



Microsoft Entra ID Governance

Employee Lifecycle Automation

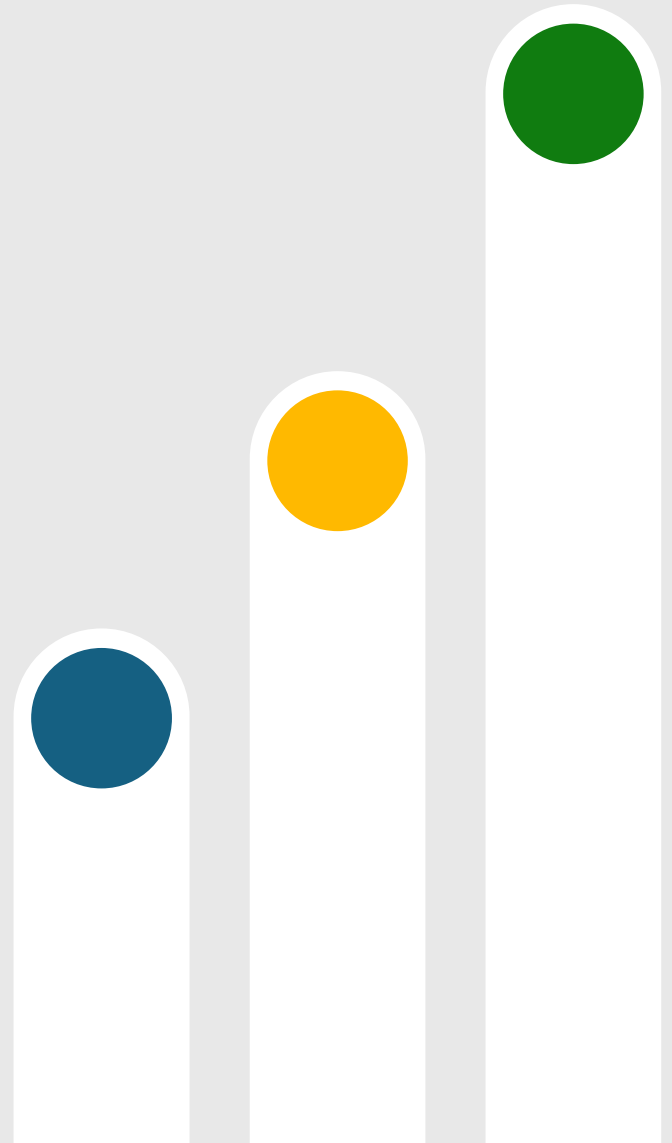
Proof of concept deployment



Introduction

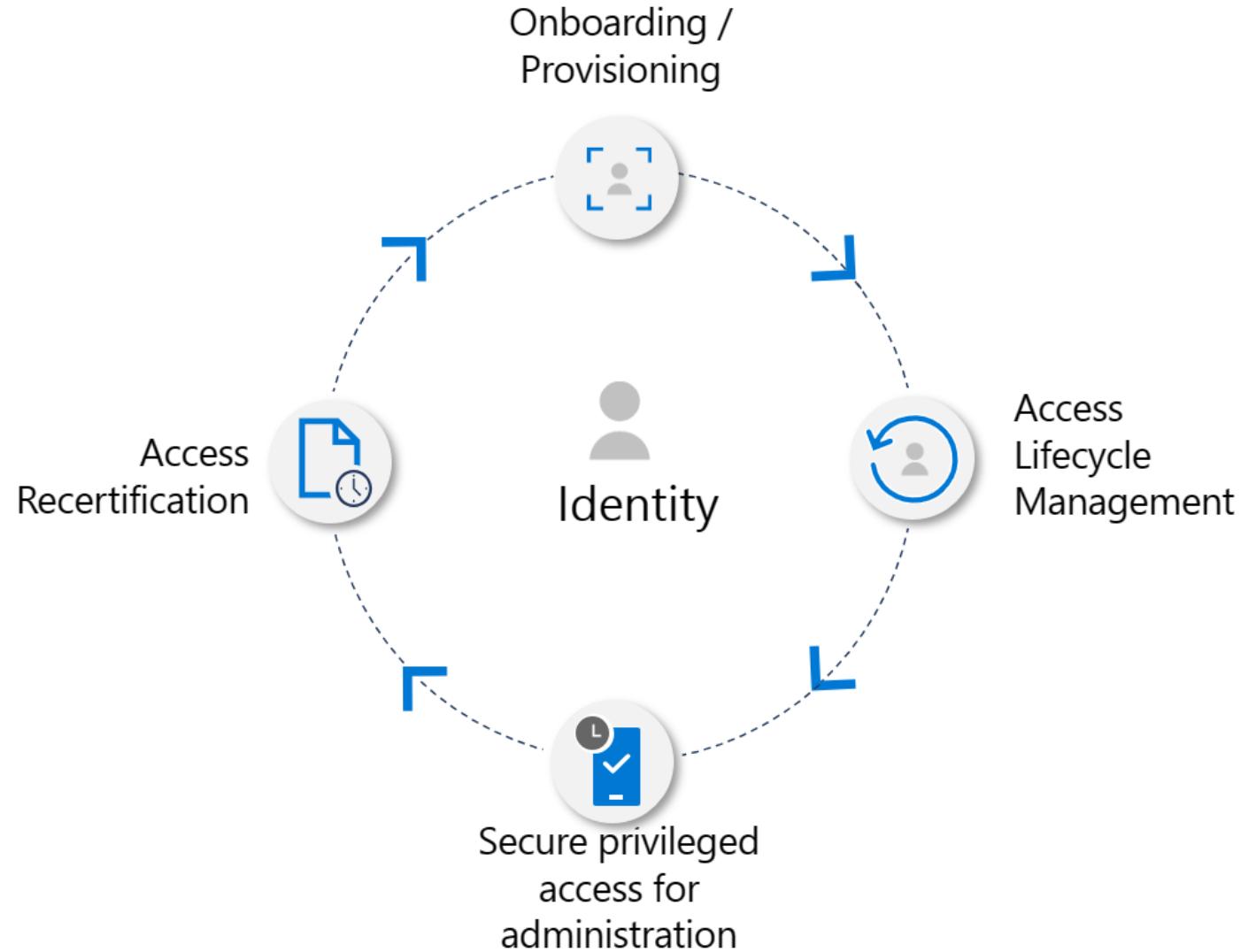
Historically, the IT staff has heavily relied on inconsistent processes to manage the creation, updating, and deletion of employees and their data within their identity providers.

Unfortunately, this traditional approach has posed several challenges, particularly when it comes to ensuring timely and accurate access provisioning.



To address these issues and streamline the employee identity management process, organizations are increasingly adopting modern solutions and automated approaches.

By leveraging advanced identity management systems and technologies, the IT staff can overcome the limitations of manual procedures and enhance efficiency.



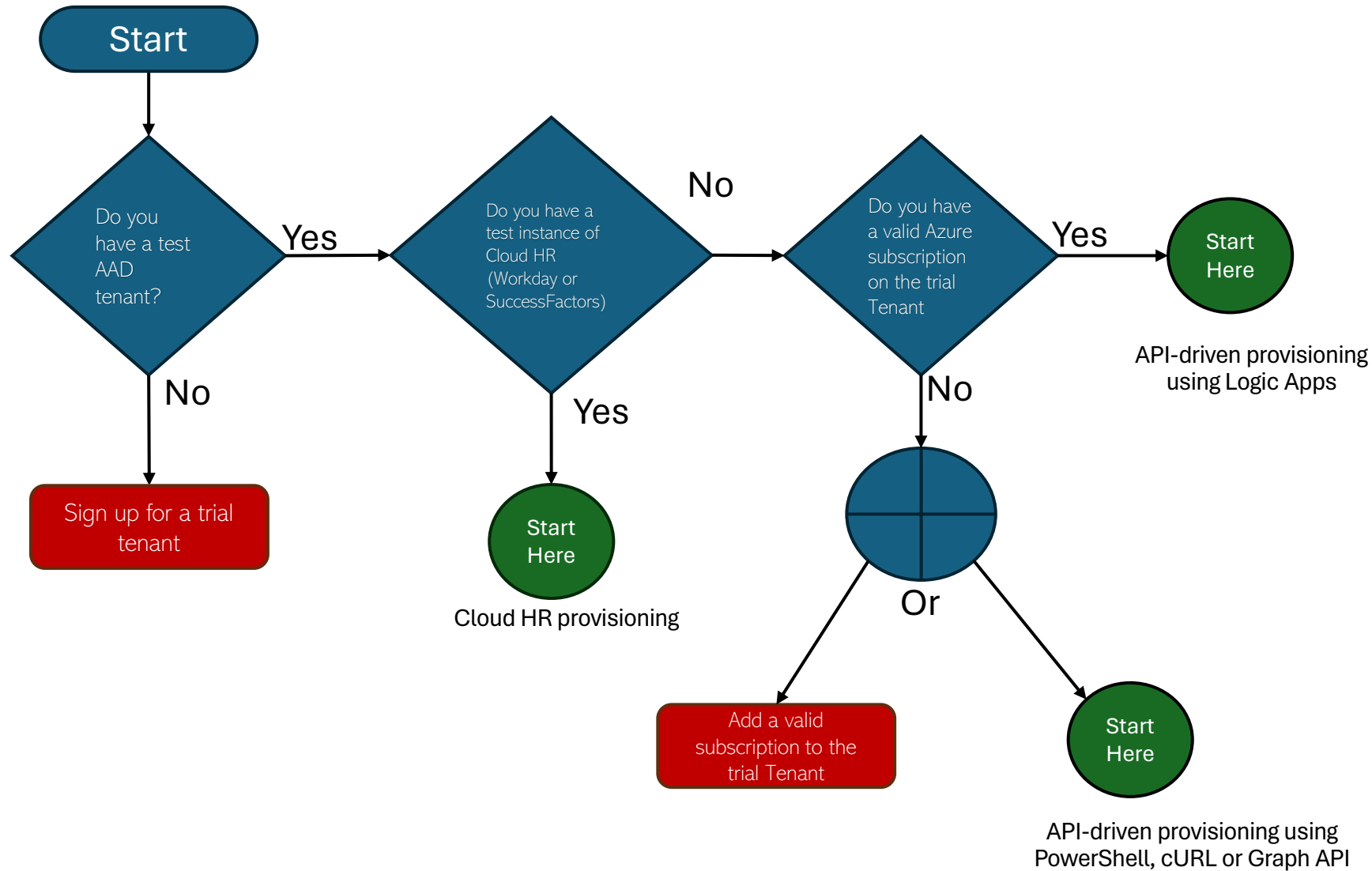
Key benefits from using Microsoft Entra ID Governance

- **Increased user productivity:** Typically, new hires undergo a two-week onboarding process to acquire access to essential resources. Microsoft Entra ID governance can this wait is significantly shortened. Through streamlined automation of the entire lifecycle, from initial provisioning to granting application access, the onboarding process becomes expedited and efficient
- **Cost savings:** Through the automation of user lifecycles and the elimination of manual or disconnected tasks, IT administrators can redirect their attention towards more critical priorities
- **Increased Security:** Streamline the automation of user departures, application access revocation, team and group removal as well as management of the lifecycle of guest accounts
- **Compliance:** Adhering to compliance regulations is crucial for organizations. Implementing a robust identity lifecycle greatly aids in meeting these requirements

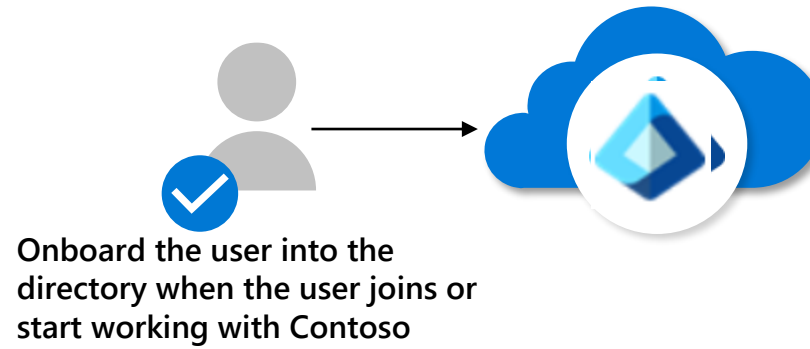
Modern Identity Management solutions

- Automatic Provisioning
- Efficient Access Management
- Centralized Identity Governance
- Self Service Capabilities
- Audit and Reporting
- Integration with HR systems

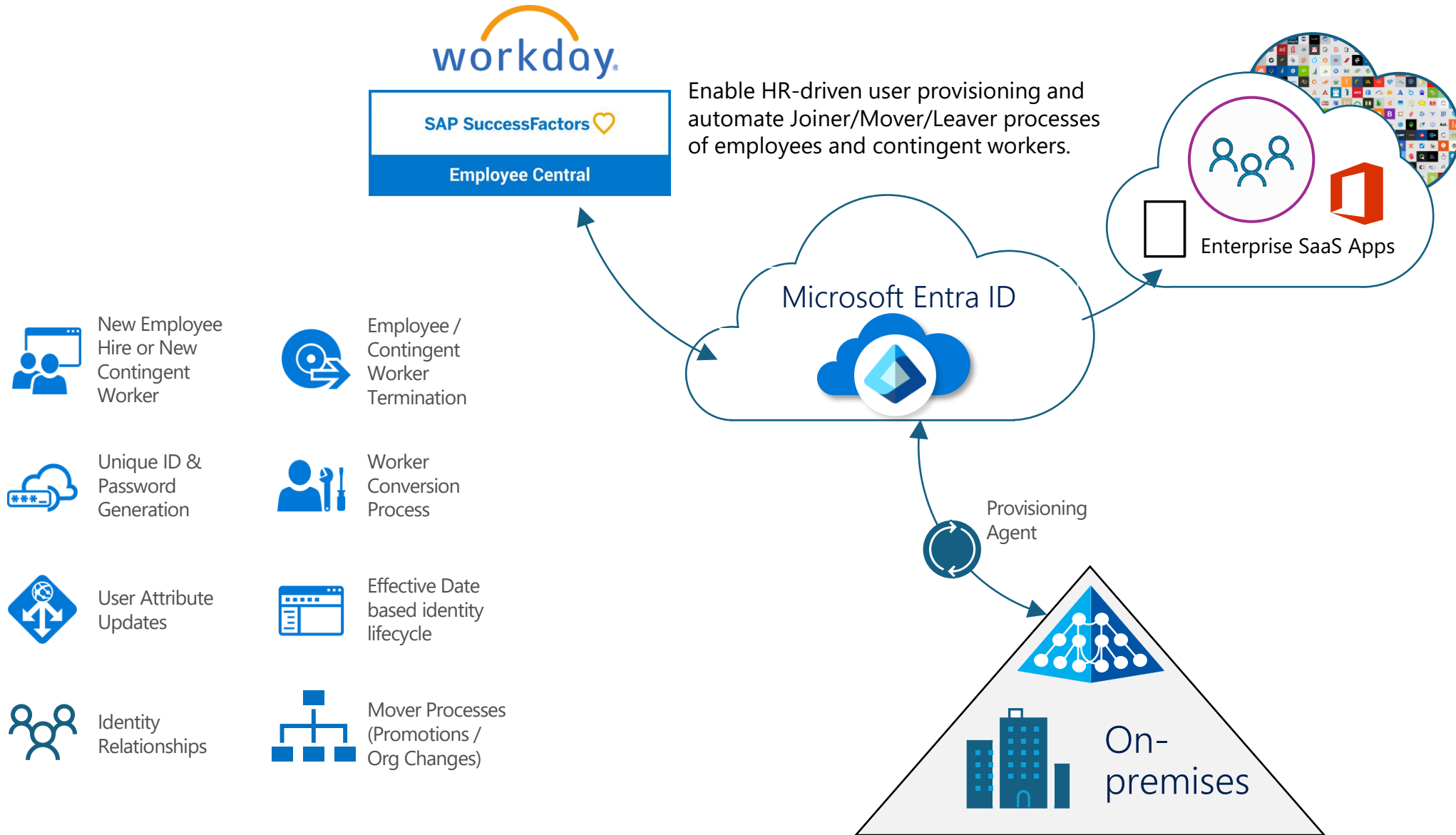
POC Decision Flow



HR Driven provisioning

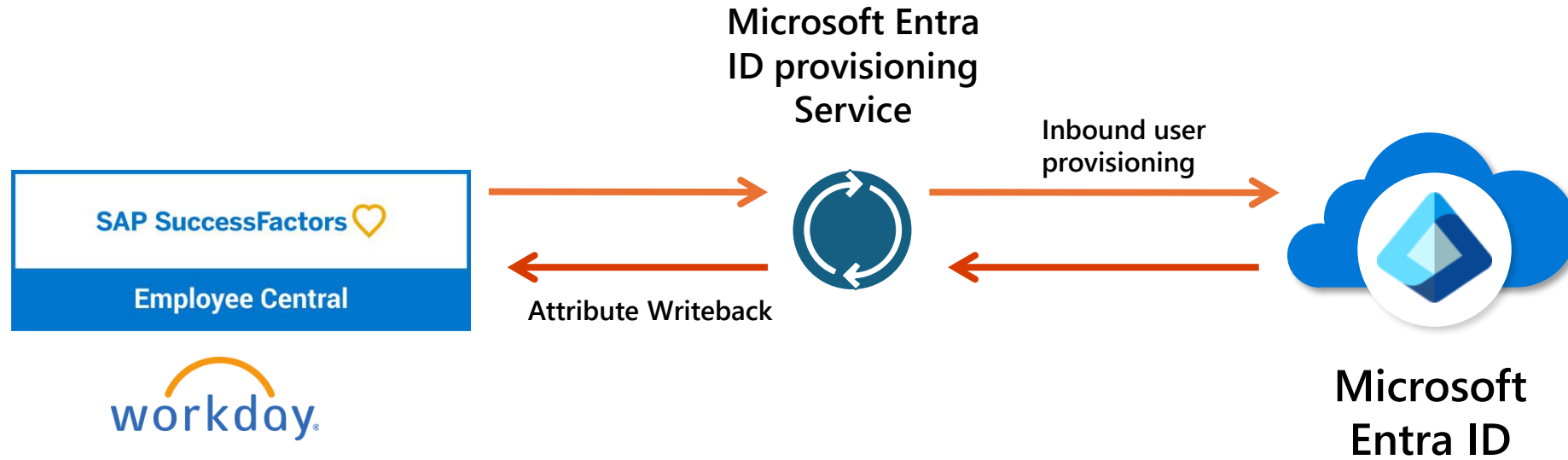


High level architecture – Cloud HR to Microsoft Entra ID / AD



Cloud HR to Microsoft Entra ID

Cloud HR to Microsoft Entra ID



Features

- **Employee is added to Cloud HR:** user should be automatically created in Microsoft Entra ID and other SaaS applications that supports user provisioning
- **Employee attribute updates:** When employee record is updated in Cloud HR , their user account in Microsoft Entra ID and supporting SaaS Applications should be automatically updated
- **Employee Terminations:** When employee is terminated in Cloud HR , their user account is automatically disabled in Microsoft Entra ID and supporting SaaS applications
- **Employee rehires:** When Employee is rehired in Cloud HR, customers should have the ability to re-activate or re-provision accounts in Microsoft Entra ID and supporting SaaS applications (depending on customer's preference)

Planning

- Determine the matching ID
 - This is an important step – since customers get the ability to match existing accounts between source and target systems
 - What is the unique Identifier in the Cloud HR app?
 - Discuss how to handle re-hires (re-use employee IDs)
 - Determine specific lifecycle cases like employees contingent workers, conversions , etc.
- Determine account status
 - Provisioning connector maps the HR user profile status to the user account status
 - Determine the Joiner / Leaver scenarios and requirements

Planning (Cont'd)

- Determine attribute mappings, transformation and scoping filters
 - How attributes will be mapped to the Microsoft Entra ID user when records are updated in Cloud HR

Deploy POC (Workday to Microsoft Entra ID)

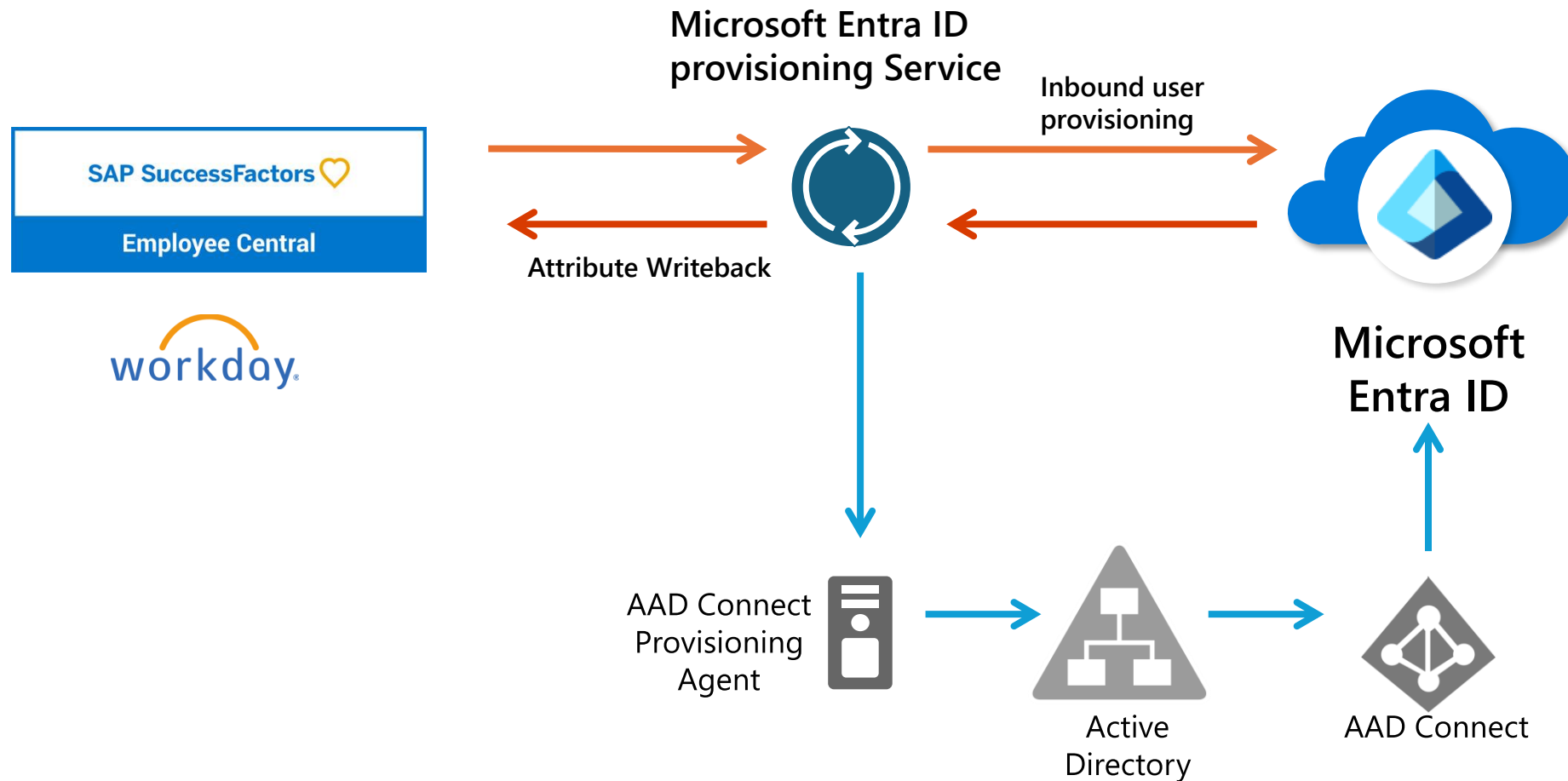
| Step | Instructions |
|--|---|
| 1. Select Cloud HR provisioning | Select cloud HR provisioning connector apps |
| 2. Design Provisioning topology | Design provisioning topology |
| 3. Create integration system | Configure integration system user in Workday |
| 4. Add Workday Enterprise App | Workday provisioning connector |
| 5. Configure Workday mappings | Workday and Azure AD attribute mapping |
| [Optional] Configure Workday Writeback | Configure Workday writeback in Azure Active Directory |
| 6. Enable and Launch provisioning | Enable and launch provisioning |

Deploy POC (SuccessFactors to Microsoft Entra ID)

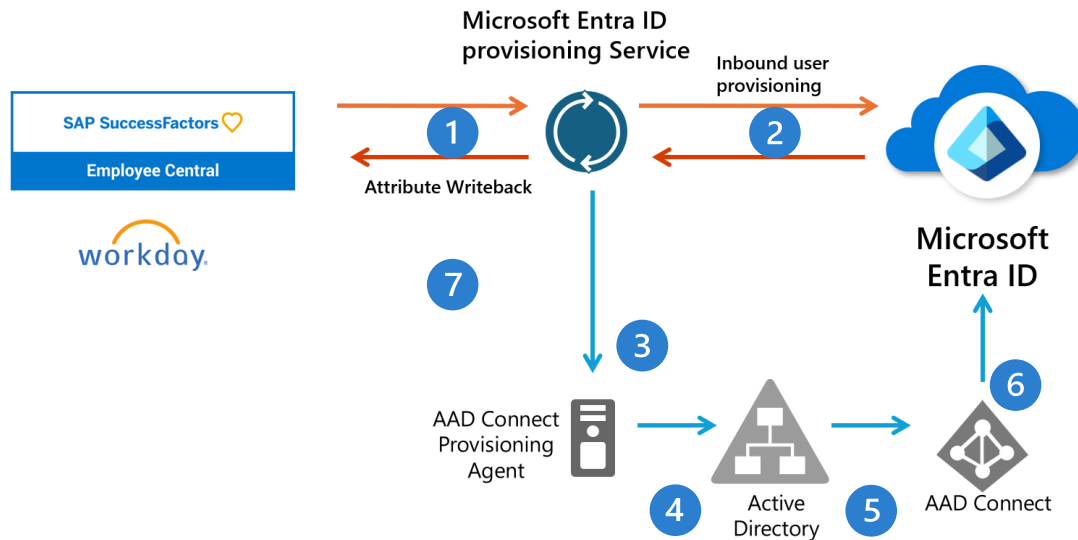
| Step | Instructions |
|---|--|
| 1. Select Cloud HR provisioning | Select cloud HR provisioning connector apps |
| 2. Design Provisioning topology | Design provisioning topology |
| 3. Create integration API | Create API user account in SuccessFactors |
| 4. Create permissions for API | Create API permissions in SuccessFactors |
| 5. Add SuccessFactors provisioning connector | Add SuccessFactors inbound connector app |
| 6. Configure Attribute mappings | Configure SuccessFactors attribute mappings |
| [Optional] Configure SuccessFactors Writeback | Configure attribute write-back from Azure AD to SAP SuccessFactors |
| 7. Enable and launch provisioning | Enable and Launch provisioning |

Cloud HR to Active Directory

Cloud HR to Active Directory



Cloud HR to Active Directory



User data Flow:

1. The HR team performs worker transactions (Joiners/Movers/Leavers or New Hires/Transfers/Terminations) in Cloud HR
2. The Microsoft Entra ID Provisioning Service runs scheduled synchronizations of identities from CloudHR and identifies changes that need to be processed for sync with on-premises Active Directory.
3. The Microsoft Entra ID Provisioning Service invokes the on-premises Microsoft Entra ID Connect Provisioning Agent with a request payload containing AD account create/update/enable/disable operations.
4. The Microsoft Entra ID Connect Provisioning Agent uses a service account to add/update AD account data.
5. The Microsoft Entra ID Connect / AD Sync engine runs delta sync to pull updates in AD.
6. The Active Directory updates are synced with Microsoft Entra ID
7. If the CloudHR Writeback app is configured, it writes back attributes such as email, username and phone number to CloudHR.

Features

- **Employee is added to Cloud HR:** user should be automatically created in Active Directory, Microsoft Entra ID and other SaaS applications that supports user provisioning
- **Employee attribute updates:** When employee record is updated in Cloud HR , their user account in Active Directory, Microsoft Entra ID and supporting SaaS Applications should be automatically updated
- **Employee Terminations:** When employee is terminated in Cloud HR , their user account is automatically disabled in Active Directory, Microsoft Entra ID and supporting SaaS applications
- **Employee rehires:** When Employee is rehired in Cloud HR, customers should have the ability to re-activate or re-provision accounts in Active Directory, Microsoft Entra ID and supporting SaaS applications (depending on customer's preference)

Planning

- Setup of the Microsoft Entra ID Connect provisioning agent
- Number of Workday to AD user provisioning apps to deploy
- Selecting the right matching identifier, attribute mapping, transformation and scoping filters

Deploy POC (Workday to Active Directory)

| Step | Instructions |
|--|--|
| 1. Select Cloud HR provisioning | Select cloud HR provisioning connector apps |
| 2. Design Provisioning topology | Design provisioning topology |
| 3. Create integration system | Configure integration system user in Workday |
| 4. Configure Workday provisioning App and connectors | Provisioning connector app and Provisioning Agent |
| 5. Install and configure Agents | Install and configure on-prem Agents |
| 6. Configure app connectivity to AD | Configure connectivity to Workday and Active Directory |
| 7. Configure Attribute mappings | Configure attribute mappings |
| 8. Enable and launch provisioning | Enable and launch user provisioning |

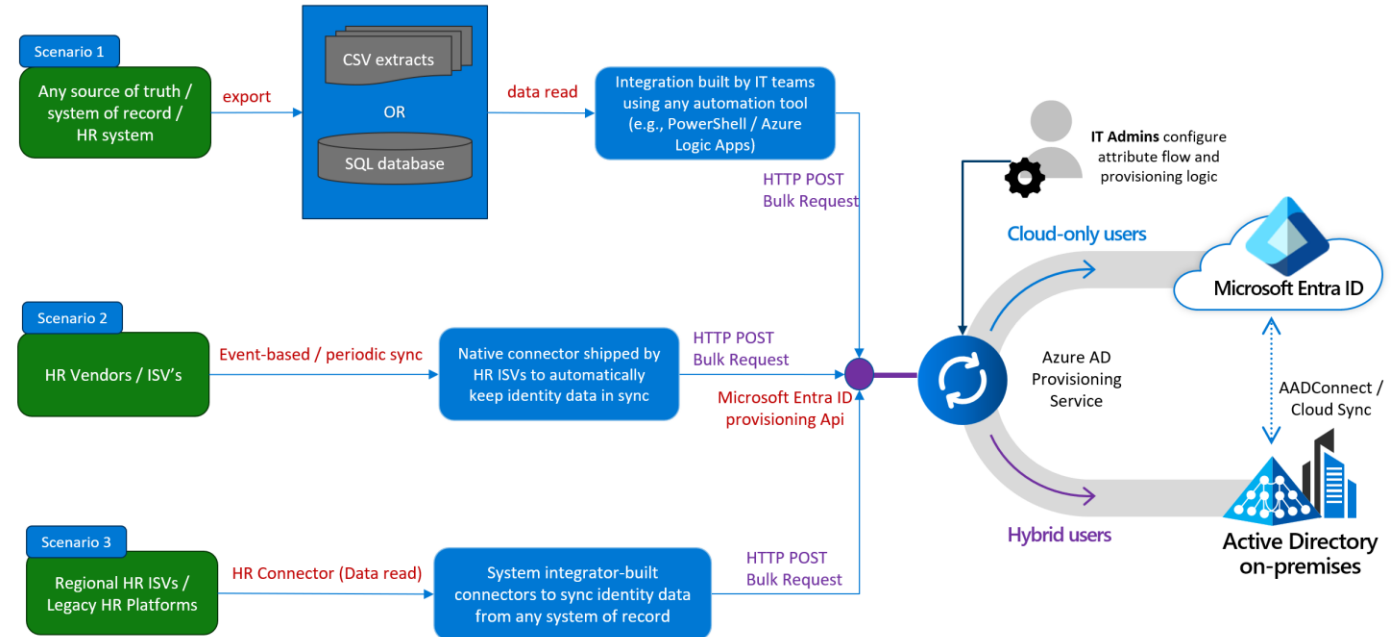
Deploy POC (SuccessFactors to Active Directory)

| Step | Instructions |
|---|---|
| 1. Select Cloud HR provisioning | Select cloud HR provisioning connector apps |
| 2. Design Provisioning topology | Design provisioning topology |
| 3. Create integration API | Create API user account in SuccessFactors |
| 4. Configure SuccessFactors provisioning App and connectors | SuccessFactors inbound provisioning app and agent |
| 5. Install and configure Agents | Install on-prem agents |
| 6. Configure app connectivity to AD | Configure App connectivity to AD |
| 7. Configure Attribute mappings | Configure attribute mappings |
| 8. Enable and launch provisioning | Enable and launch user provisioning |

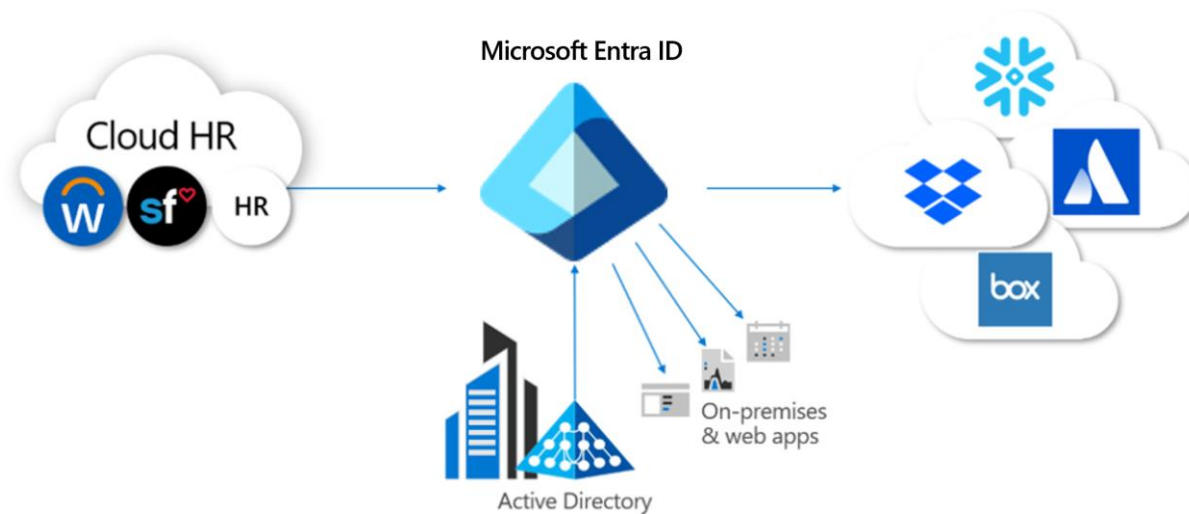
API-driven inbound user provisioning

What if my organization doesn't use Workday or SuccessFactors?

You can use Microsoft Entra ID API-driven provisioning



Key Benefits of API-driven user provisioning



Provisioning app in Microsoft Entra ID that uses a valid OAuth token

Admin control through permissions consent

Graph API endpoint that uses SCIM schema constructs.

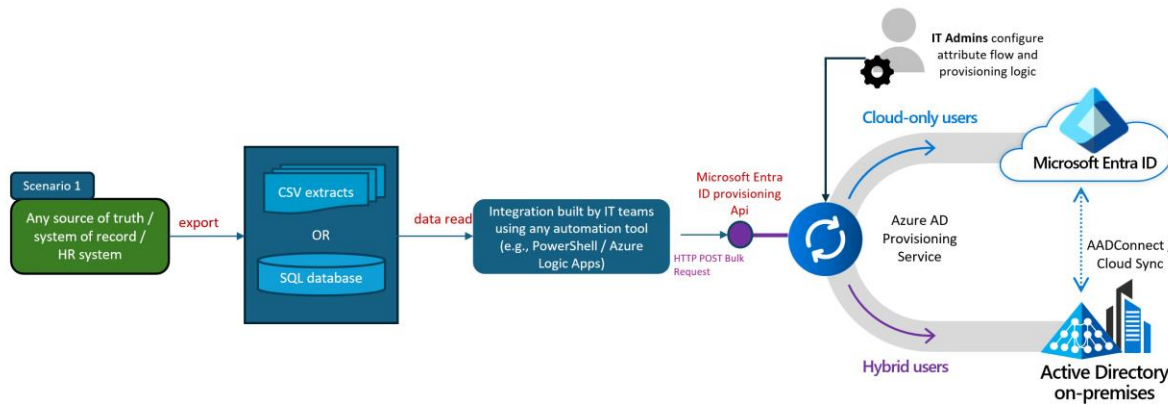
Can be integrated with multiple data sources

Near real-time Incoming bulk request processing

Admins can check progress through provisioning and API logs

Microsoft Entra ID – API-driven provisioning

Scenario 1: Enable IT teams to import HR data extracts using any automation tool

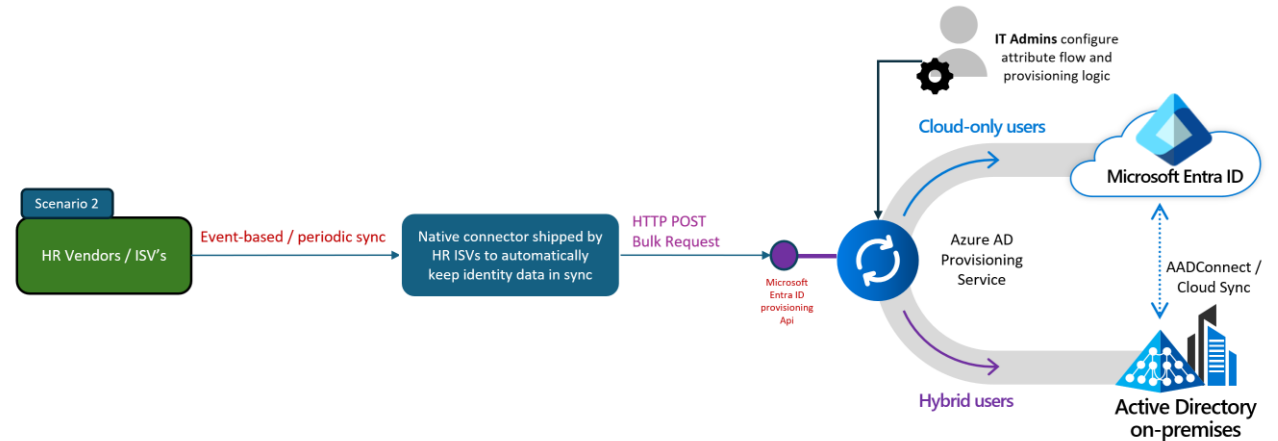


- Commonly used for sources like:
 - Flat files
 - CSV files
 - SQL staging tables
- Integration with any automation tool like:
 - PowerShell scripts
 - Azure Logic Apps
 - Workflows using HTTP calls

Microsoft Entra ID – API-driven provisioning

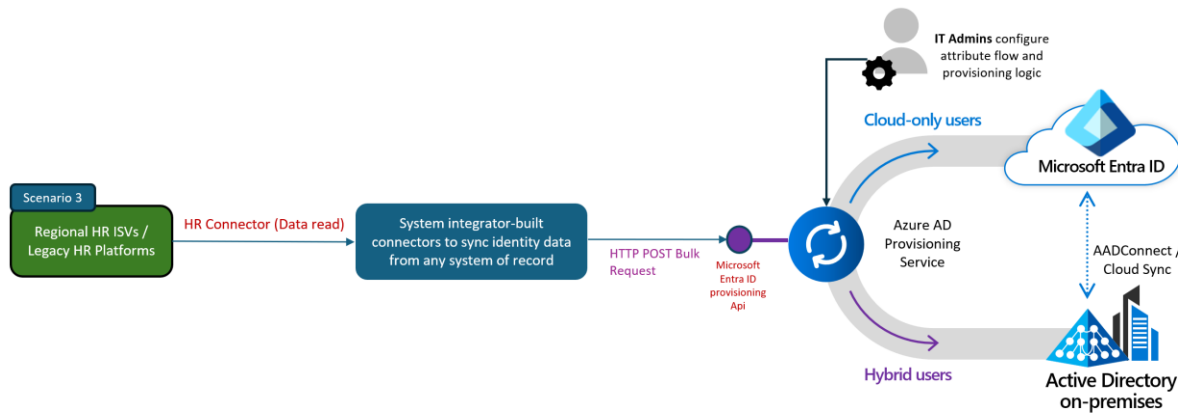
Scenario 2: Enable Independent Software Vendors (ISV) to build direct integration with Microsoft Entra ID

- Enables HR ISVs to ship native synchronization experiences
- Changes in the origin HR system will automatically flow into Microsoft Entra ID and Active Directory domains



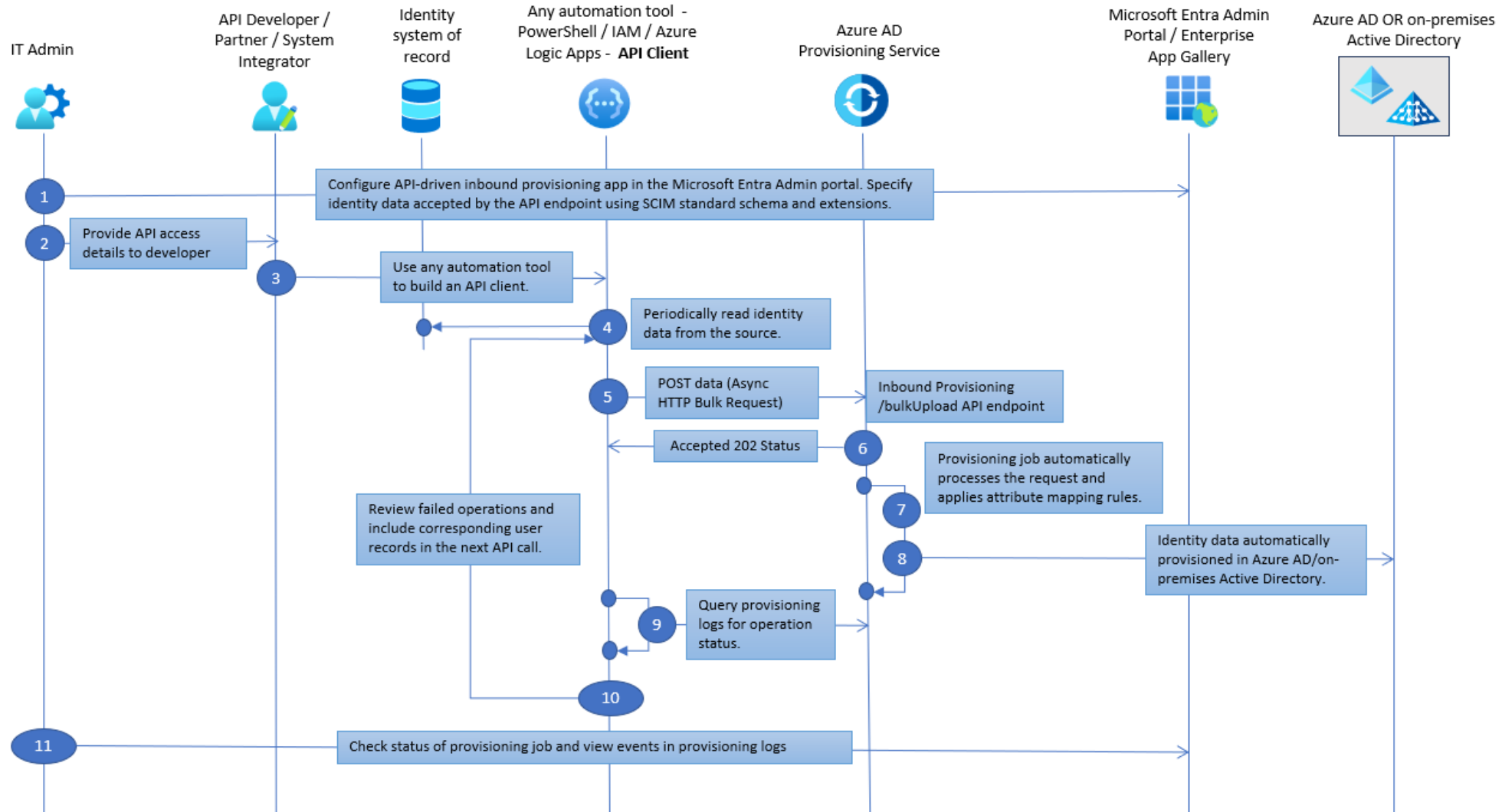
Microsoft Entra ID – API-driven provisioning

Scenario 3: Enable system integrators to build more connectors to systems of record



- Partners can build custom HR connectors
- Used to meet different requirements around data flows

End to End flow



How is the new inbound provisioning /bulkUpload API different from MS Graph Users API?

There are significant differences between the provisioning /bulkUpload API and the MS Graph Users API endpoint.

Payload format:

- The MS Graph Users API endpoint expects data in OData format. The request payload format for the new inbound provisioning /bulkUpload API uses SCIM schema constructs. When invoking this API, set the 'Content-Type' header to application/scim+json.

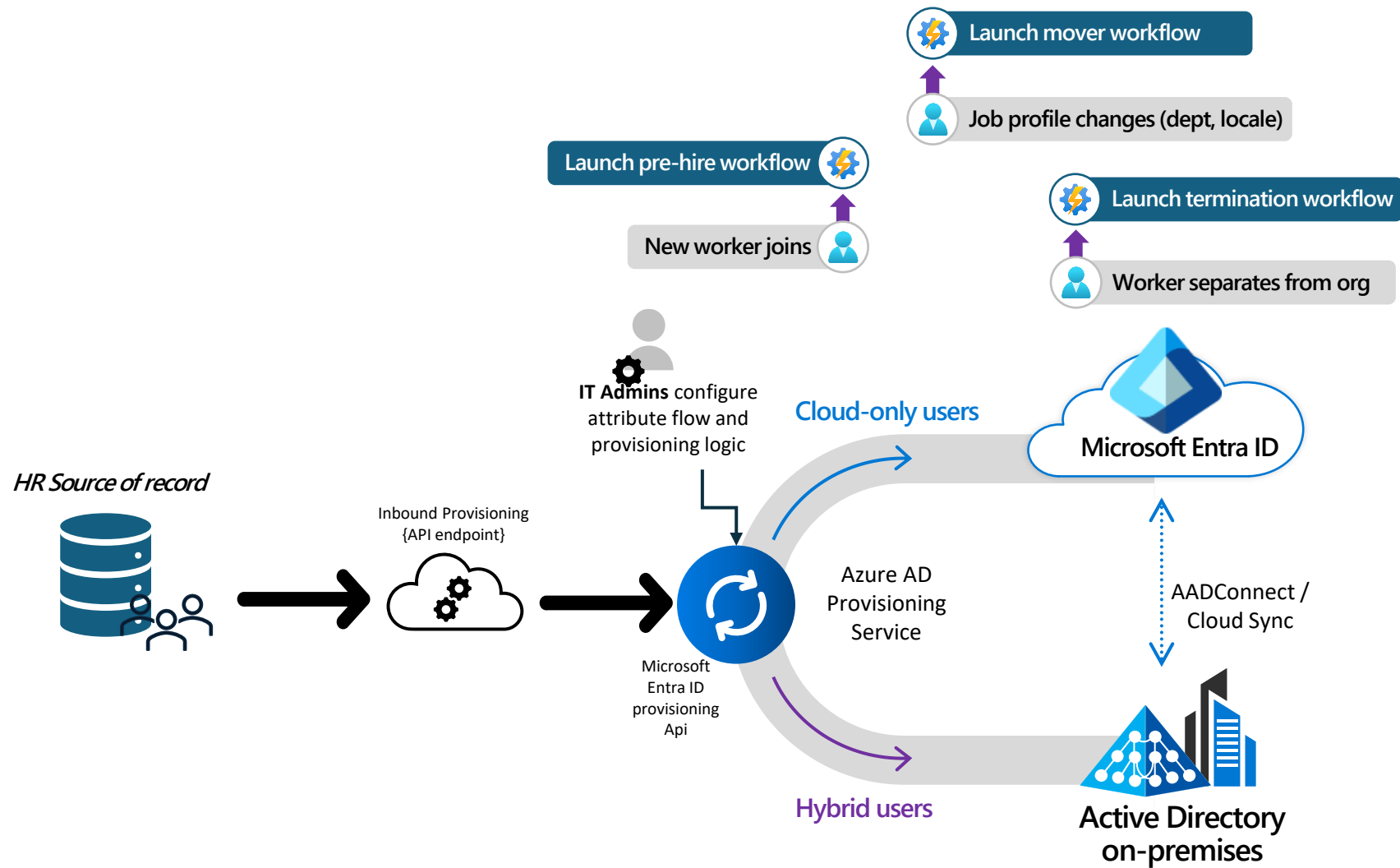
Operation end-result:

- When identity data is sent to the MS Graph Users API endpoint, it's immediately processed, and a Create/Update/Delete operation takes place on the Azure AD user profile.
- Request data sent to the provisioning /bulkUpload API is processed asynchronously by the Azure AD provisioning service. The provisioning job applies scoping rules, attribute mapping and transformation configured by the IT admin. This initiates a Create/Update/Delete operation on the Azure AD user profile or the on-premises AD user profile.

IT admin retains control:

- With API-driven inbound provisioning, the IT admin has more control on how the incoming identity data is processed and mapped to Azure AD attributes. They can define scoping rules to exclude certain types of identity data (for example, contractor data) and use transformation functions to derive new values before setting the attribute values on the user profile.

API-driven user provisioning and Lifecycle Workflows



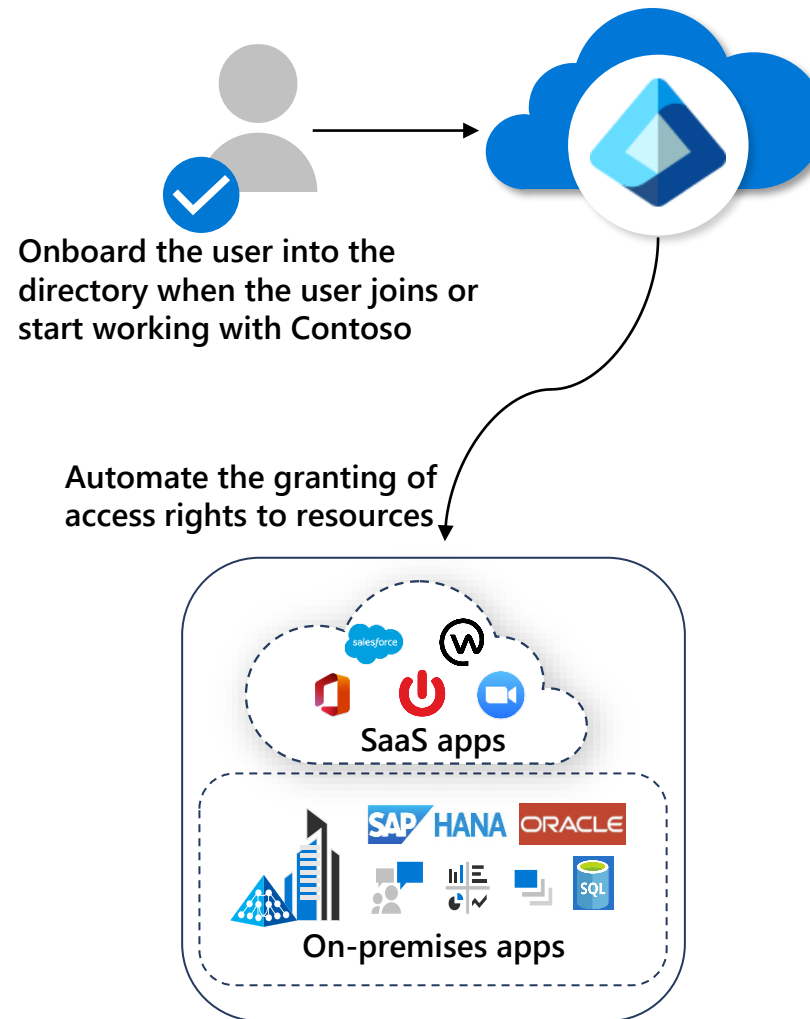
Deploy POC API-driven inbound provisioning

| Step | Instructions |
|---|---|
| 1. Create API-drive provisioning app | API-driven provisioning app |
| 2. Configure API-driven inbound provisioning | AD - Configure API-driven inbound provisioning app AD Azure AD - Configure API-driven inbound provisioning app - AAD |
| 3. Grant Access to the inbound provisioning API | Grant access to inbound provisioning API |
| [Optional] Customize attribute mappings | Customize user provisioning attribute mappings |
| [Optional] Sync custom attributes | Sync custom attributes |
| 4. Go to next slide for QuickStart samples | Next Slide |

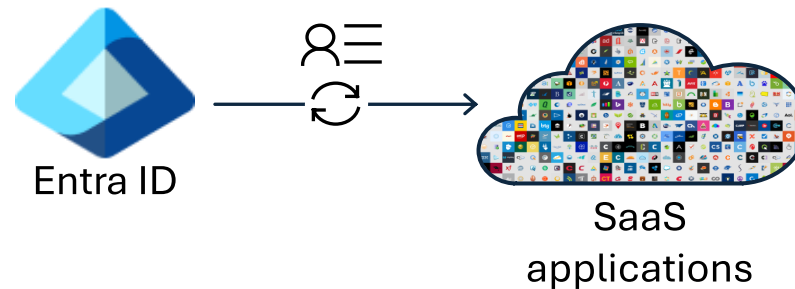
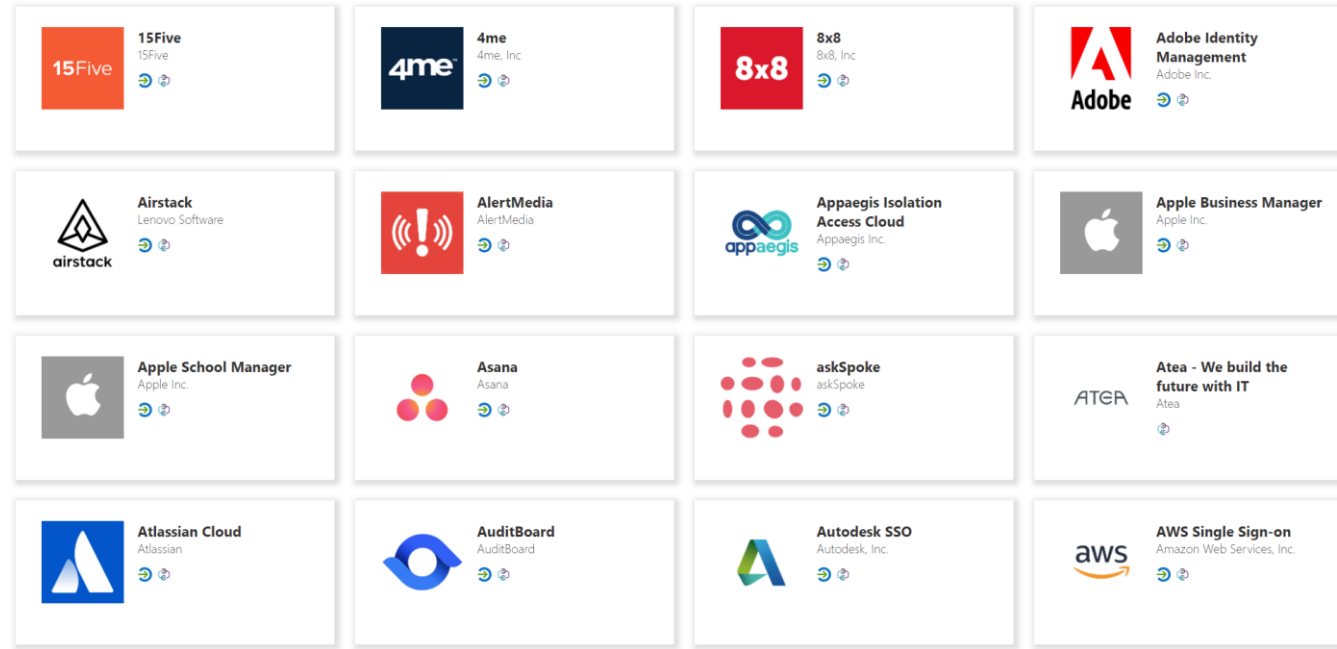
QuickStart guides

| Title | Link |
|------------------|---|
| cURL | Quickstart API-driven inbound provisioning with cURL |
| Postman | Quickstart API-driven inbound provisioning with Postman |
| Graph Explorer | Quickstart API-driven inbound provisioning with Graph Explorer |
| PowerShell | API-driven inbound provisioning with PowerShell |
| Azure Logic Apps | API-driven inbound provisioning with Azure Logic Apps |
| FAQ | Frequently asked questions (FAQs) API-driven inbound provisioning |

Provisioning user to Apps



Provisioning to SaaS apps using SCIM

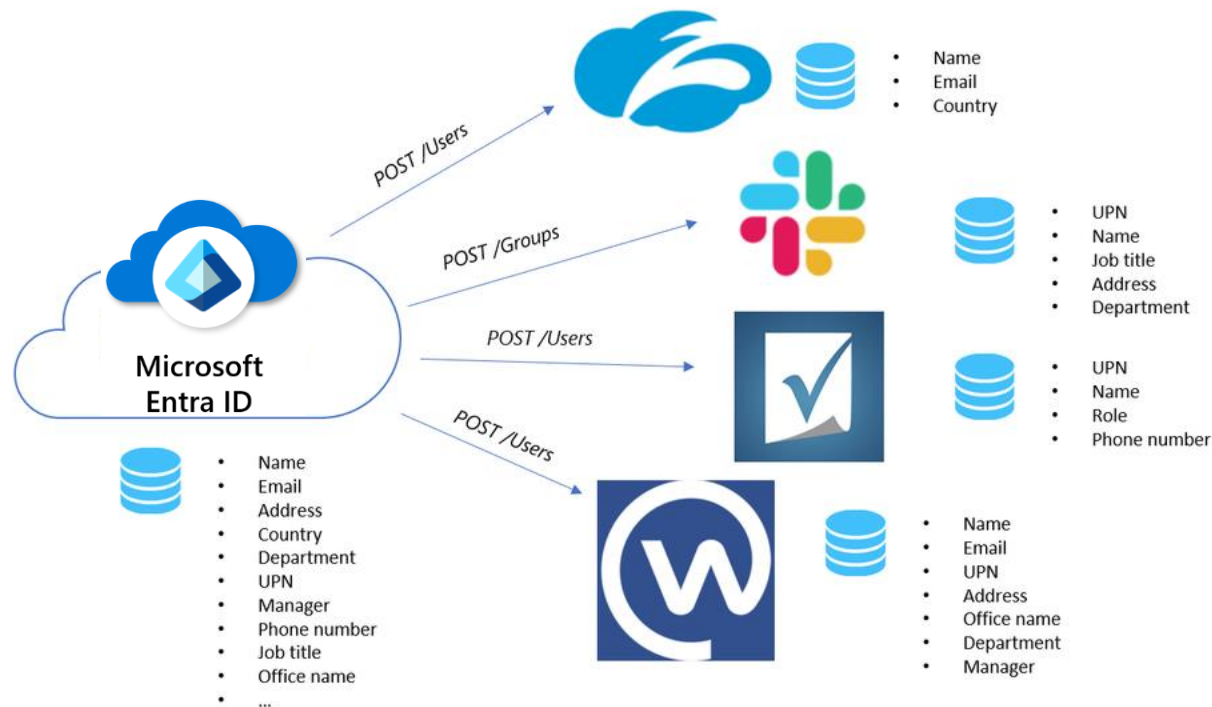


Features

- Automate provisioning
- Automate deprovisioning
- Synchronize data between systems
- Provision groups
- Govern access
- Seamlessly deploy in brown field scenarios
- Use rich customization
- Get alerts for critical events



Configure provisioning with SCIM endpoint



- SCIM 2.0 is a standardized definition of two endpoints */Users* and */Groups*

Deploy POC SaaS sample app provisioning

Select up to 4 apps

| Step | Instructions |
|-----------------------|---|
| Service Now | Configure ServiceNow for automatic user provisioning |
| Salesforce | Configure Salesforce for automatic user provisioning |
| Box | Configure Box for automatic user provisioning |
| Cisco Webex | Configure Cisco Webex for automatic user provisioning |
| Workplace by Facebook | Configure Workplace by Facebook for automatic user provisioning |
| Zoom | Configure Zoom for automatic user provisioning |

Optional: Provisioning to on-premises applications

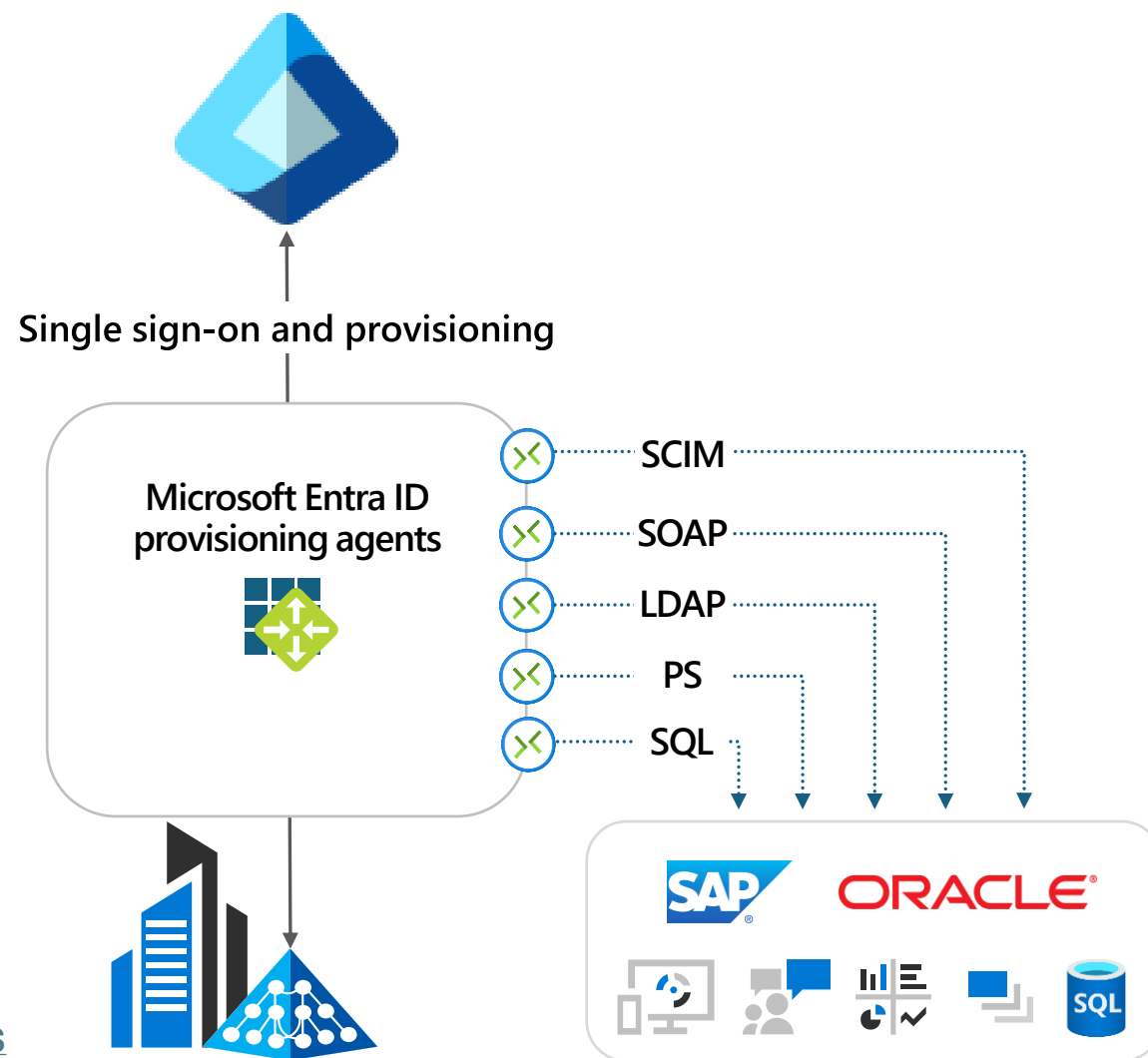
Users and schema defined in the cloud

- Supports provisioning from custom schema extensions to app-specific properties.

Translation to the provisioning protocols expected by apps

- Microsoft-delivered connectors: LDAP, SQL, etc.
- Ecosystem of third-party connectors for other apps requiring custom API integrations
- Customers can re-use their existing MIM configuration

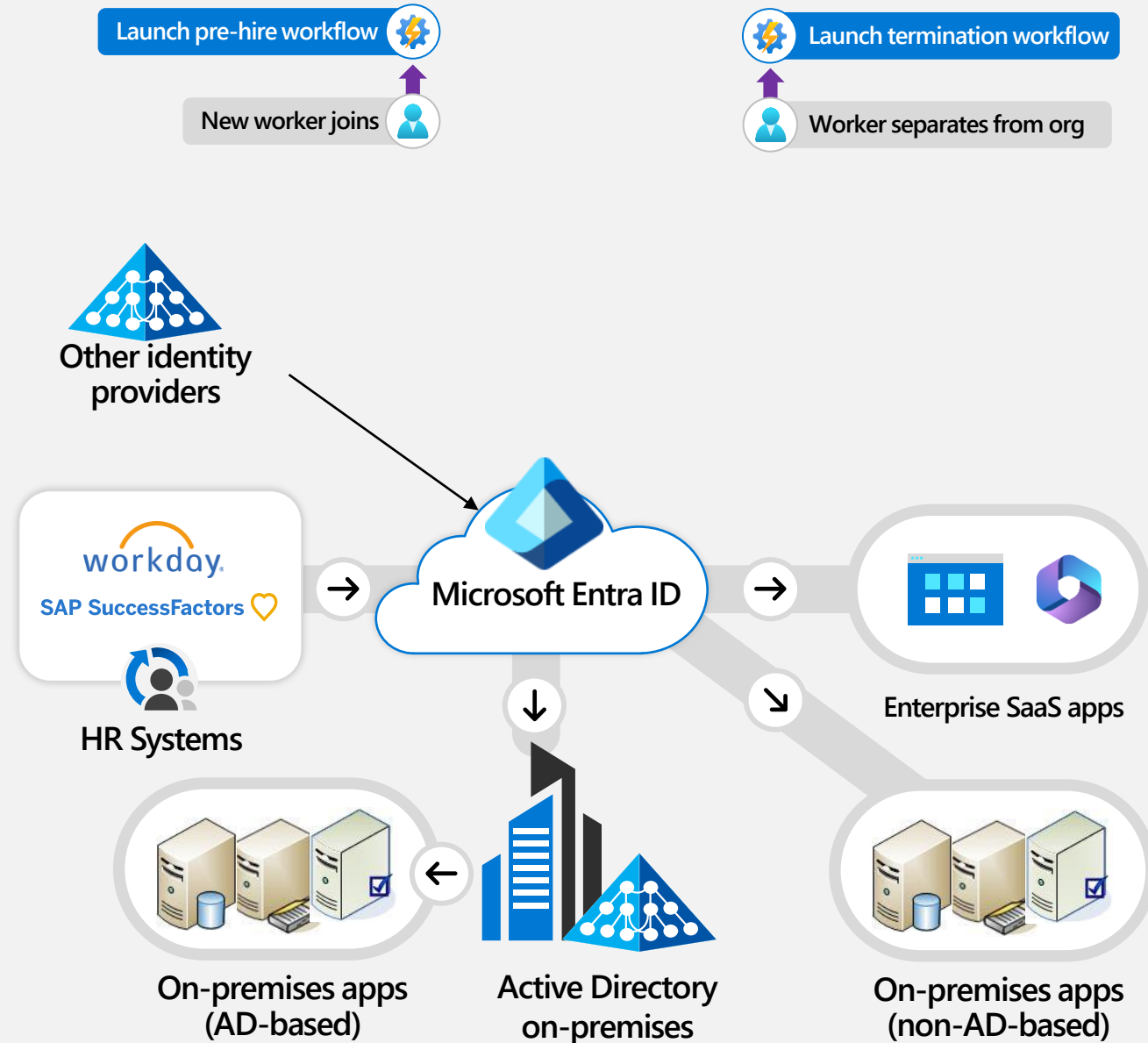
Samples: [App provisioning samples for SCIM-enabled apps](#)



Lifecycle Workflows

Automate join/move/leave employee lifecycle events

- Organizations can schedule tasks to occur before, at or after a join or leave date; these can also be run on-demand.
- Built-in tasks include generating temporary credentials, sending emails, updating user attributes, and memberships, and removing licenses.
- Customers and partners can extend lifecycle workflows with additional tasks via Azure Logic Apps.

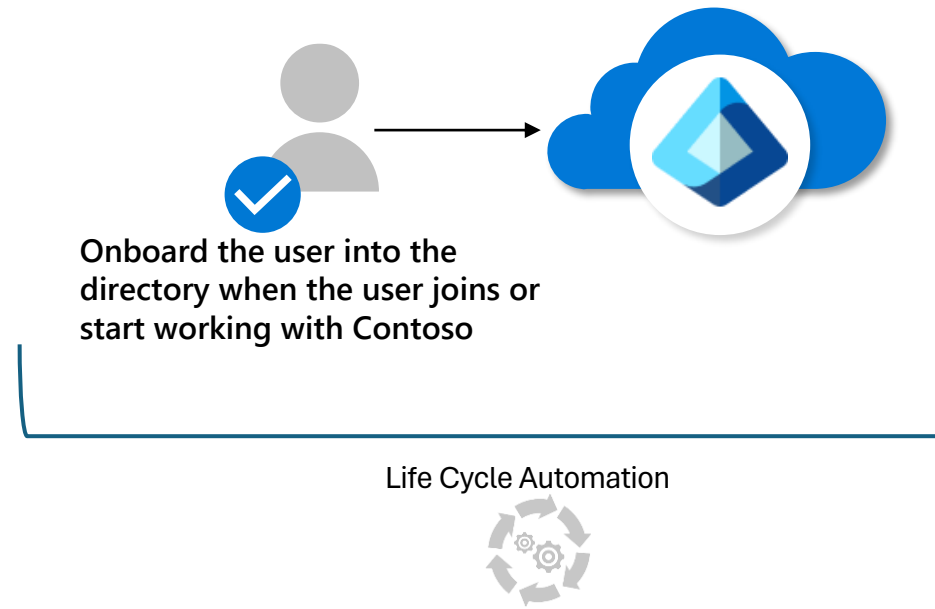


Synchronize relevant attributes for scheduling triggers

First things first

- **This can be done using things like:**
 - Cloud HR provisioning
 - Custom import from Active Directory (Sync)
 - Microsoft Graph (cloud users)
 - Visit : [How to synch attributes for LCW](#) for more information
- **Attributes needed for this workflow:**
 - employeeHireDate
 - createdDateTime
 - employeeLeaveDateTime
 - Manager (for email to manager tasks)

Onboarding Automation



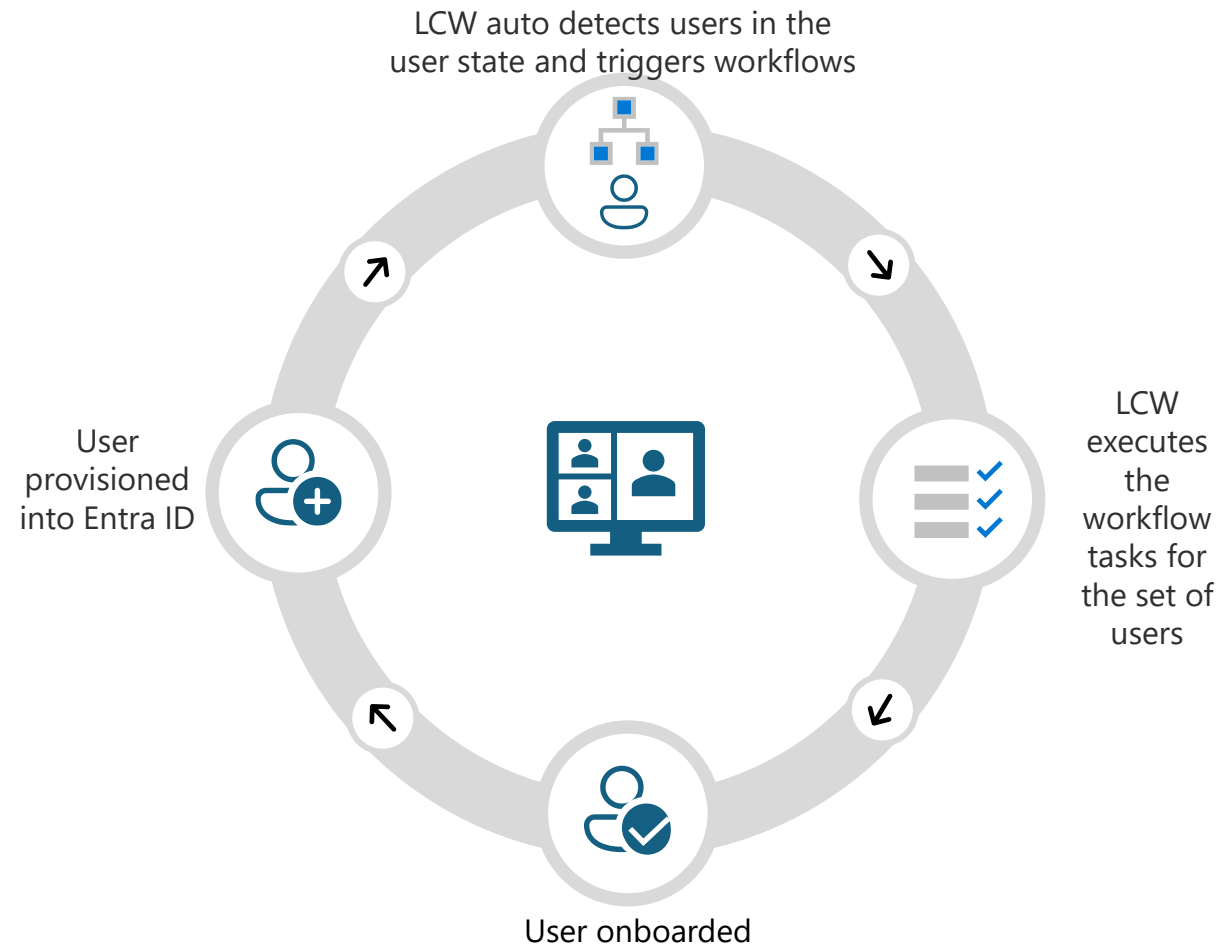
Joiner scenario

Onboarding new users

Before Lifecycle Workflows

It can take days for new team members to be fully productive while access to resources is being provisioned.

Onboarding users manually is time consuming for IT and often requires admins to jump between apps and scripts and go through logs in order to find errors.



With Lifecycle Workflows

“Rinse and repeat” templates and automated actions through workflows makes the process more efficient and infallible for IT admins, and faster for new team members.

Joiner scenario - Example



For pre-hire, "X" days before employee's start date,

Joiner · PRE-HIRE



Create user account (status=disabled)

'X' days before start



Send email to onboarding DL



Launch custom Logic Apps workflow



Generate Temporary Access Pass (TAP)



Send email to hiring manager with TAP



On the employee's start date, perform the following tasks.

Joiner · HIRE



On start date

On start date



Enable user account



Group assignments



Send welcome email to new hire



Add user to Teams "New Hires" channel



Employee's day 1 and Manager shares the temporary access pass.

Joiner · WELCOME



On start date

On Day 1



Login with temporary access pass



Setup passwordless sign-in Windows Hello / FIDO key



Welcome messages in Teams channel



"Best onboarding experience ever!"

Lifecycle workflows built-in templates

Choose a workflow

Choose a workflow template to start creating your custom workflow. [Learn more](#)

Joiner

Onboard pre-hire employee

Configure pre-hire tasks for onboarding employees before their first day

[Select](#) | [Details](#)

Joiner

Onboard new hire employee

Configure new hire tasks for onboarding employees on their first day

[Select](#) | [Details](#)

Joiner

Post-Onboarding of an employee

Configure onboarding tasks for an employee after their first day of work

[Select](#) | [Details](#)

Mover On-demand

Real-time employee change

Execute real-time tasks for employee job changes

[Select](#) | [Details](#)

Leaver On-demand

Real-time employee termination

Execute real-time termination tasks for employees on their last day of work

[Select](#) | [Details](#)

Leaver

Pre-Offboarding of an employee

Configure pre-offboarding tasks for employees before their last day of work

[Select](#) | [Details](#)

Leaver

Offboard an employee

Configure offboarding tasks for employees on their last day of work

[Select](#) | [Details](#)

Leaver

Post-Offboarding of an employee

Configure offboarding tasks for employees after their last day of work

[Select](#) | [Details](#)

Planning – Define Parameters

Onboarding Pre-hire employee customization

- Tasks to be completed by the workflow (Generate TAP and send email)
- Trigger date - How many days before the event (EmployeeHireDate)
- Execution conditions (Rule based on attributes)

Onboarding New-hire employee customization

- Tasks to be completed by the workflow (Add User To Group, Enable User Account, Send Welcome Email)
- Event User attribute (EmployeeHireDate, createdDateTime)
- Execution conditions (Rule based on attributes)

Post-onboarding employee customization

- Tasks to be completed by the workflow (Add User To Group, Add user to selected teams)
- Event User attribute (EmployeeHireDate, createdDateTime)
- Trigger date - How many days after the event (start date)
- Execution conditions (Rule based on attributes)

Employee Change (Mover)

Make sure that tasks
are completed
immediately when an
employee changes
roles in Contoso

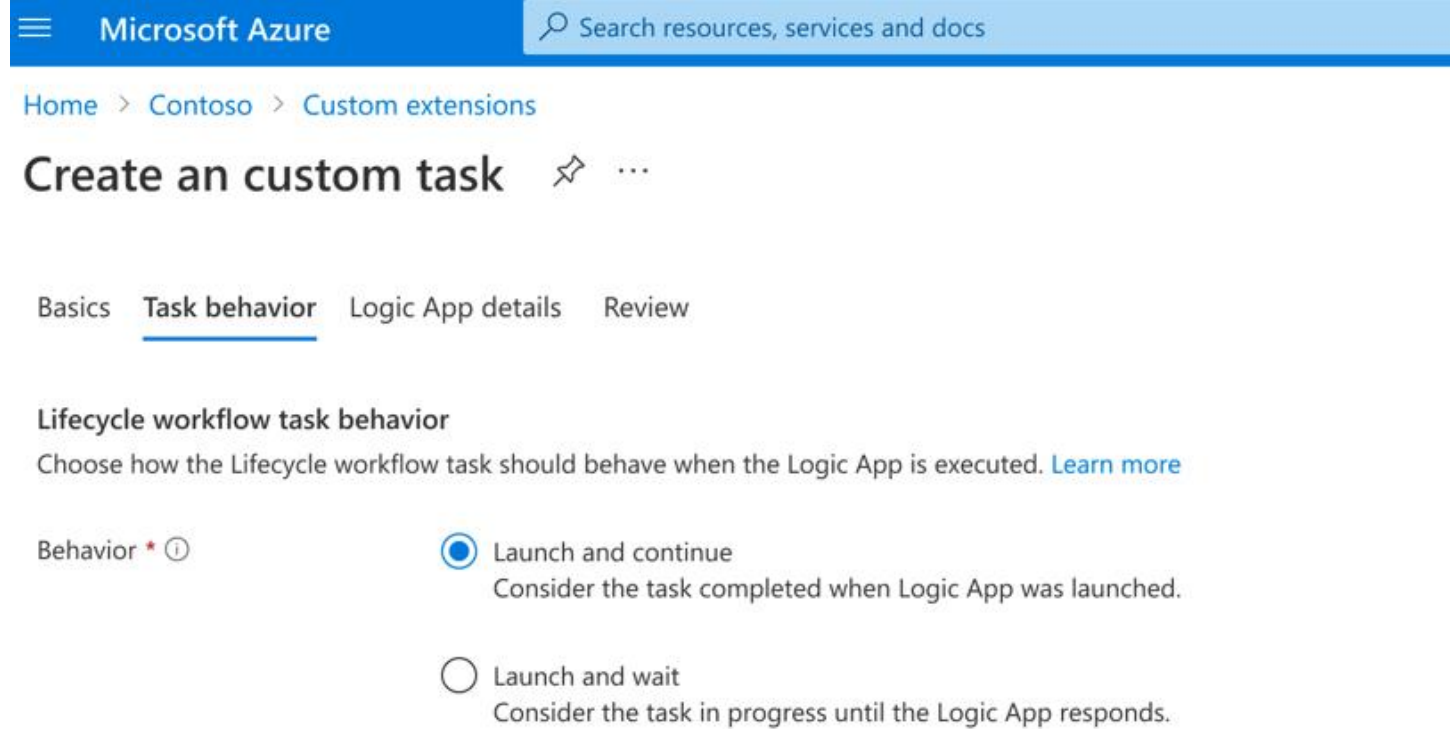


Life Cycle Automation



Lifecycle Workflows custom task extension

Utilize these extensions to call-out external systems with Logic apps as part of the workflow



The screenshot shows the Microsoft Azure portal interface. At the top, there is a blue header with the Microsoft Azure logo and a search bar. Below the header, the breadcrumb navigation shows 'Home > Contoso > Custom extensions'. The main heading is 'Create an custom task', followed by a star icon and a three-dot menu icon. Below this, there are four tabs: 'Basics', 'Task behavior' (which is selected and underlined), 'Logic App details', and 'Review'. The section title is 'Lifecycle workflow task behavior', followed by a description: 'Choose how the Lifecycle workflow task should behave when the Logic App is executed.' and a 'Learn more' link. Under the 'Behavior' section, there are two radio button options: 'Launch and continue' (which is selected) and 'Launch and wait'. Each option has a descriptive text below it.

Microsoft Azure

Search resources, services and docs

Home > Contoso > Custom extensions

Create an custom task

Basics Task behavior Logic App details Review

Lifecycle workflow task behavior

Choose how the Lifecycle workflow task should behave when the Logic App is executed. [Learn more](#)

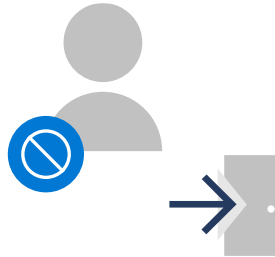
Behavior ⓘ

☒ Launch and continue
Consider the task completed when Logic App was launched.

☐ Launch and wait
Consider the task in progress until the Logic App responds.

Employee Termination

Make sure access rights are removed when user leaves or stops working with Contoso



Life Cycle Automation



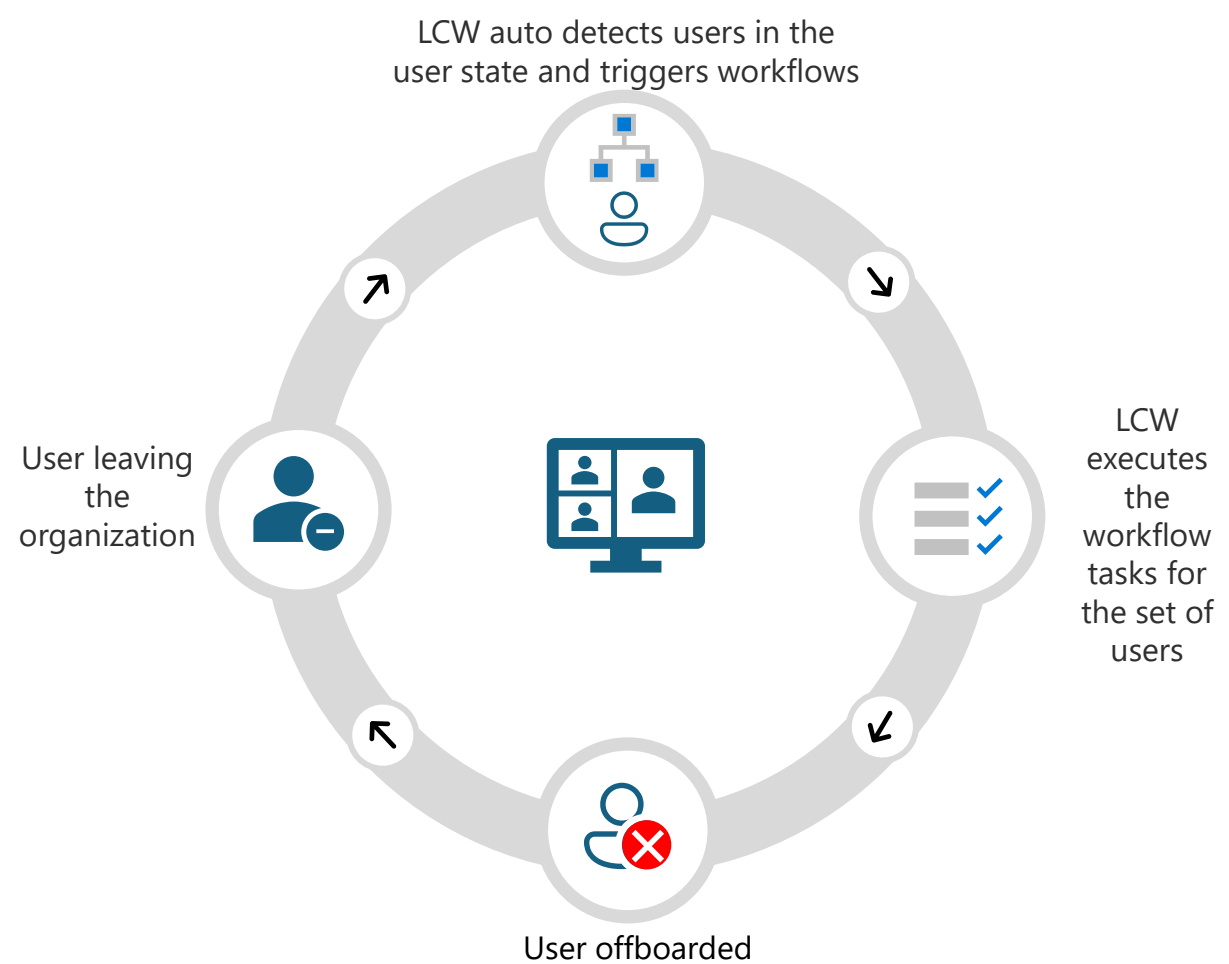
Leaver scenario

Offboarding users

Before Lifecycle Workflows

When team members leave the organization, their personal information must be handled in compliance with regulations and policies, it is also important to reduce the risk for organizations for potential data unauthorized access after termination

This puts pressure on IT and compliance teams to ensure that all resource access is removed from the former employee without impacting the organization.



With Lifecycle Workflows

Using customizable workflow templates for common offboarding tasks ensures timely, reliable, graceful resource access removal for IT, and peace of mind for former team members.

Leaver scenario - Example



For pre-leaver, "X" days before employee's last day.

Leaver · Pre-Termination



14 days before last day



Send reminder email to manager



Launch custom Logic Apps workflow



Remove user from selected Teams



Remove selected Groups



On the employee's last day, perform the following tasks.

Leaver · Termination



On Last Day



Block user sign-in



Remove user from all Teams channels



Remove all Group assignments



Send reminder email to manager



Upon an employee's real-time termination

Leaver · Emergency Termination



Real-time



Block user sign-in



Remove all Group assignments



Send termination email to team



Remove user from all Teams channels



For post-leaver, "X" days after employee's last day.

Leaver · Post-Termination



30 days after last day



Remove all license assignments



Remove user from all Teams channels



Launch custom Logic Apps workflow



Delete user account

Lifecycle workflows built-in templates

Choose a workflow

Choose a workflow template to start creating your custom workflow. [Learn more](#)

Joiner

Onboard pre-hire employee

Configure pre-hire tasks for onboarding employees before their first day

[Select](#) | [Details](#)

Joiner

Onboard new hire employee

Configure new hire tasks for onboarding employees on their first day

[Select](#) | [Details](#)

Joiner

Post-Onboarding of an employee

Configure onboarding tasks for an employee after their first day of work

[Select](#) | [Details](#)

Mover On-demand

Real-time employee change

Execute real-time tasks for employee job changes

[Select](#) | [Details](#)

Leaver On-demand

Real-time employee termination

Execute real-time termination tasks for employees on their last day of work

[Select](#) | [Details](#)

Leaver

Pre-Offboarding of an employee

Configure pre-offboarding tasks for employees before their last day of work

[Select](#) | [Details](#)

Leaver

Offboard an employee

Configure offboarding tasks for employees on their last day of work

[Select](#) | [Details](#)

Leaver

Post-Offboarding of an employee

Configure offboarding tasks for employees after their last day of work

[Select](#) | [Details](#)

Planning – Define Parameters

Pre-Offboarding of an employee

- Tasks to be completed by the workflow (Remove user from all groups/Teams)
- Trigger date - How many days before the event (EmployeeLeaveDateTime)

Real-time employee termination

- Tasks to be completed by the workflow (Remove user from all groups/Teams, Delete account)
- To run on-demand (Emergency termination)

Offboard an employee

- Tasks to be completed by the workflow (Disable User Account, Remove user from all groups/Teams)
- Trigger date - How many days after the event (EmployeeLeaveDateTime)
- Execution conditions (Rule based on attributes)

Planning – Define Parameters (Cont'd)

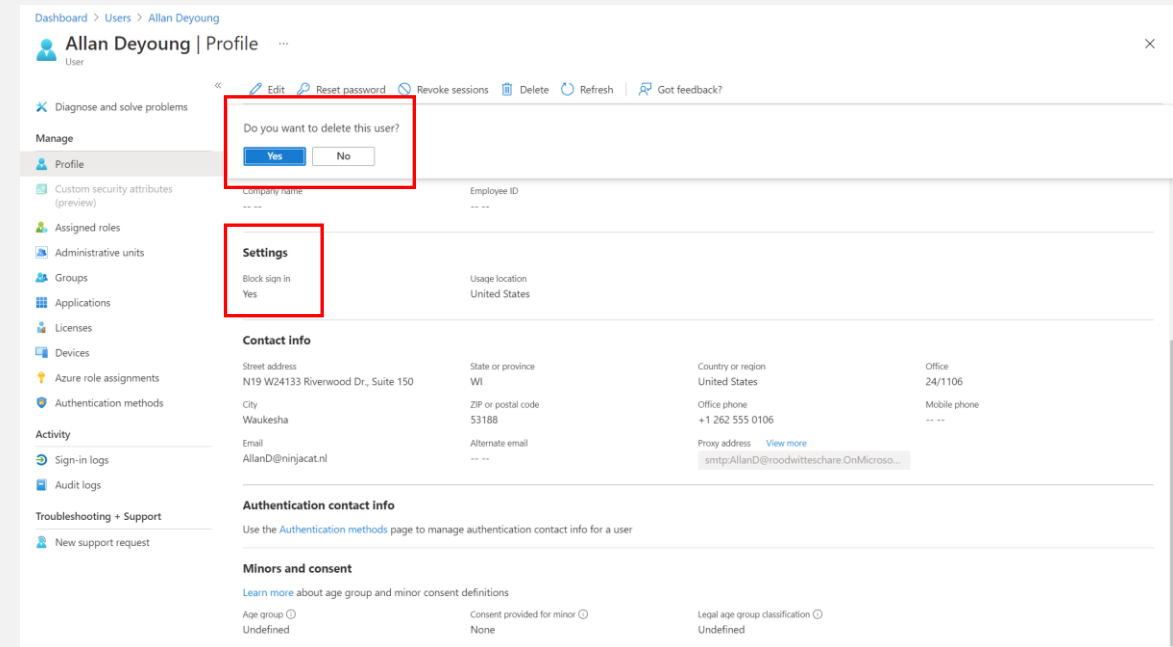
Post-Offboarding of an employee

- Tasks to be completed by the workflow (Remove all licenses for user, Remove user from all Teams, Delete User Account)
- Trigger date - How many days after the event (EmployeeLeaveDateTime)
- Execution conditions (Rule based on attributes)

Removing access

Applications

- Microsoft Entra ID provisioning service keeps source and target systems in sync
- De-provisioning accounts when users should not have access anymore:
 - The user account is deleted in Entra ID
 - The user is unassigned from the application
 - The user no longer meets a scoping filter and goes out of scope
 - The AccountEnabled property is set to False
- Soft delete a user by default if the application supports soft deletes



Lifecycle Workflows custom extensions

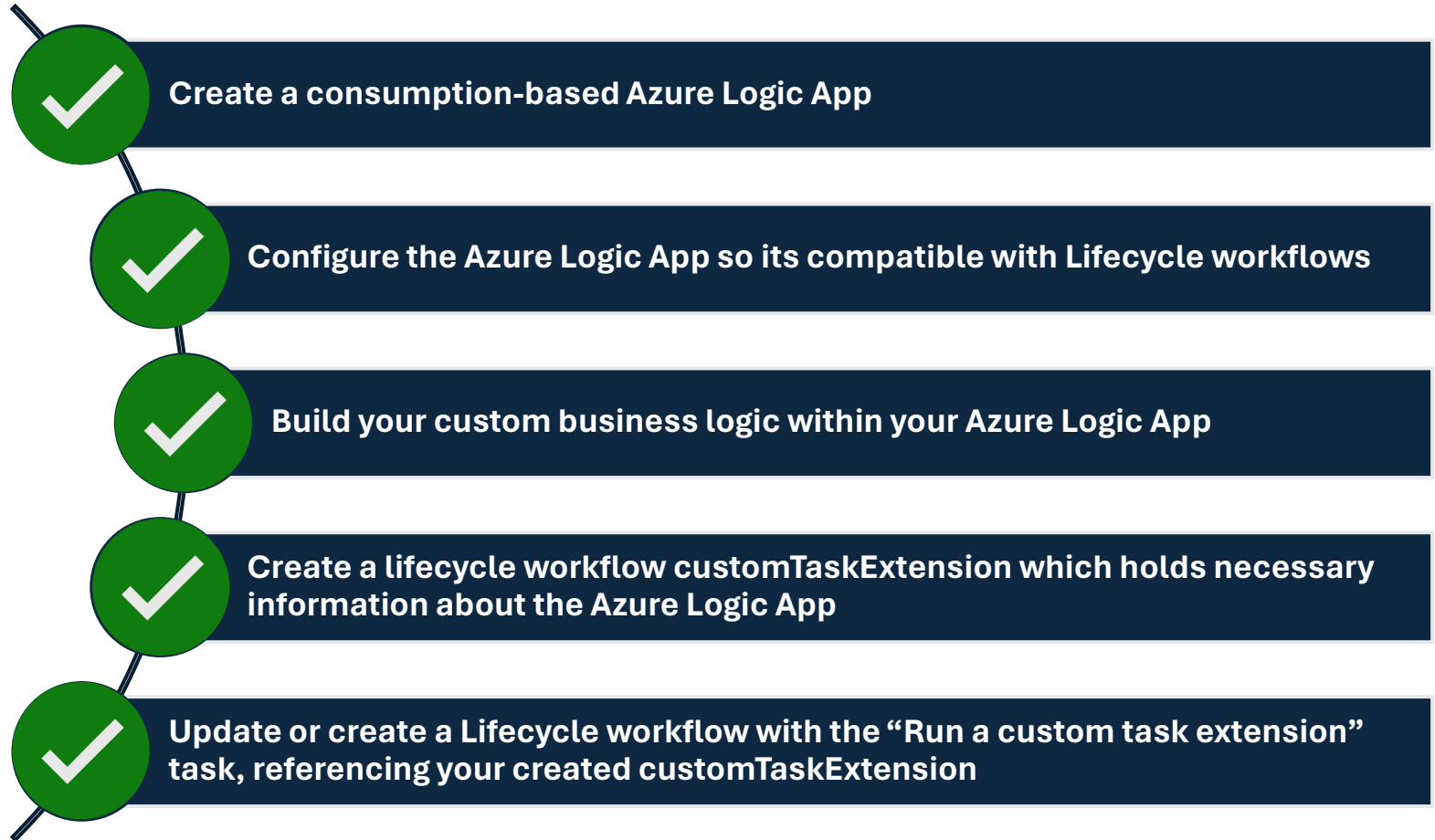
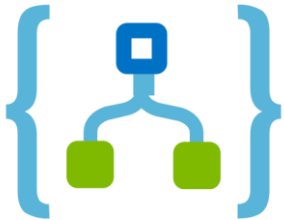
Allows you to create workflows using extensions like Logic Apps. With the extensibility feature, you're able to utilize the concept of custom task extensions to call-out to external systems as part of a workflow.

For example, when a user joins your organization, you can have a workflow with a custom task extension that assigns a Teams number, or have a separate workflow that grants access to an email account for a manager when a user leaves

Prerequisites:

- An Azure subscription
- A resource group
- Permissions to create a new consumption-based Logic App or access to an existing consumption-based Logic App

Azure logic Apps high-level steps



POC Deployment - Lifecycle Workflows (LCW)

| Title | Link |
|--|--|
| 1. How to sync attributes for LCW | Sync attributes for Lifecycle Workflows |
| 2. Prepare user accounts | Prepare user accounts for LCW |
| 3. Onboard pre-hire employee | Automate pre-hire tasks for an employee |
| 4. Onboard new hire employee | Automate Onboarding a new employee |
| 5. Post-Onboarding of an employee | Automate Post-Onboarding of an employee |
| 6. Real-time employee change | Real time employee change |
| 7. Real-time employee termination | Real time employee termination |
| 8. Pre-Offboarding of an employee | Automate Pre-Offboarding of an employee |
| 9. Offboard an employee | Automate Offboarding of an employee |
| 10. Post-Offboarding of an employee | Automate Post-Offboarding of an employee |
| [Optional] – Trigger Logic Apps based on custom extensions | Trigger LogicApps with custom extensions |

Supported Tasks and workflows

| Task | Category |
|---|-----------------------|
| Send welcome email to new hire | Joiner |
| Send onboarding reminder email | Joiner |
| Generate Temporary Access Pass and send via email to user's manager | Joiner |
| Send email to notify manager of user move | Mover |
| Request user access package assignment | Joiner, Mover |
| Add user to groups | Joiner, Leaver, Mover |
| Add user to teams | Joiner, Leaver, Mover |
| Enable user account | Joiner, Leaver |
| Run a custom task extension | Joiner, Leaver, Mover |
| Disable user account | Leaver |
| Remove user from selected group | Joiner, Leaver, Mover |
| Remove users from all groups | Leaver |
| Remove user from teams | Leaver |
| Remove user from all teams | Leaver |
| Remove access package assignment for user | Leaver, Mover |
| Remove all access package assignments for user | Leaver |
| Cancel all pending access package assignment requests for user | Leaver |
| Remove all license assignments from user | Leaver |
| Delete user | Leaver |
| Send email to manager before user's last day | Leaver |
| Send email on user's last day | Leaver |
| Send email to user's manager after their last day | Leaver |

Join Entra ID Governance Advisors - Customer Community

What is Entra ID Governance Advisors

- Entra ID Governance Advisors is a community that consists of selected customers and partners who collaborate via virtual small/large group discussions, content reviews, digital forum and more

Benefits of Joining the Entra ID Governance Advisors:

- Members benefit by participating in the following ways:
 - Direct engagement with Microsoft Product Groups
 - Dedicated sessions focused on upcoming features and deep-dives
 - Early access to Private Preview and Roadmap access
 - Valuable inputs from Microsoft and other customers all under NDA
 - Learn and interact with other customers across verticals, sizes, and segments\
-
- Please fill out the survey here if interested in joining: <https://aka.ms/MicrosoftEntraAdvisors/>

Next Steps

Give us feedback, let us know y
our comments:
aka.ms/idnecat/igapocsurvey

Are you ready for deployment?

```
object to mirror
mirror_mod.mirror_object

operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob
mirror_ob.select = 0
= bpy.context.selected_object
data.objects[one.name].select

print("please select exactly

-- OPERATOR CLASSES --

types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"
```

Thank you