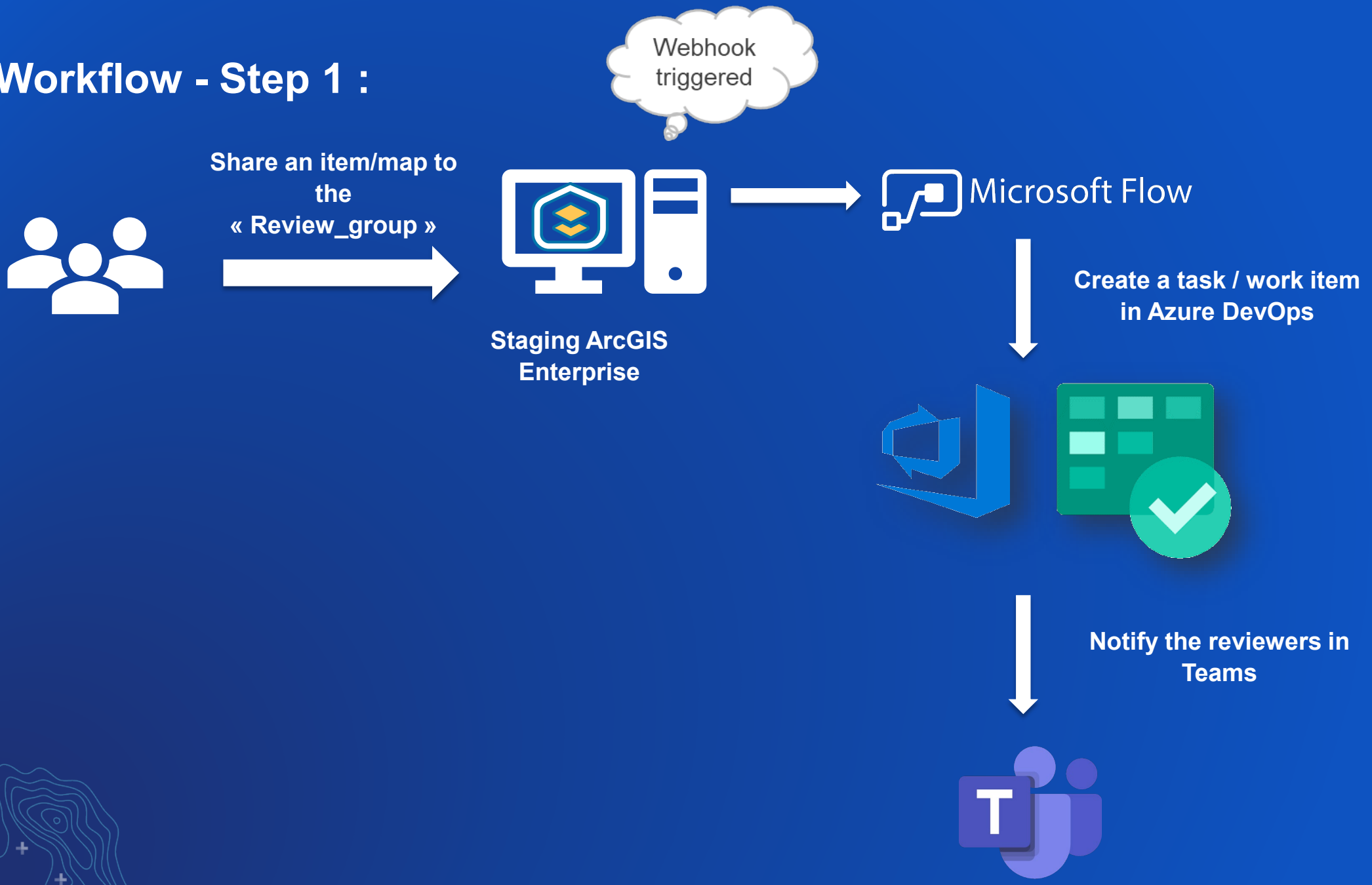


Manage more complex workflows

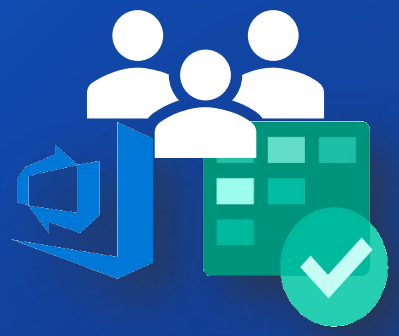
- Manage the process of portal item publication from staging to production

- Using ArcGIS Enterprise, Microsoft Flow, Azure DevOps Work items, Teams, and Azure serverless functions
  - Step 1 : When a user shares an item to the “review” group, a work item is created
  - Step 2 : When the review is finished
    - If ok : the item is cloned to the production environment using Azure serverless Python functions
    - In not ok : We nicely inform the user about the correct publishing process

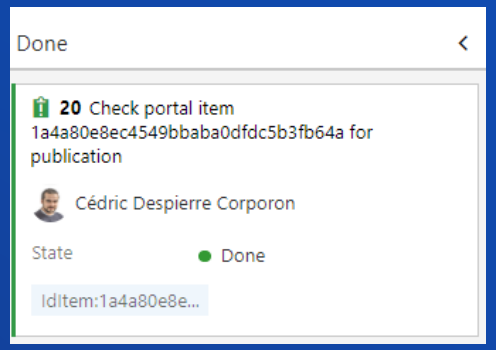
Workflow - Step 1 :



Workflow - Step 2 :



Review the map
Update the Work item status



Microsoft Flow

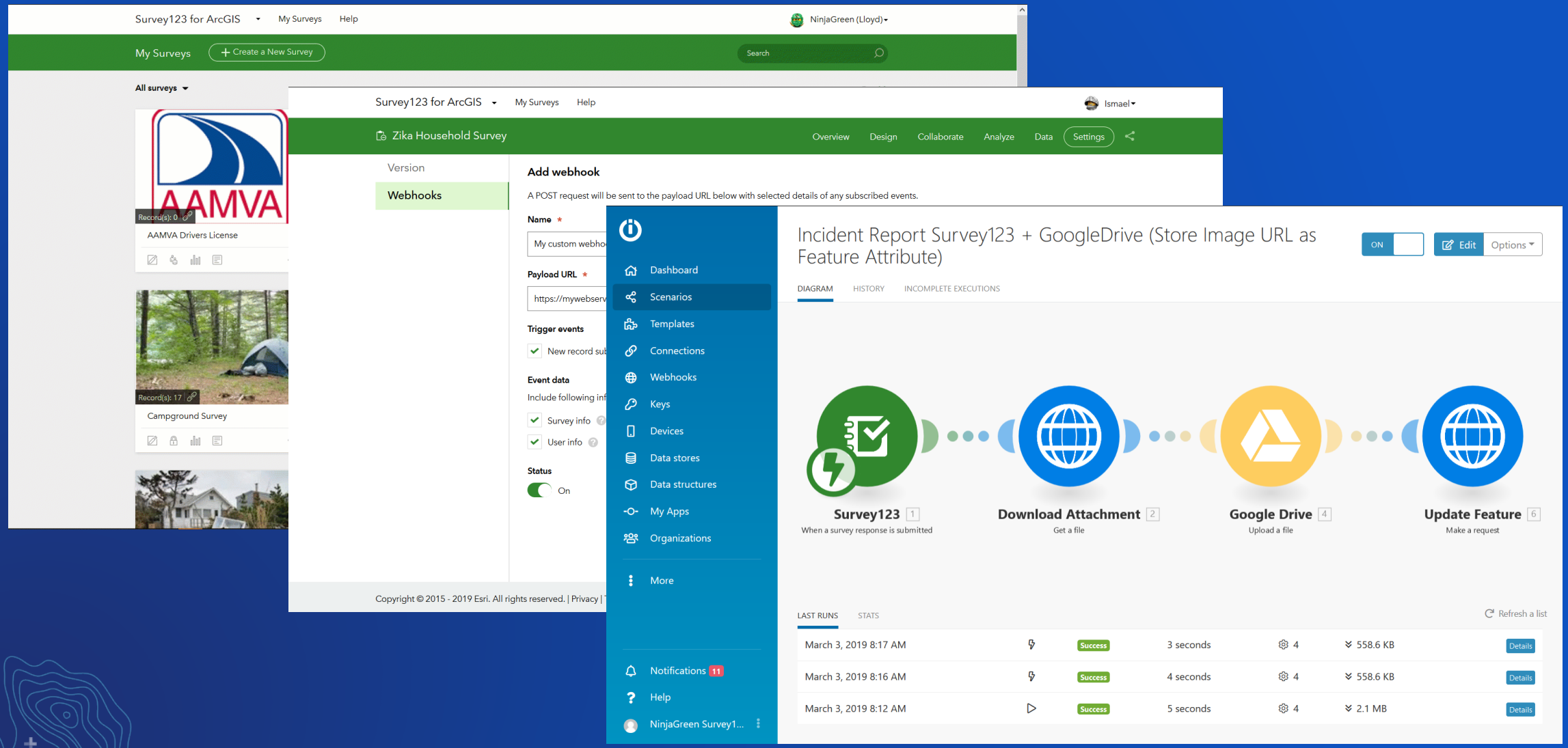
Execute a Python script in Azure function

Deep copy the content
Update sharing properties

Staging ArcGIS Enterprise

Production ArcGIS Enterprise

You also have webhooks available with Survey123



Administration tips

- Avoid redundancy
- Security
- Troubleshooting
  - Portal logs
  - Notification status
  - Deactivate
  - Fiddler
- Advanced settings
  - Number of delivery attempts
  - Notification timeout
  - Elapsed time between delivery attempts

/webhooks/<webhookID>/notificationStatus

Invoked Timestamp	Invoked Successful?	Response Text
2019-02-05 11:45:17	true	{"success":"true"}
2019-02-04 11:14:06	false	Exception: peer not authenticated
2019-02-04 11:14:06	true	{"success":"true"}
2019-02-04 09:00:34	false	Exception: peer not authenticated
2019-02-04 09:00:34	true	{"success":"true"}
2019-02-04 08:59:10	true	{"success":"true"}
2019-02-01 11:45:48	false	Exception: peer not authenticated

What’s next

- Support for additional Item, User and Group events
- Support for Feature layer editing events
- Integration with webhook services such as Microsoft Flow, Integromat, Zapier etc. in ArcGIS Enterprise
- UI and UX within ArcGIS Enterprise portal

Resources

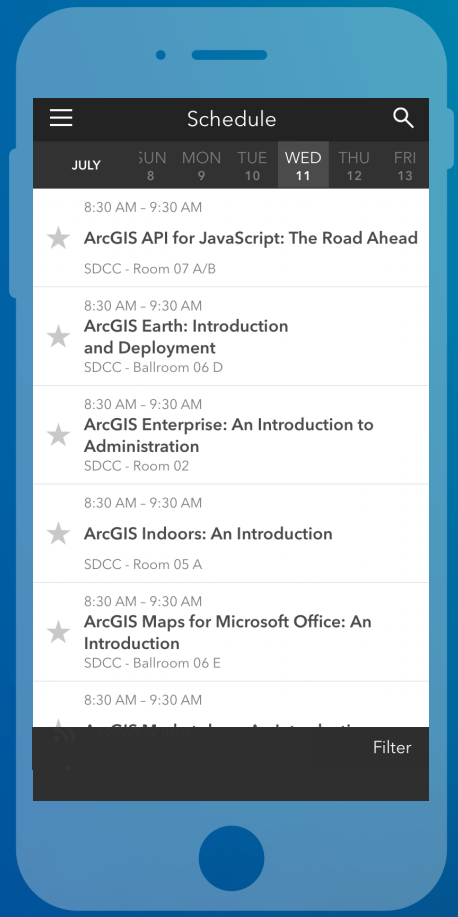
- **GeoDev Webinar**
  - https://www.esri.com/en-us/landing-page/product/2018/geodev-webinar-series
- **GitHub repository:**
  - https://github.com/Esri/webhooks-samples
- **ArcGIS Enterprise Overview:**
  - https://www.esri.com/en-us/arcgis/products/arcgis-enterprise/overview
- **Documentation:**
  - https://enterprise.arcgis.com/en/portal/latest/administer/windows/create-and-manage-webhooks.htm
  - https://developers.arcgis.com/rest/users-groups-and-items/webhooks.htm
- **Blogs:**
  - https://www.esri.com/arcgis-blog/products/arcgis-enterprise/administration/webhooks-dont-call-us-well-call-you/
  - https://www.esri.com/arcgis-blog/products/arcgis-enterprise/administration/webhooks-dev-summit-2019/

Questions ?

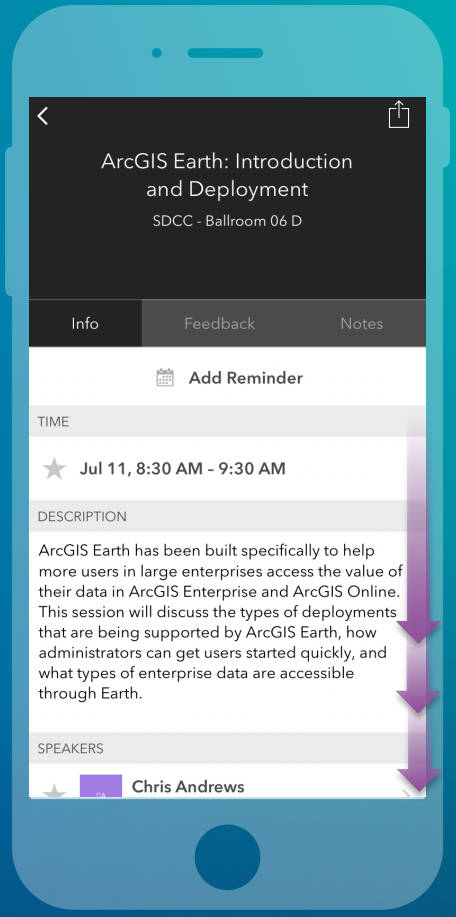
Please Take Our Survey on the App

Download the Esri Events app and find your event

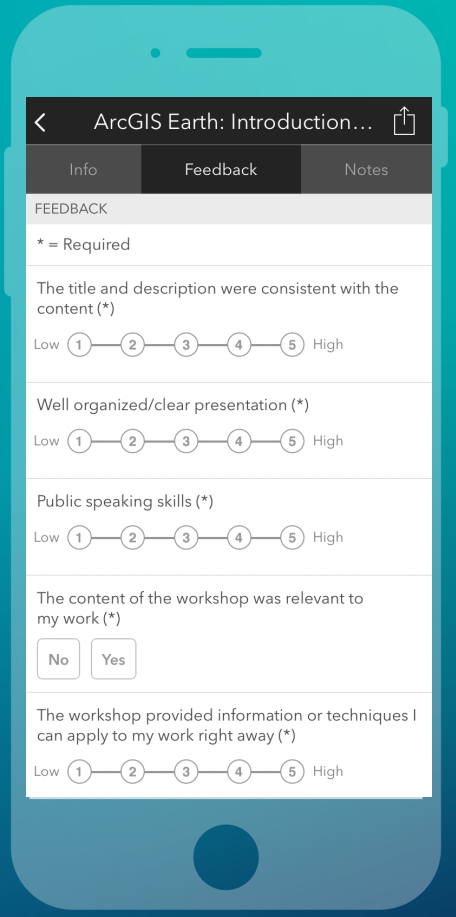
Select the session you attended



Scroll down to find the feedback section



Complete answers and select “Submit”



esri

THE SCIENCE OF WHERE