

Cogs and Gears

● JAMF
NATION
LIVE

Agenda

1 | Jamf Pro API

Quick dive into the API and a clever use case that does something very simple so well.

2 | Webhooks

Event based data, but what to do with this data?

3 | External Events

Using an external resource like Azure to help you automate devices within Jamf Pro

4 | Developer and Marketplace

Don't forget our developer and marketplace portals.





tinker / inspire...

Jamf Pro API



🙄 Classic API

It is all in the name! Classic resides on /JSSResource and accepts XML but can respond in XML and JSON.



Bearer Token

Good news, Classic API now accepts bearer token. Please update any old workflows using the older username and password method as this will be removed.



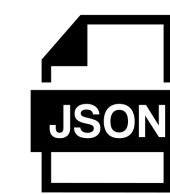
😎 Jamf Pro API

The main API moving forward. Resides at /api/v1 and only uses JSON.



XML or JSON

XML is old school, or is that CSV. JSON is the way of the future and the majority of services only use this.





<https://yourjamfporserver.com/api/>



Software Updates...

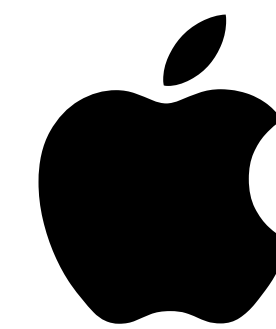
S.U.P.E.R.M.A.N

Software Update Policy Enforcement (with) Recursive Messaging And Notification



intel

- ▶ Fully automatic macOS software update workflow
- ▶ User centric




- ▶ Apple Silicon needs additional authorisation
- ▶ Three methods to update


S.U.P.E.R.M.A.N


- Configure with config profile
- bearer token and JSON
- Minimal API privileges
 - Configuration profile with computer ID
- **v1/macos-managed-software-updates/send-updates**
 - Command to download
 - Command to install / force restart

```
<key>JamfProID</key>  
<string>$JSSID</string>  
<key>DefaultDefer</key>  
<string>100</string>  
<key>FocusDefer</key>  
<string>100</string>  
<key>MenuDefer</key>  
<string>100,200,300</string>  
<key>DisplayTimeout</key>  
<string>15</string>  
<key>DisplayRedraw</key>  
<string>10</string>  
<string>icon.icns</string>  
<key>IconSizeIbm</key>  
<string>128</string>  
<key>IconSizeJamf</key>  
<string>128</string>  
<key>CountDeadline</key>  
<string>5</string>  
<key>PreferJamf</key>  
<false/>  
<key>ForceRestart</key>  
<true/>  
<key>VerboseMode</key>
```

Setup

 Jamf Pro

 policy

 super

	CREATE	READ	UPDATE	DELETE
Jamf Pro Server Objects Create, Read, Update and Delete				
Computers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jamf Pro Server Actions				
Send Computer Remote Command to Download and Install macOS Update				<input checked="" type="checkbox"/>

Setup



 policy

 super

```
/Library/Management/super/super \  
--jamf-account "AccountName" \  
--jamf-password "Password"
```



Setup



 policy

 super

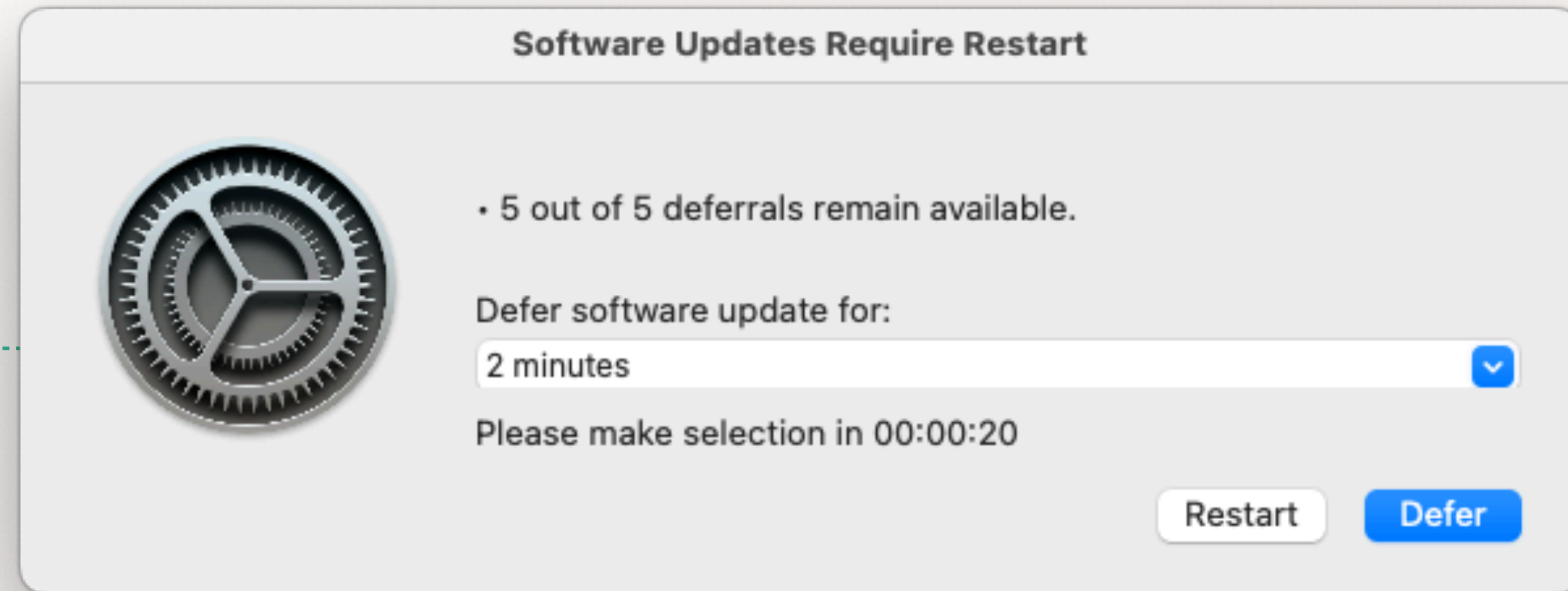
```
{  
  "deviceIds": [  
    "${jamfProID}"  
  ],  
  "skipVersionVerification": false,  
  "applyMajorUpdate": false,  
  "updateAction": "DOWNLOAD ONLY"  
}
```

Setup



 super

 notification



Setup



 super

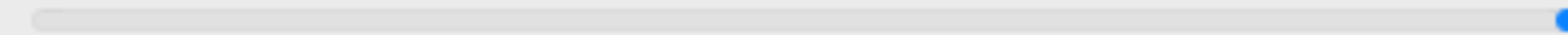
 notification

Software Updates Require Restart



A required software update will automatically restart this computer in about 5 minutes.

During this time you can continue to use the computer or lock the screen, but please do not restart or sleep the computer as it will prolong the update process.



Setup



 super

 notification

```
{  
  "deviceIds": [  
    "${jamfProID}"  
  ],  
  "skipVersionVerification": false,  
  "applyMajorUpdate": false,  
  "updateAction": "DOWNLOAD_AND_INSTALL",  
  "forceRestart": true  
}
```

Setup



 super

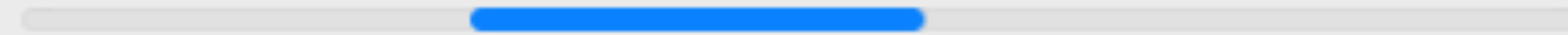
 notification

Software Updates Require Restart



This computer will automatically restart very soon.

Save any open documents now.



Takeaway

- ▶ Download S.U.P.E.R.M.A.N
- ▶ Very smart use of the API to solve a problem
- ▶ Security first
- ▶ Make sure to watch Bill Smiths JNUC talk from 2021
 - ▶ *How to collect user information and apply it throughout Jamf Pro*



Jamf Pro Webhooks



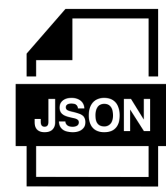
Event Based

Webhooks happen when there is an event with Jamf Pro. Could be a smartgroup membership change or when a device enrolls for the first time.



Basic or Header Authentication

You can secure your webhook endpoint with either the username password combo, or use an authentication header



JSON

Event data is sent as JSON



XML

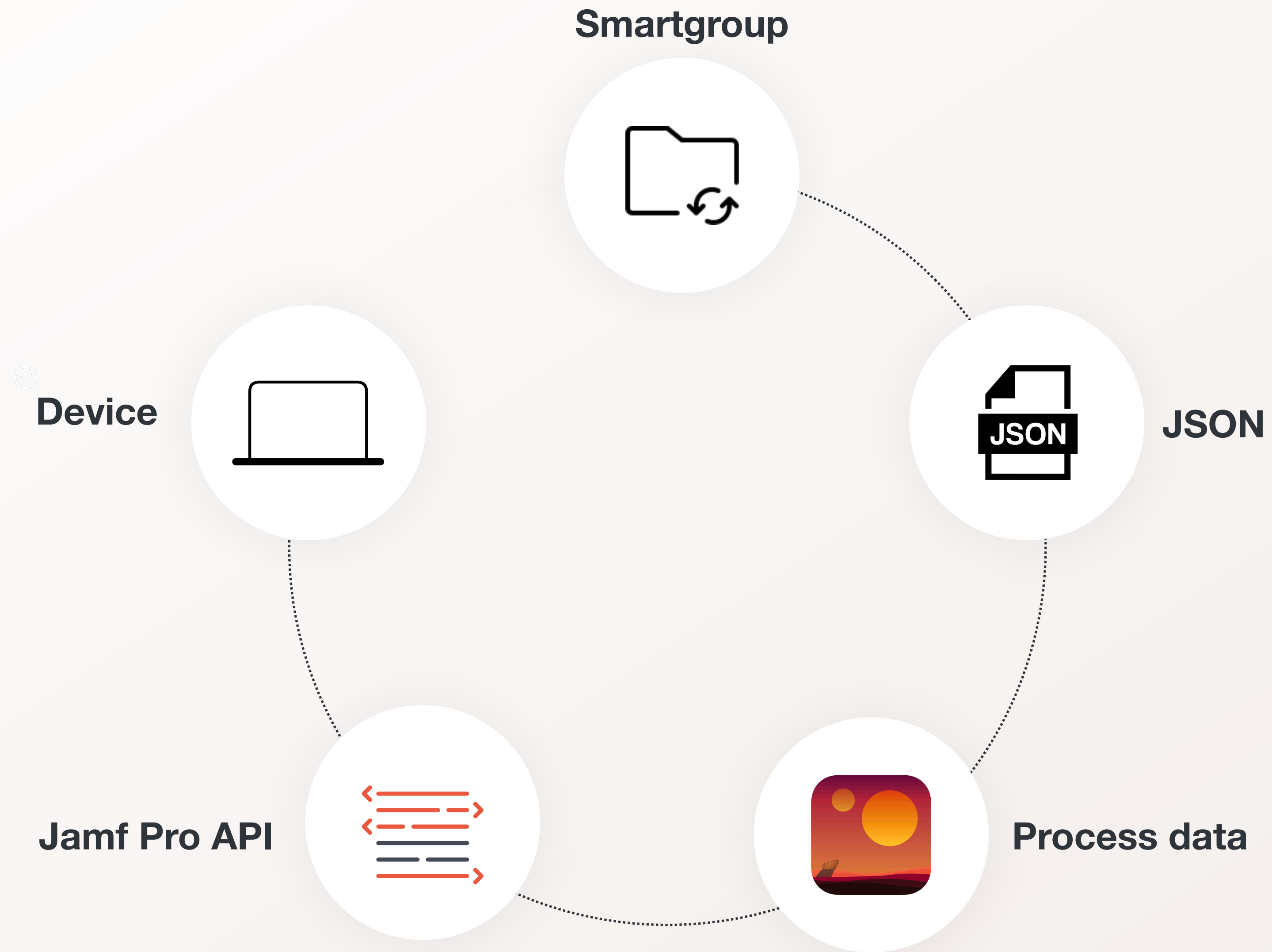
Event data is sent as XML

Webhooks in Jamf Pro

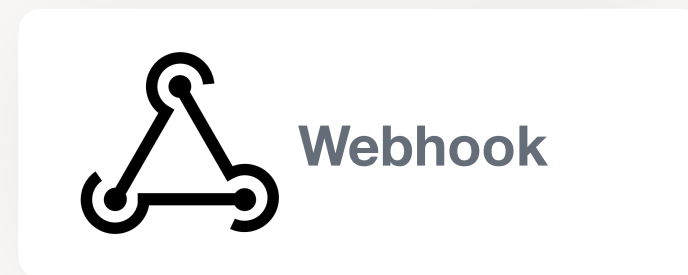
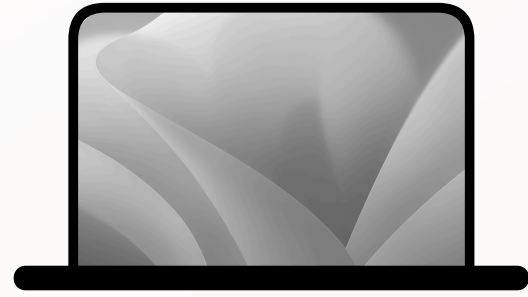
Events within Jamf Pro

- ✓ ComputerAdded
- ComputerCheckIn
- ComputerInventoryCompleted
- ComputerPatchPolicyCompleted
- ComputerPolicyFinished
- ComputerPushCapabilityChanged
- DeviceAddedToDEP
- DeviceRateLimited
- JSSShutdown
- JSSStartup
- MobileDeviceCheckIn
- MobileDeviceCommandCompleted
- MobileDeviceEnrolled
- MobileDevicePushSent
- MobileDeviceUnEnrolled
- PatchSoftwareTitleUpdated
- PushSent
- RestAPIOperation
- SCEPChallenge
- SmartGroupComputerMembershipChange**
- SmartGroupMobileDeviceMembershipChange
- SmartGroupUserMembershipChange

Dataflow



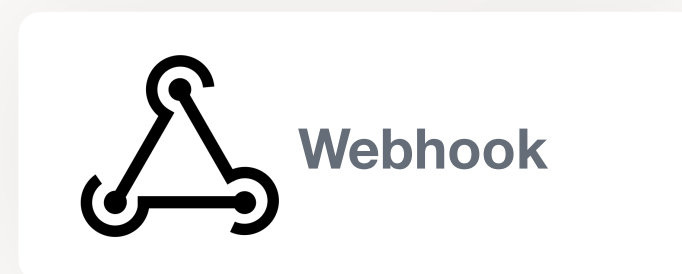
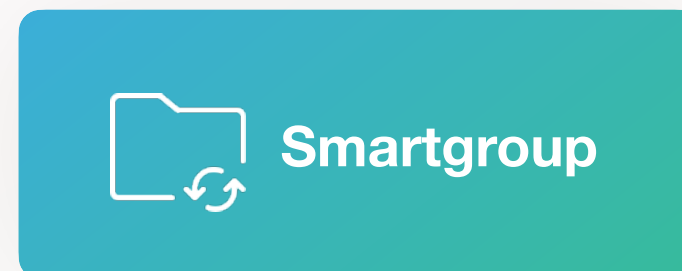
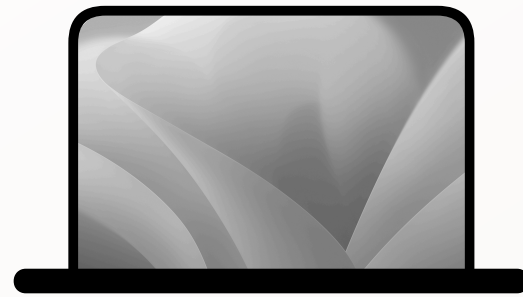
Setup



Smart Group Criteria

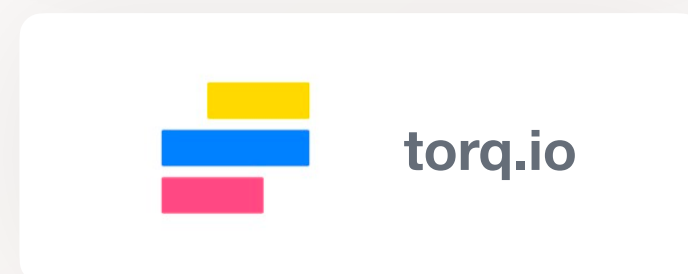
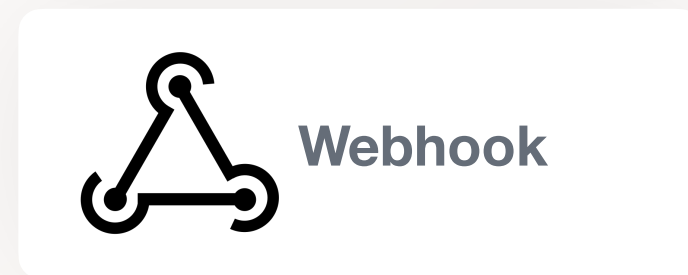
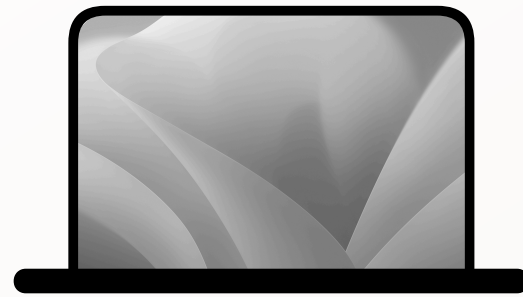
- **Last Check-in more then 14 days**
- **Profile Name has MDM Profile**

Setup



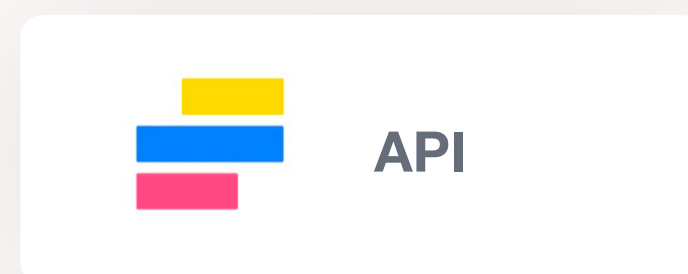
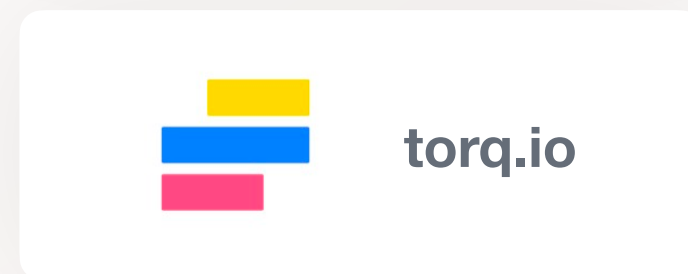
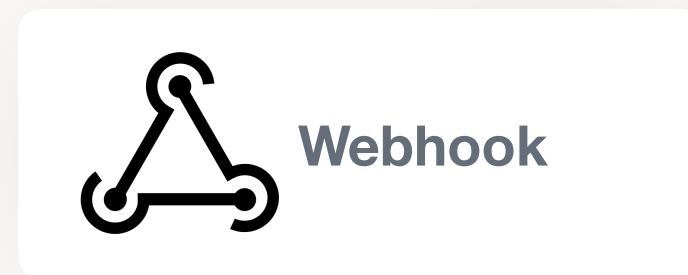
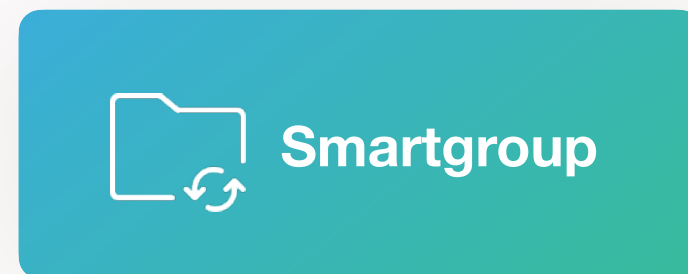
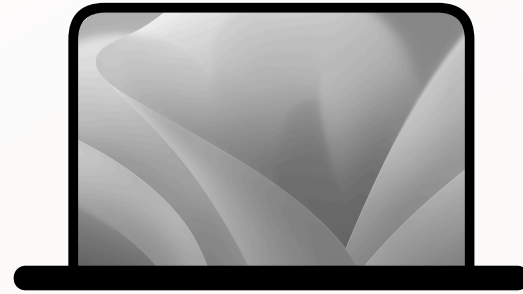
```
{
  "event": {
    "name": "Framework Update Needed",
    "jssid": 15,
    "computer": true,
    "smartGroup": true,
    "groupAddedDevices": [],
    "groupRemovedDevices": [],
    "groupAddedDevicesIds": [
      18
    ],
    "groupRemovedDevicesIds": []
  },
  "webhook": {
    "id": 6,
    "name": "Update Framework",
    "webhookEvent":
    "SmartGroupComputerMembershipChange",
    "eventTimestamp": 1654252593148
  }
}
```

Setup

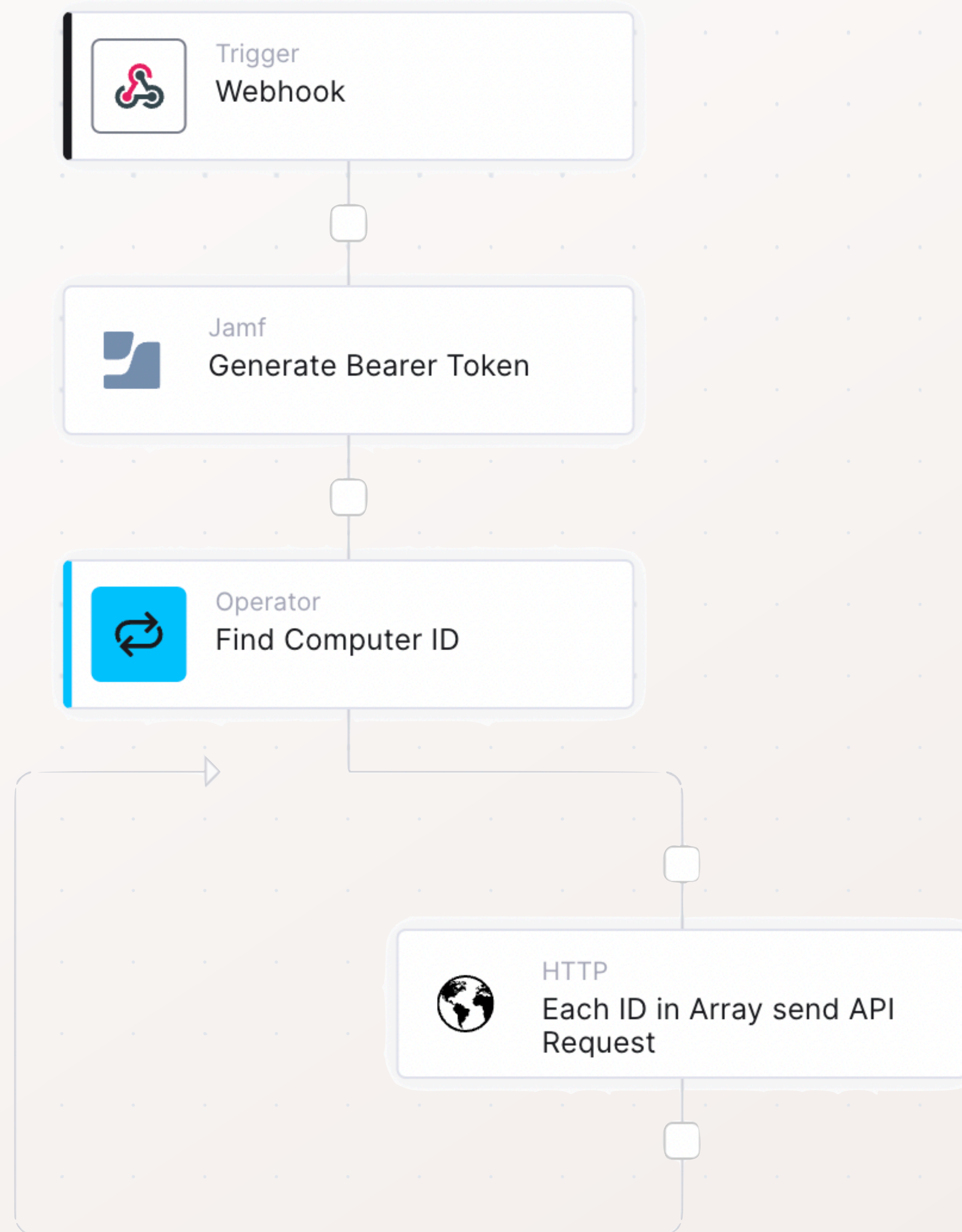


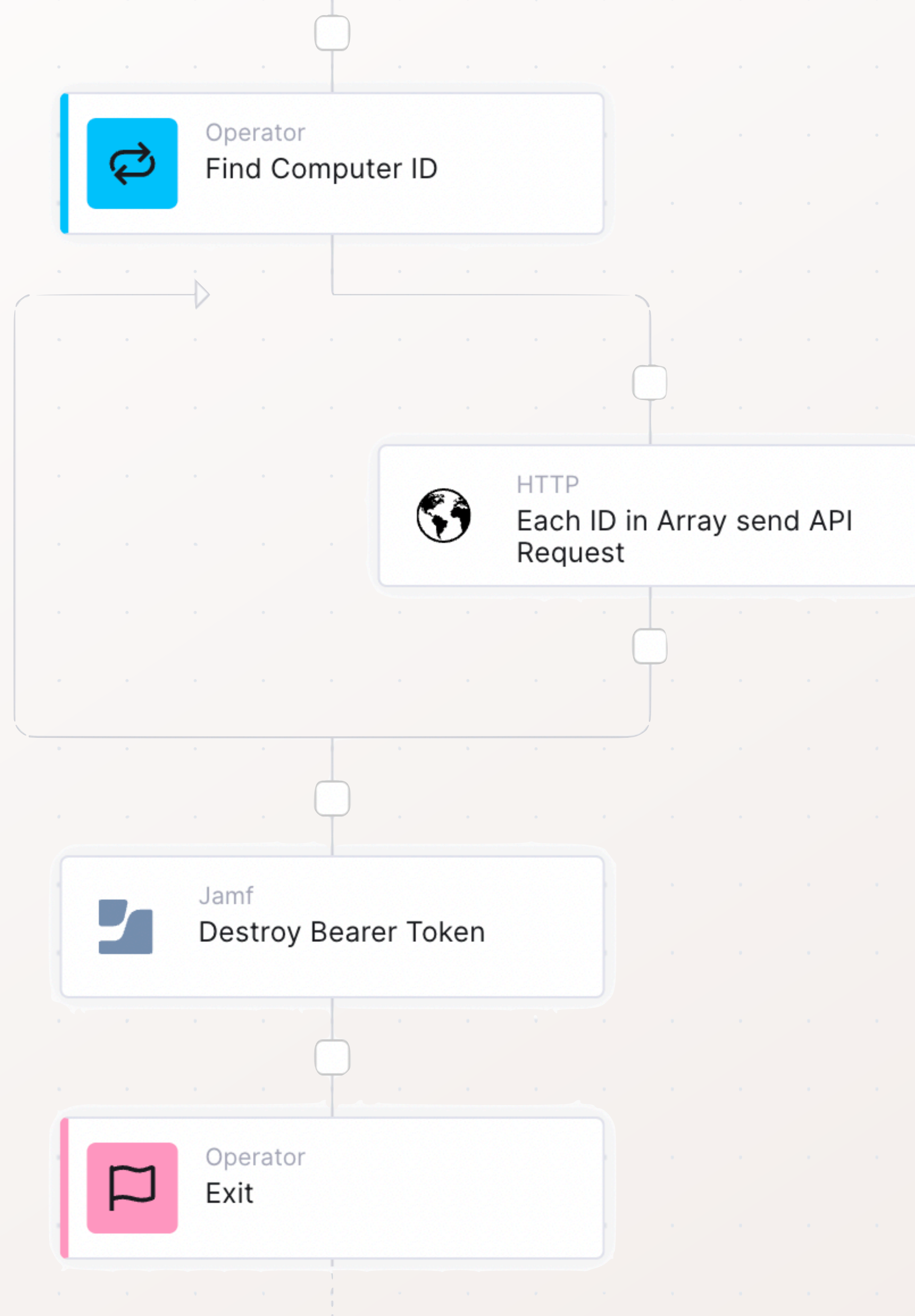
```
"groupAddedDeviceIds": [  
  18  
],
```

Setup



<https://emeia.jamfce.com/api/v1/jamf-management-framework/redeploy/18>





Setup

The screenshot displays the Jamf Pro web interface. On the left is a dark sidebar with navigation options: Computers, Devices, and Users. Below these are sections for INVENTORY (Search Inventory, Search Volume Content, Licensed Software), CONTENT MANAGEMENT (Policies, Configuration Profiles, Restricted Software, Mac Apps, Patch Management, eBooks), GROUPS (Smart Computer Groups, Static Computer Groups, Classes), and ENROLLMENT (Enrollment Invitations, ProStage Enrollments). The main content area is titled 'Computers' and shows a breadcrumb for 'admin's Mac mini'. Underneath are tabs for 'Inventory', 'Management', and 'History'. A list of log types is shown on the left, including Application Usage Logs, Computer Usage Logs, Audit Logs, Policy Logs, Patch Management Logs, Jamf Remote Logs, and Screen Sharing Logs. The 'Management History' log is selected and expanded on the right, showing '144 Completed Commands' and a search filter. A red box highlights the 'InstallApplication' command in the list of commands.

jamf | PRO

Computers Devices Users

INVENTORY

- Search Inventory
- Search Volume Content
- Licensed Software

CONTENT MANAGEMENT

- Policies
- Configuration Profiles
- Restricted Software
- Mac Apps
- Patch Management
- eBooks

GROUPS

- Smart Computer Groups
- Static Computer Groups
- Classes

ENROLLMENT

- Enrollment Invitations
- ProStage Enrollments

Computers

← admin's Mac mini

Inventory Management History

- Application Usage Logs
- Computer Usage Logs
- Audit Logs
- Policy Logs
- Patch Management Logs
- Jamf Remote Logs
- Screen Sharing Logs
- Management History > 0 Pending, 0 Failed

Management History

144 Completed Commands

Filter Re

COMMAND

- InstallApplication
- ProfileList
- iTunes Account Status
- iTunes Account Info
- ProfileList
- UserList

Takeaway

- ▶ Event based triggers within Jamf Pro
- ▶ Webhooks can be sent from a script
- ▶ Be creative
- ▶ Respect security

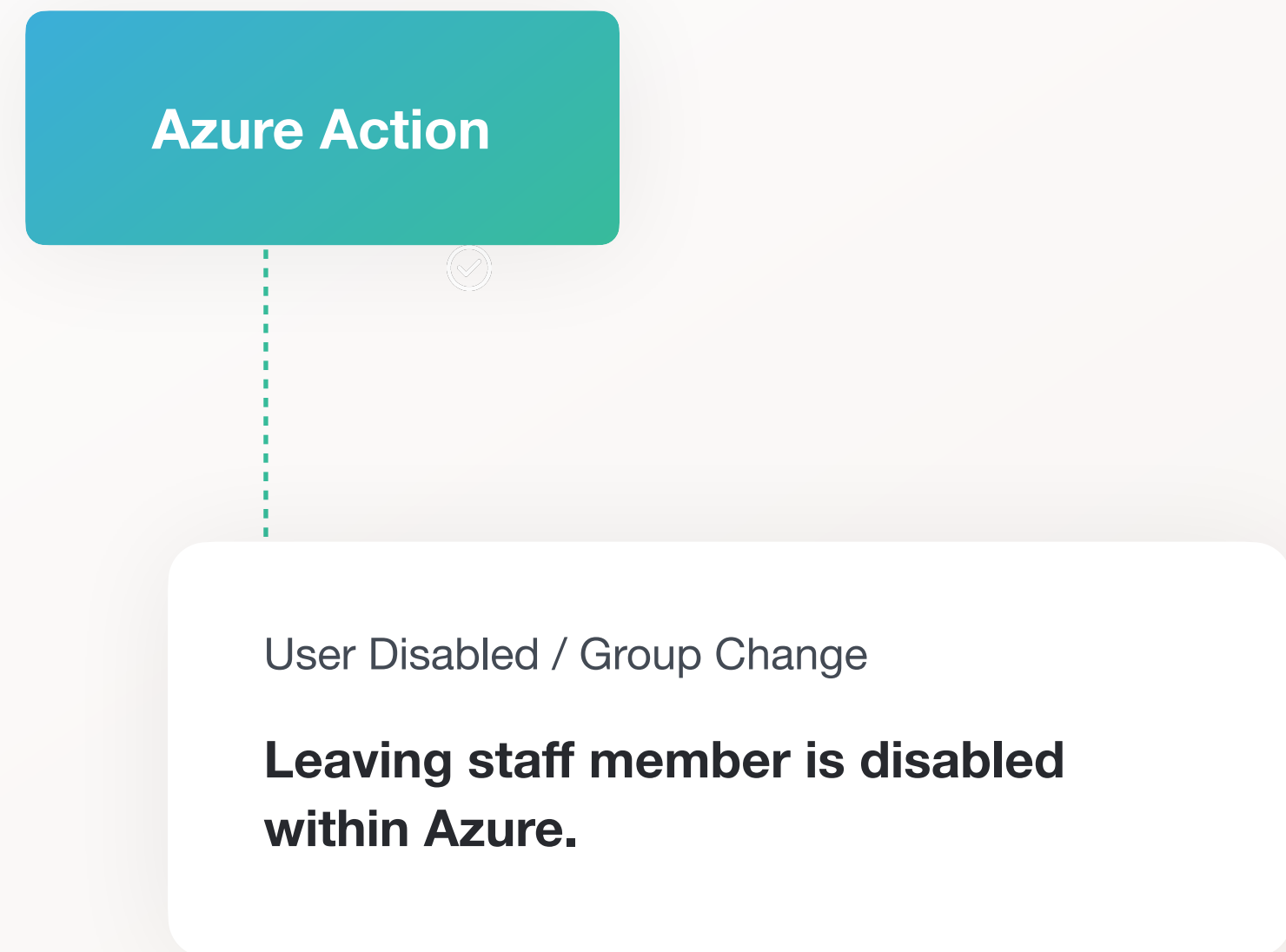


Azure Events

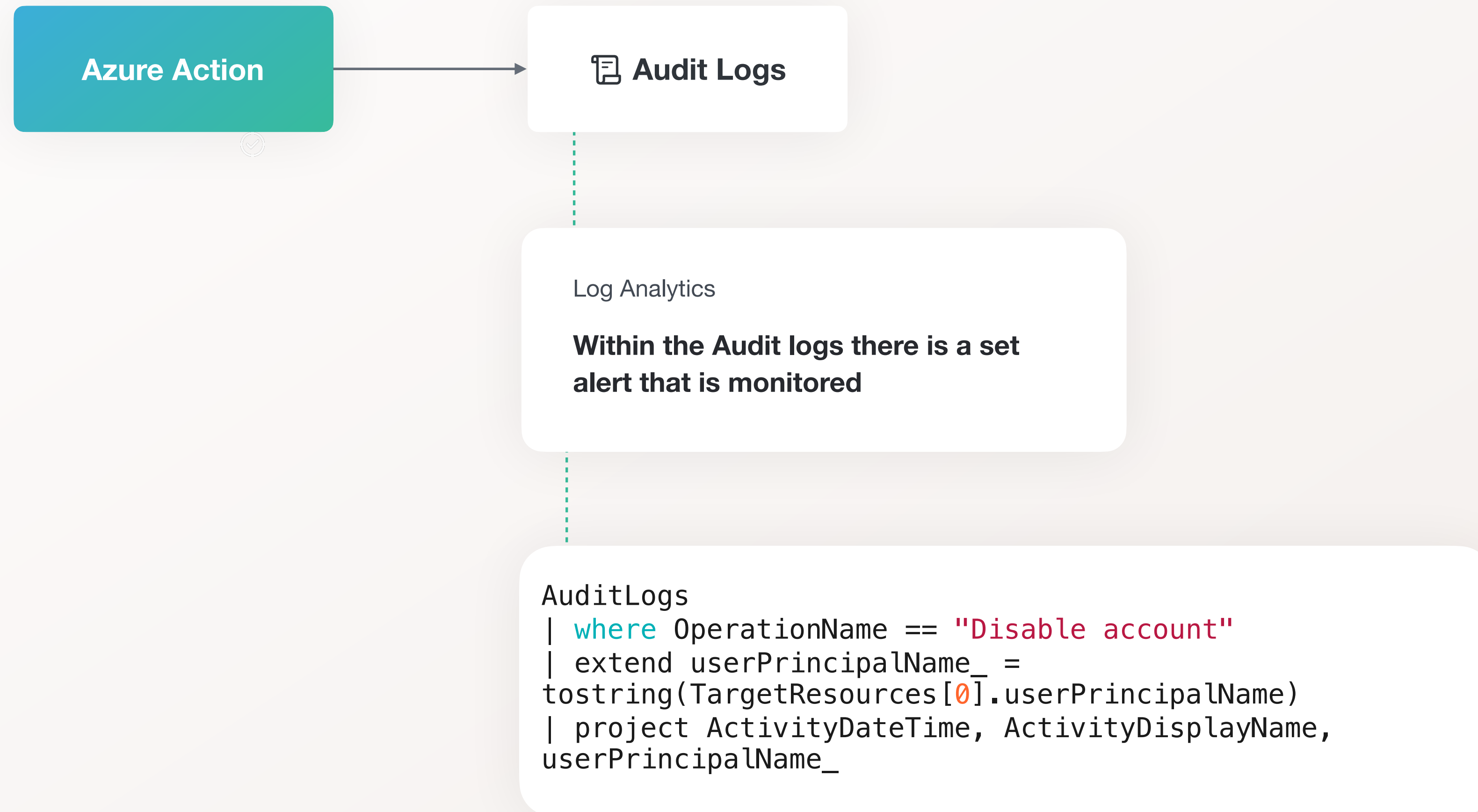
- ▶ Using changes within Azure
- ▶ User orientated
 - ▶ Disabled User
 - ▶ User group changed
- ▶ Using automations to then write back via API



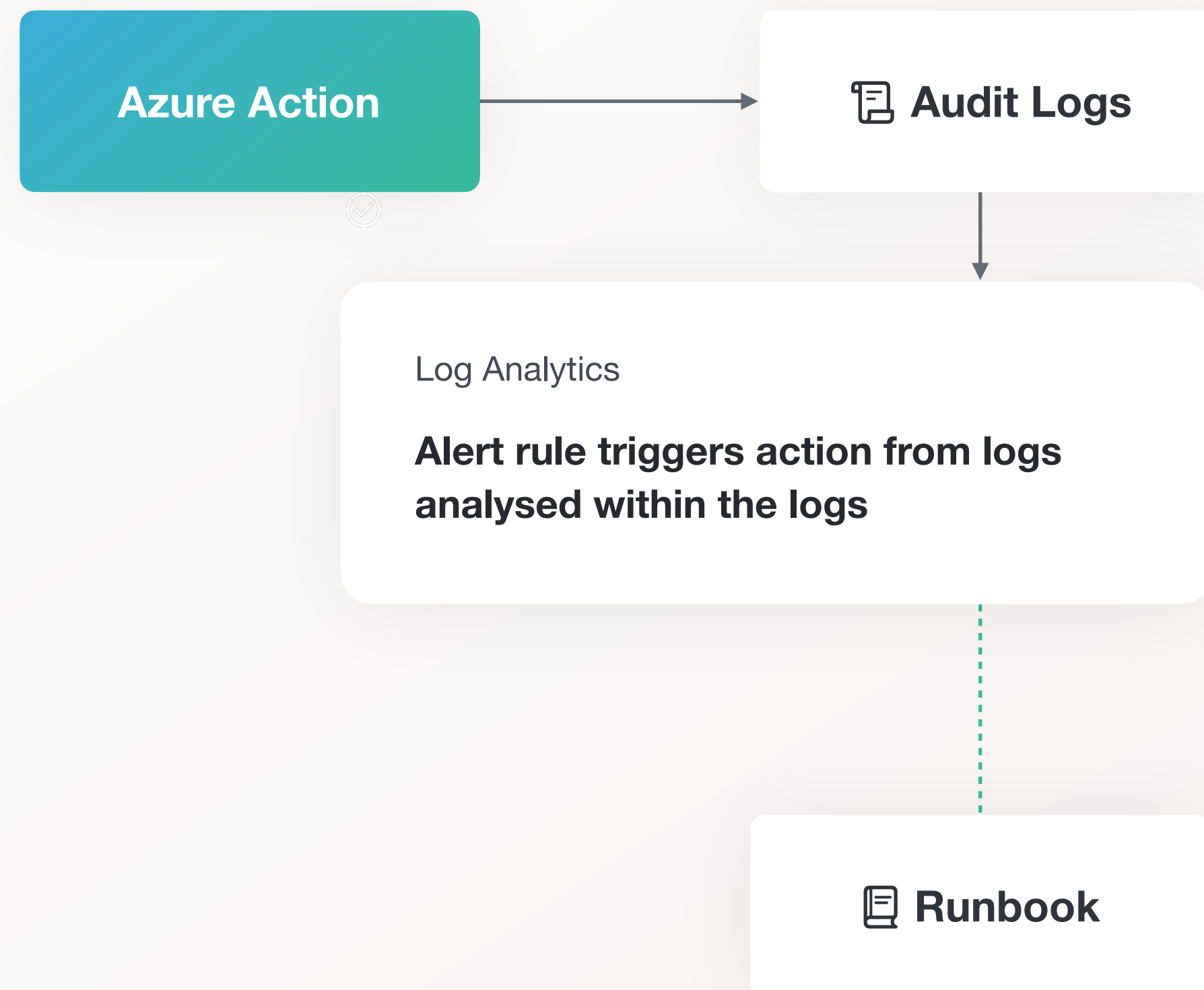
Workflow



Workflow

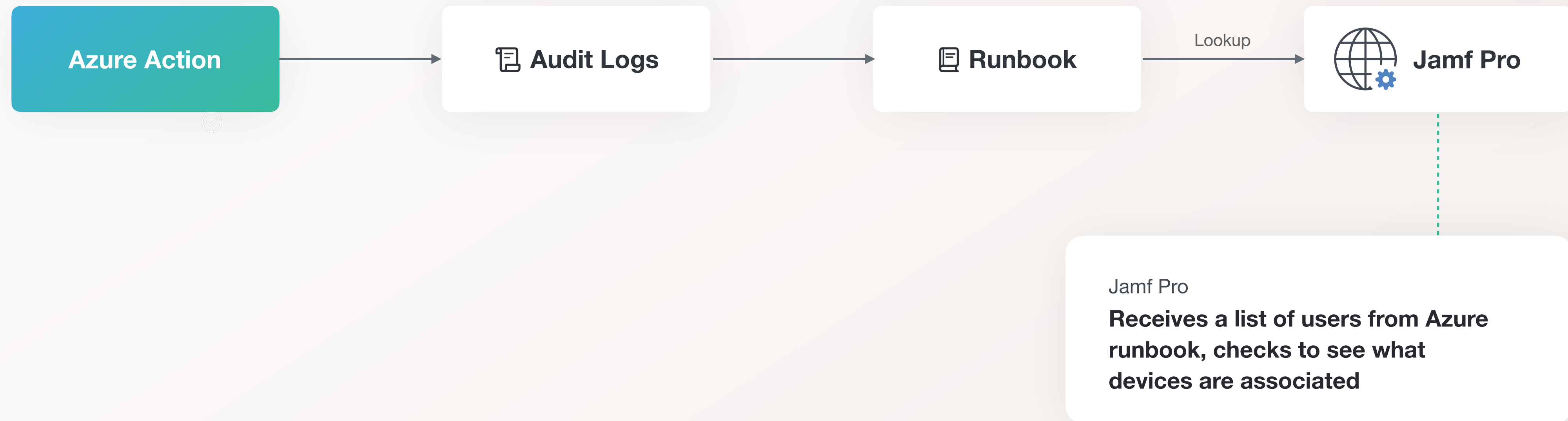


Workflow

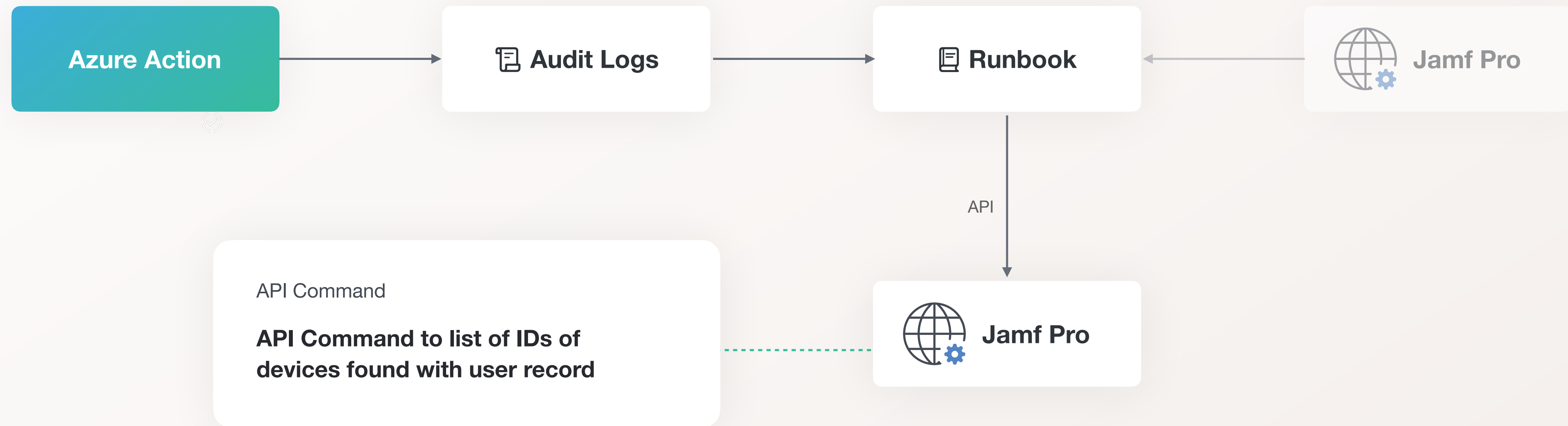


```
#####  
# Part 1: The Infos From Azure  
#####  
  
If ($WebhookData){  
  
    $JsonTmp = $WebhookData.RequestBody | ConvertFrom-Json  
    $NumberOfResults = $JsonTmp.data.alertContext.ResultCount  
    $JsonData = $JsonTmp.data.alertContext.SearchResults  
  
    If ($NumberOfResults -eq 1){  
  
        $Columns = $JsonData.tables.columns.Name  
  
        $TableList = $JsonData.tables.rows.ForEach({  
            begin {  
                $i = 0  
                $out = [ordered]@{}  
            }  
            process {  
                $Out[$Columns[$i++]] = $_  
                if($i -eq $Columns.count)  
                {  
                    [pscustomobject]$Out  
                    $Out = [ordered]@{}  
                    $i = 0  
                }  
            }  
        })  
    } Else {  
        $TableList =  
        ForEach($Row in $JsonData.tables.rows )  
        {  
            $TmpHash = [Ordered]@{}  
            For($i = 0; $i -lt $Row.Length; ++$i )  
            {  
                $TmpHash.Add( $JsonData.tables.columns.name[$i], $Row[$i] )  
            }  
            [PSCustomObject]$TmpHash  
        }  
    }  
} Else {
```

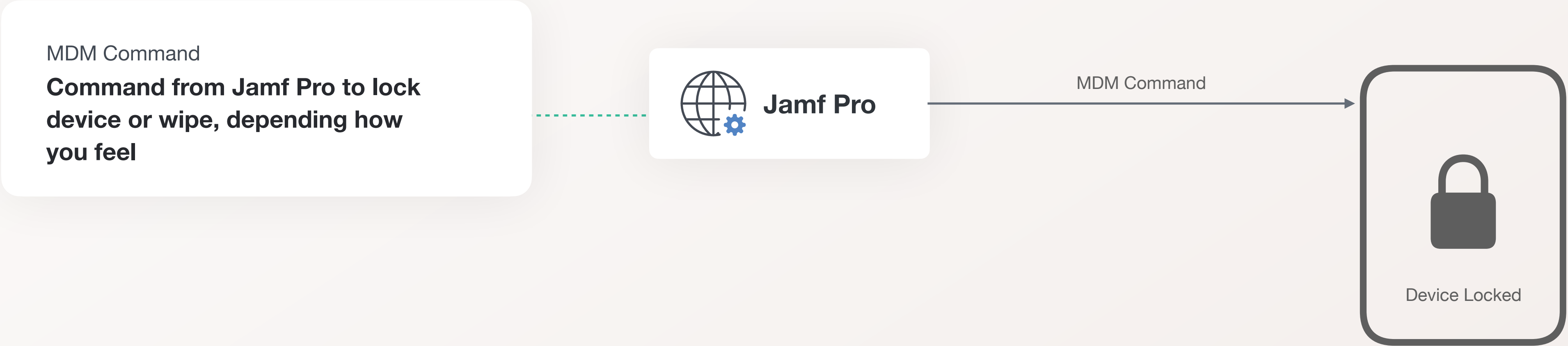
Workflow



Workflow



Workflow



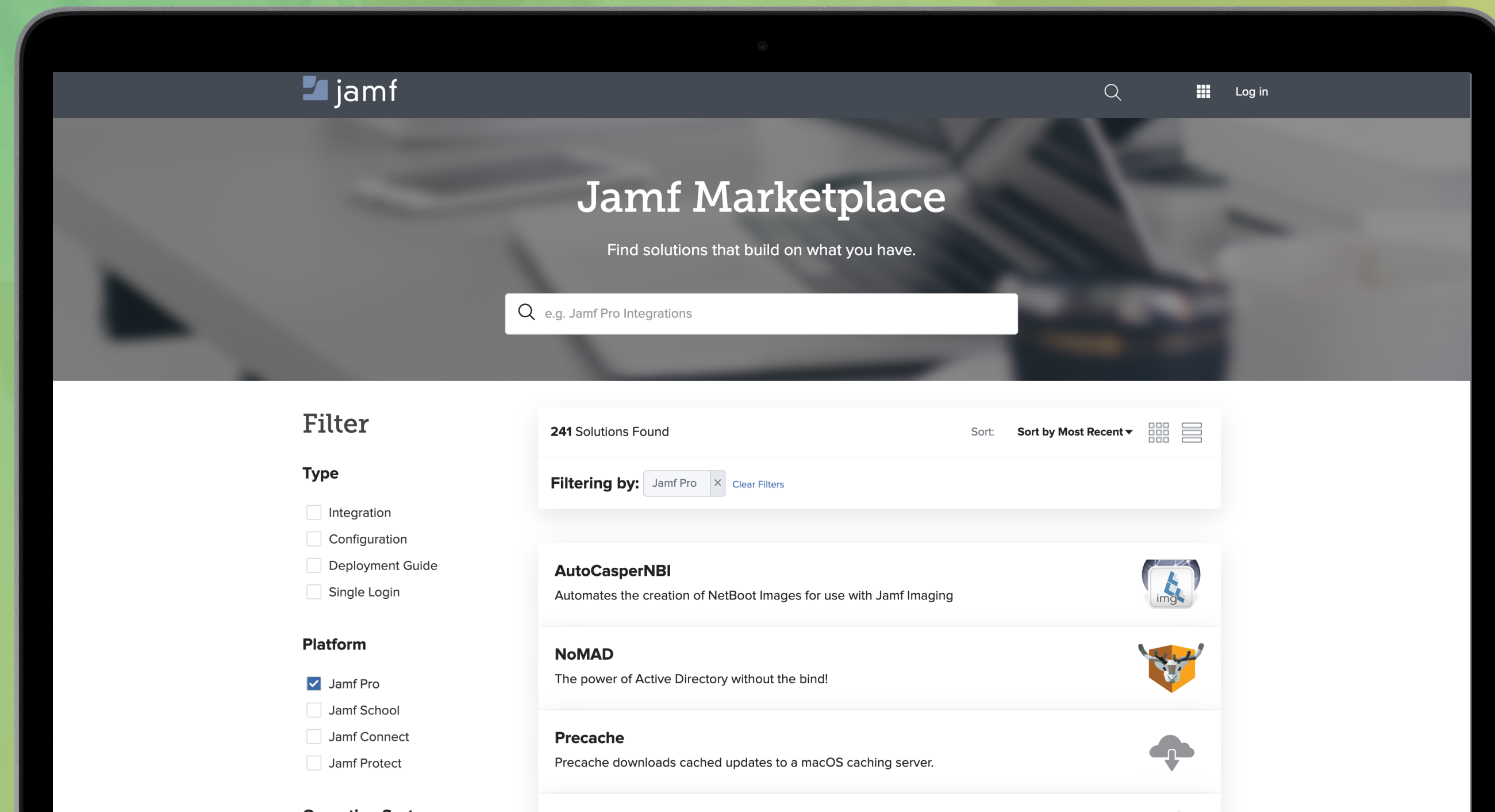
Takeaway

- ▶ Doesn't have to be a webhook
- ▶ External systems can write to Jamf Pro
- ▶ Think outside the box



Where can I find more info?

If you want to be as happy as the people on this slide...



The screenshot displays the Jamf Marketplace website interface. At the top, the Jamf logo is visible on the left, and search, grid, and login icons are on the right. The main heading is "Jamf Marketplace" with the tagline "Find solutions that build on what you have." Below this is a search bar containing the text "e.g. Jamf Pro Integrations".

The results section shows "241 Solutions Found" and is sorted by "Sort by Most Recent". A filter bar indicates "Filtering by: Jamf Pro" with a "Clear Filters" option. The left sidebar contains filter categories:

- Filter**
- Type**
 - Integration
 - Configuration
 - Deployment Guide
 - Single Login
- Platform**
 - Jamf Pro
 - Jamf School
 - Jamf Connect
 - Jamf Protect

The main content area lists three solutions:

- AutoCasperNBI**: Automates the creation of NetBoot Images for use with Jamf Imaging. (Icon: img)
- NoMAD**: The power of Active Directory without the bind! (Icon: NoMAD logo)
- Precache**: Precache downloads cached updates to a macOS caching server. (Icon: cloud with arrow)

marketplace.jamf.com

From a seamless onboarding experience to empowering users to get the most up-to-date resources needed to be productive, apps are the crux of every step in a user's journey.

The screenshot shows the Jamf Marketplace interface. At the top, there is a dark navigation bar with the Jamf logo, a search icon, a grid icon, and a 'Log in' link. Below this is a large hero section with the text 'Jamf Marketplace' and the tagline 'Find solutions that build on what you have.' A search bar contains the text 'e.g. Jamf Pro Integrations'. The main content area is divided into a left sidebar and a right main panel. The sidebar has a 'Filter' section with three categories: 'Type' (Integration, Configuration, Deployment Guide, Single Login), 'Platform' (Jamf Pro, Jamf School, Jamf Connect, Jamf Protect), and 'Operating System' (macOS, iPadOS). The main panel shows '241 Solutions Found' and a 'Sort by Most Recent' dropdown. Below this, there is a 'Filtering by:' section with a 'Jamf Pro' tag and a 'Clear Filters' link. The main panel displays a list of solutions, each with a title, description, and icon: 'AutoCasperNBI' (Automates the creation of NetBoot Images for use with Jamf Imaging), 'NoMAD' (The power of Active Directory without the bind!), 'Precache' (Precache downloads cached updates to a macOS caching server.), and 'The MUT' (A GUI tool designed to simplify mass updates to your devices in Jamf.).

developer.jamf.com

From a seamless onboarding experience to empowering users to get the most up-to-date resources needed to be productive, apps are the crux of every step in a user's journey.

The screenshot shows the Jamf developer API documentation page. The top navigation bar includes the Jamf logo, version 'v10.38.0', and links for Home, Guides, API Reference, and Changelog. A search bar is located on the right. The left sidebar contains a list of API endpoints under the 'api-authentication' category, with the selected endpoint highlighted. The main content area displays the endpoint title, a GET method, and the URL 'https://yourServer.jamfcloud.com/api/v1/auth'. Below this, there is a description, a note about authentication details, and a 'RESPONSE' section showing a 200 status code and the text 'Current authorization details.'. A 'CURL' section provides a request example. The right sidebar shows language options (Shell, Swift, Ruby, Python) and a 'RESPONSE' section with an 'Examples' dropdown.

jamf

API References Documentation

v10.38.0 Home Guides API Reference Changelog

Search %K

advanced-mobile-device-searches

advanced-user-content-searches

api-authentication

- Get all the Authorization details associated with the current api **GET**
- Get the authorization details associated with the current API token **POST**
- Invalidate current token **POST**
- Invalidate existing token and generates new token **POST**
- Create a token based on other authentication details (basic, etc.) **POST**

app-dynamics-configuration-preview

app-request-preview

app-store-country-codes-preview

buildings

cache-settings

categories

certificate-authority

classic-ldap

client-check-in

cloud-azure

Get all the Authorization details associated with the current api

GET <https://yourServer.jamfcloud.com/api/v1/auth>

Get all the authorization details associated with the current api token

Additional information about authentication, including a Postman collection, can be found in the [Jamf Pro API Overview](#).

RESPONSE

200
Current authorization details.

Updated 7 months ago

Create a token based on other authentication details (basic, etc.) ← Invalidate current token →

LANGUAGE

Shell Swift Ruby Python

CURL

REQUEST

```
1 curl --request GET \  
2 --url https://yourserver.jamfcloud.com/api/v1/auth \  
3 --header 'Accept: application/json'
```

RESPONSE Examples

Choose an example:

application/json

200

Thank You



Rob Potvin

Consulting Engineer
Jamf



GO HERE 