**SOLUTION DESIGN DOCUMENT**

**Service Request Automation**

Version v.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Author | Description |
| 24-07-2024 | V.1 | Ravi Teja, Ragam | SDD |

Contents

Revision History 1

[Distribution 3](#_Toc1342706237)

Introduction 4

[Overview 4](#_Toc397618729)

[Scope 4](#_Toc1716037912)

[Contacts 5](#_Toc2021859910)

Quick Facts 5

[Idea Assessment 6](#_Toc1611864808)

[Project 6](#_Toc1722526419)

[Architecture 6](#_Toc76305519)

[Overview 6](#_Toc1117649874)

[Flow Diagram 7](#_Toc836676382)

[Required Fields for Creation of Cost Center](#_Toc836676382) 8

[Component 1 –HTTP Connector for ServiceNow 1](#_Toc1341941437)0

[Component 2 – Dataverse Connector 1](#_Toc260402148)1

[Component 3 – Custom Connector for Workday 1](#_Toc260402148)4

Data 18

[Regulatory Requirements 1](#_Toc861510299)8

[Security 1](#_Toc1525485457)8

[Risks and Mitigation 1](#_Toc1803789990)8

[Solution Metrics 1](#_Toc1450064832)8

[License Estimation 1](#_Toc1439350807)8

[Triggers 1](#_Toc2112533367)8

[Business 1](#_Toc1222066373)9

[Technical](#_Toc1814719706) 19

# Distribution

The information has restricted distribution and viewing within Chanel.

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 24/07/2024 | V1.0 | SDD | Ravi Teja, Ragam |
|  |  |  |  |

Document Sign-off Requirements (NA)

The following table contains the people required to sign-off and/or review this document and those that require the document for information only.

|  |  |  |
| --- | --- | --- |
| Name | Department | Responsibility |
| **KANAKARATHNAM, KARTHICK** |  | Senior Manager |
|  |  |  |
|  |  |  |

# Introduction

## Overview

## Business Requirement

## To automate the process of Cost Center creation in Workday using Power Platform Solution.

## Process Description

As part of this automation, we will create a Canvas Application using Power App, in which the Cost Center Service request ticket details are present. To create the Cost Center, the User needs to call APIM through Power Apps Custom Connector by hitting the **Create Cost Center button** UI in App. APIM subsequently creates the Cost Center and Cost Center Hierarchy in Workday using API calls.

**Features of App**

* The details of Cost Center tickets from ServiceNow should be shown in the Power App interface.
* The user should be able to create the cost center and cost center hierarchy by using this power app interface.
* Power Apps Custom Connector Creates the Cost Centers by using APIM and returns the results to Power App.

## Prerequisite

* Power App development environment.
* Power Platform Premium License for HTTP Connector, Dataverse and Azure Key Vault.
* For extracting ServiceNow Cost Center related tickets, Custom API (Application Programming Interfaces) is required.
* For Creation of Cost Center in Workday, APIM is needed.

### Current Behavior:

In the current process, managing cost centers and cost center hierarchies involves a manual workflow heavily reliant on user intervention and prone to inefficiencies and errors. Users manually review the ServiceNow tickets assigned to the team if it is related to Cost Center Creation or not. This requires careful examination of ticket details to ensure accurate identification and categorization of the request. If it is related to Cost Center Creation, then they will create the cost center using Workday catalog after that they will create the hierarchy with local and global Cost Centers in Workday.

### Proposed Solution:

Hence, to automate the above process, the Team proposed the Power Platform Solution of creating an interface where user will be able to see all the ticket details from ServiceNow in Power Apps and Workday team can review it. After reviewing, on click of the button, APIM will Create the Cost Center using Workday API through Custom Connector and return the New Cost Center name in the Power App interface.

## Scope

**In Scope:**

* Region: Global
* Markets: Global
* Staff: IT Unit
* Data Files: NA

## Contacts

|  |  |  |
| --- | --- | --- |
| Key Contacts | Email ID | Role |
| **KANAKARATHNAM, KARTHICK** | karthick.kanakarathnam@capgemini.com | Phoenix Workday SC Lead |
| Dilip Thankappan | dilip.thankappan@capgemini.com | Phoenix RPA SC Lead |
| Ravi Teja, Ragam | ragam.ravi-teja@capgemini.com | Phoenix Power Platform Developer |
| Chaudhari, Shivpriya | shivpriya.pavan-chaudhari@capgemini.com | Phoenix Workday Developer |

# Quick Facts

## Idea Assessment

|  |  |
| --- | --- |
| Topic | Value |
| Business Unit  *(1. Fashion, 2. Fragrance and Beauty, 3. WFJ)* | 3 |
| Use Case  *(Automation, Mobile/tablet App, Web Portal, Workflow, Chatbot,* *IDP, ML/AI, Others)* | *Automation* |
| Impact on Business  *(Business Critical, Not Business Critical)* | Not Business Critical |
| Users  *(Individual or Small Team, Large Team or Service, Department or Region)* | Individual or Small Team |
| Power Platform Features  *(Power Automate, Power Apps, Power Virtual Agents, Power* *BI, Power Automate Desktop,* *AI Builder)* | Power Automate, Power Apps |
| Power Platform Connectors  *(List of Power Platform Connectors)* | HTTP Connector and Dataverse |
| System Interactions | ServiceNow and Workday |

## 

## Project

|  |  |
| --- | --- |
| Topic | Value |
| Project Type  *(GYRB)* | R |
| Environments  *(GYRB)* | R |
| Licensing | Microsoft 365 E5 |
| Track  *(Business Project, Citizen IT)* | Business Project |
| External Partner\* | NA |

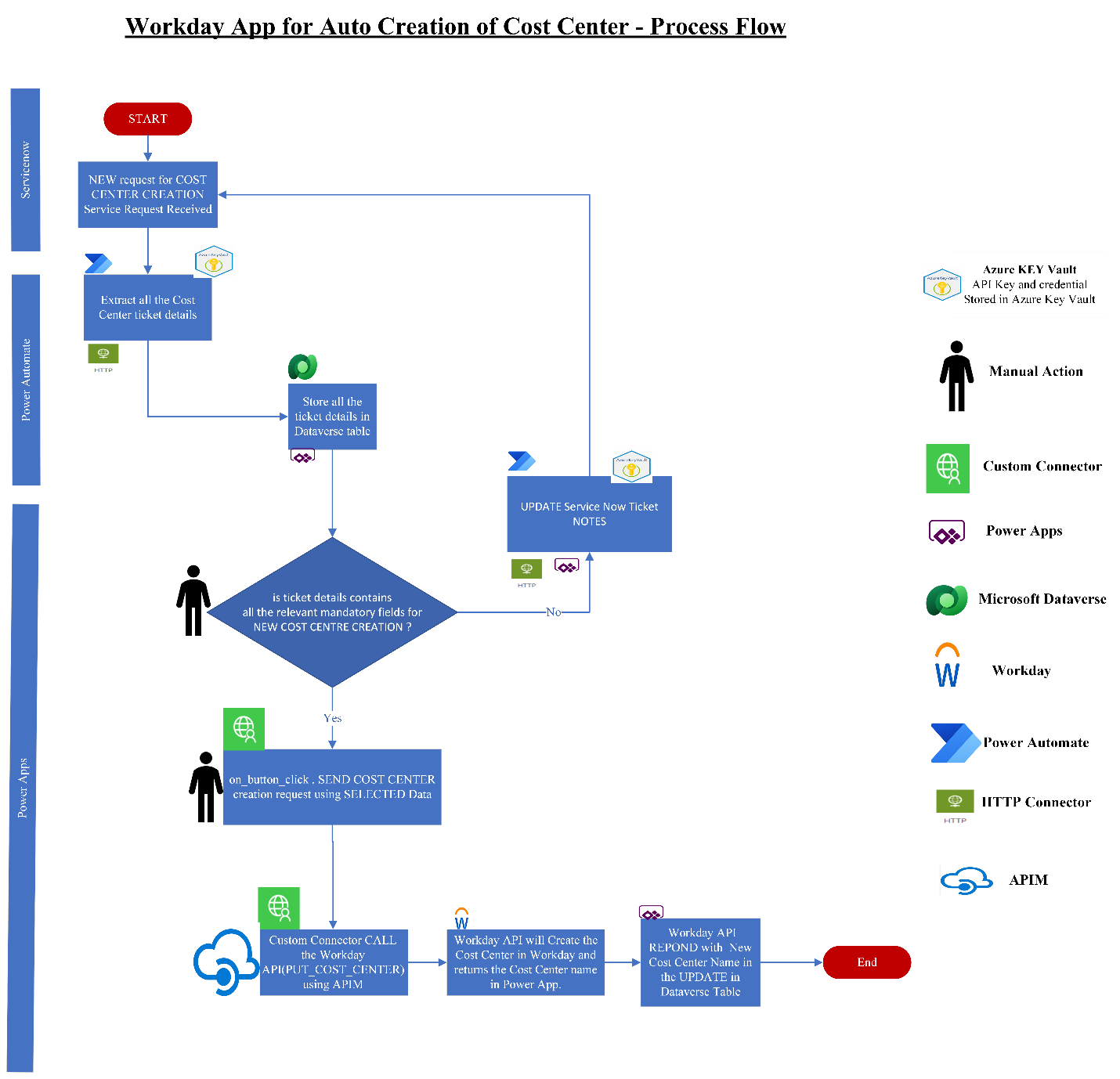
*\* Business Projects only*

**Architecture**

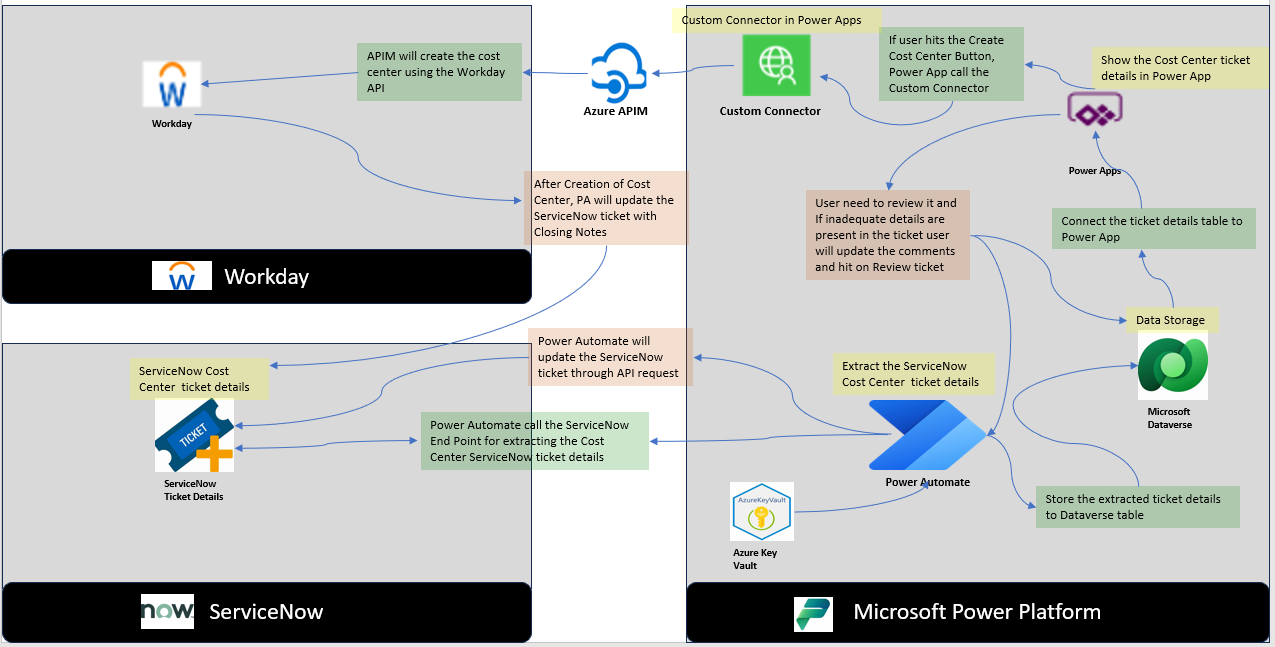
## Overview

This architecture involves using Power Automate HTTP Connector to call ServiceNow API. The ServiceNow API will return Cost Center ticket details such as Name, Code, Reference ID, Local Cost Center Name and Global Cost Center Name. The extracted information is then stored in a Dataverse table and displayed in a Power App. Within the Power App, the user needs to review and hit on “Create Cost Center” button if they want to create the Cost Center. On hit, APIM will Create the Cost Center using Workday API through Custom Connector and returns the Cost Center Name to the Power App.

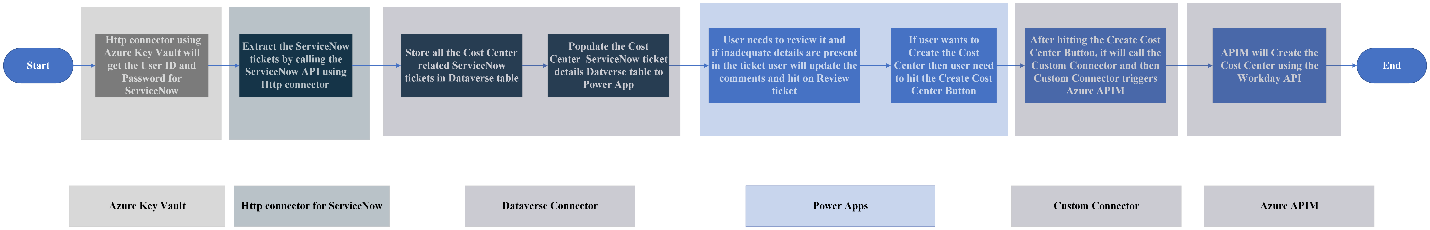
## Flow Diagram



**Diagram 1.0**: Flow Diagram for Service Request Automation

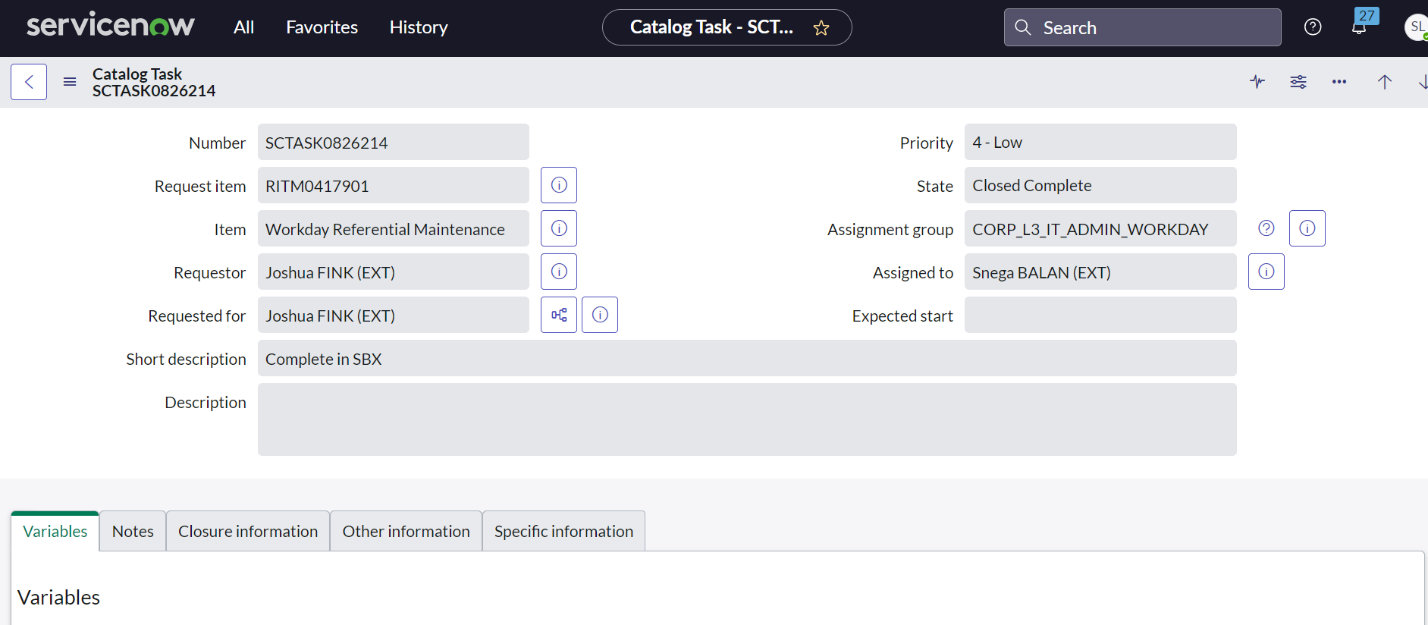


**Fig: Solution Design Workflow for Service Request Automation**

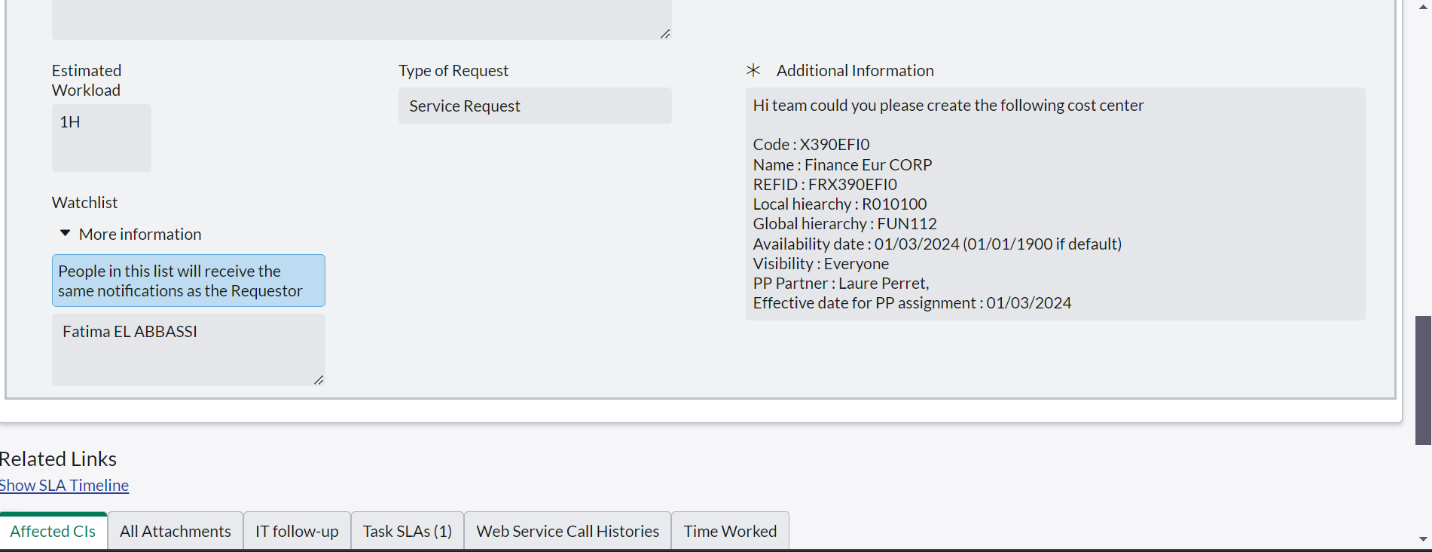


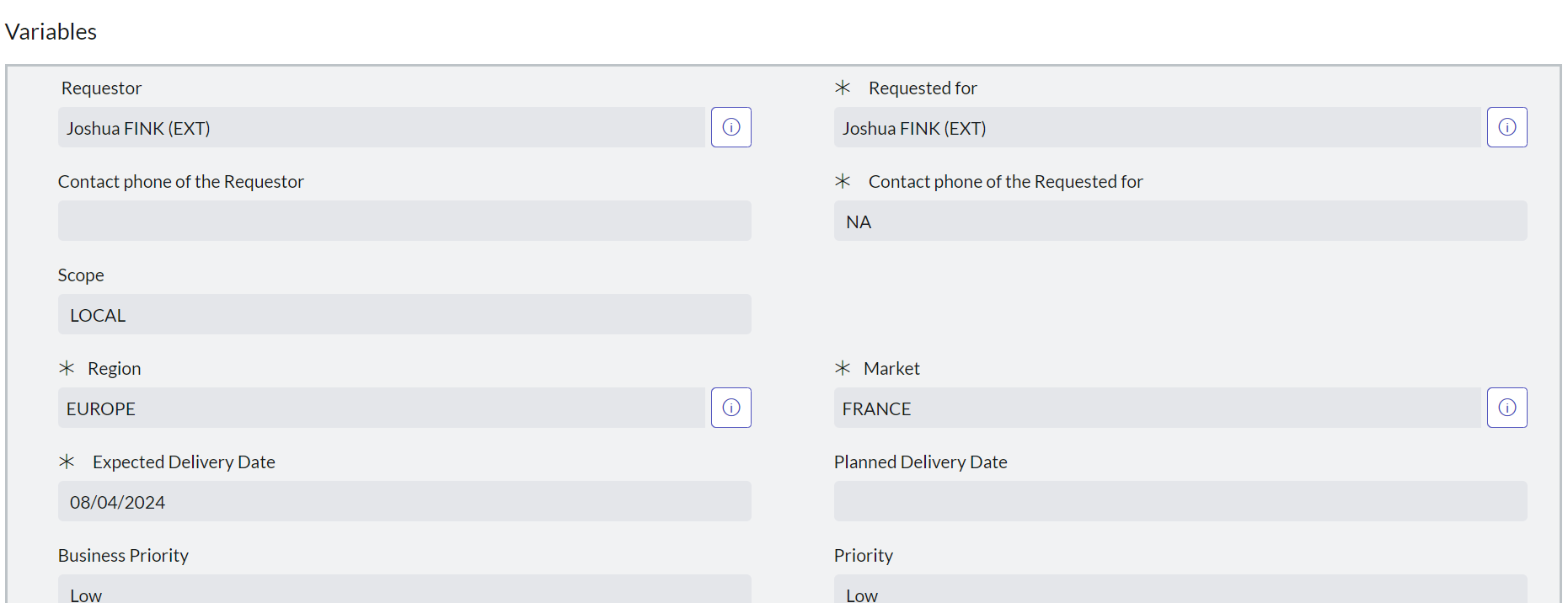
**Diagram** **2.0**: High-level flow diagram

**Required Fields for Creation of Cost Center:**

****

****

****



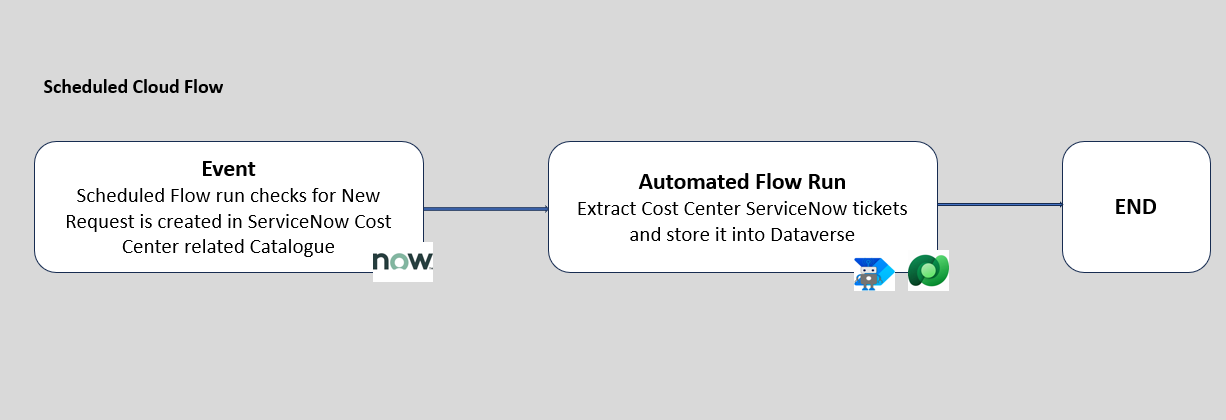
**Required Parameters for Cost Center Workday API:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Field Names** | **Description** | **Default** |
| 1 | Re Organization | (01/01/1900) | Yes |
| 2 | Name | Extract from ServiceNow API | No |
| 3 | Code | Extract from ServiceNow API | No |
| 4 | Include Code in Name | Boolean Value | Yes |
| 5 | Subtype | Cost Center | Yes |
| 6 | Visibility | Everyone | Yes |
| 7 | Include Organizations à Local Cost Center Name and Global Cost Center Name | Extract from ServiceNow API | No |
| 8 | Reference ID Values | Extract from ServiceNow API | No |

**Solution Components**

The following steps will be performed in the process.

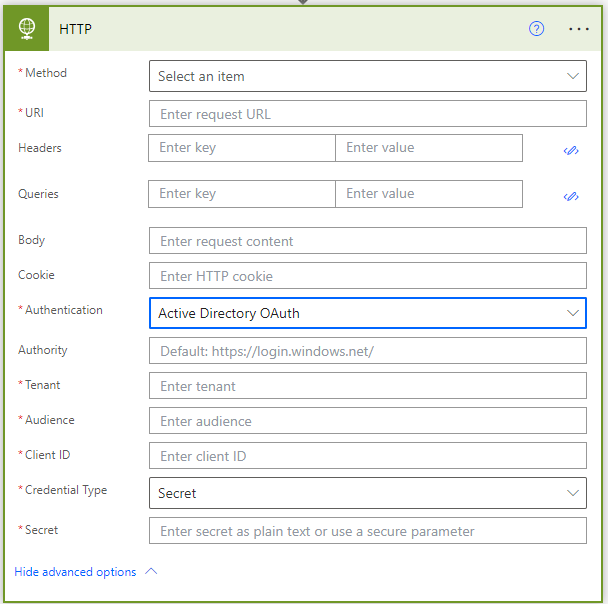
***Extract the Cost Center ServiceNow ticket details and store it into Microsoft Dataverse Table***

******

***Fig: Extract the Cost Center ServiceNow ticket details***

## HTTP Connector for extract the Cost Center ServiceNow Ticket details.

The ServiceNow team will provide the API that End Point lands in the Create Cost Center Catalog. Using the API, we will extract the details and store it into the Dataverse table.



**Method:** Get method is used to fetch the ticket details from ServiceNow

**URI:** The URI would be the endpoint provided by the ServiceNow API for retrieving tickets. It might be something look like ‘**https://your-instance.service-now.com/api/now/table/incident**’.

**Headers:**

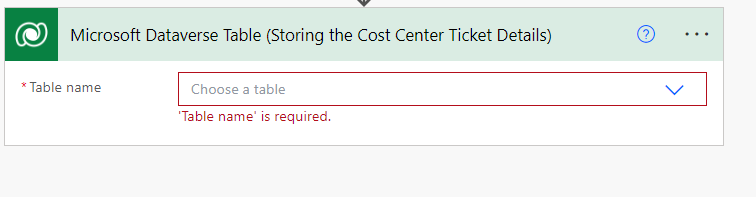
Content-Type: Content-Type: application/json

Authorization: Access token

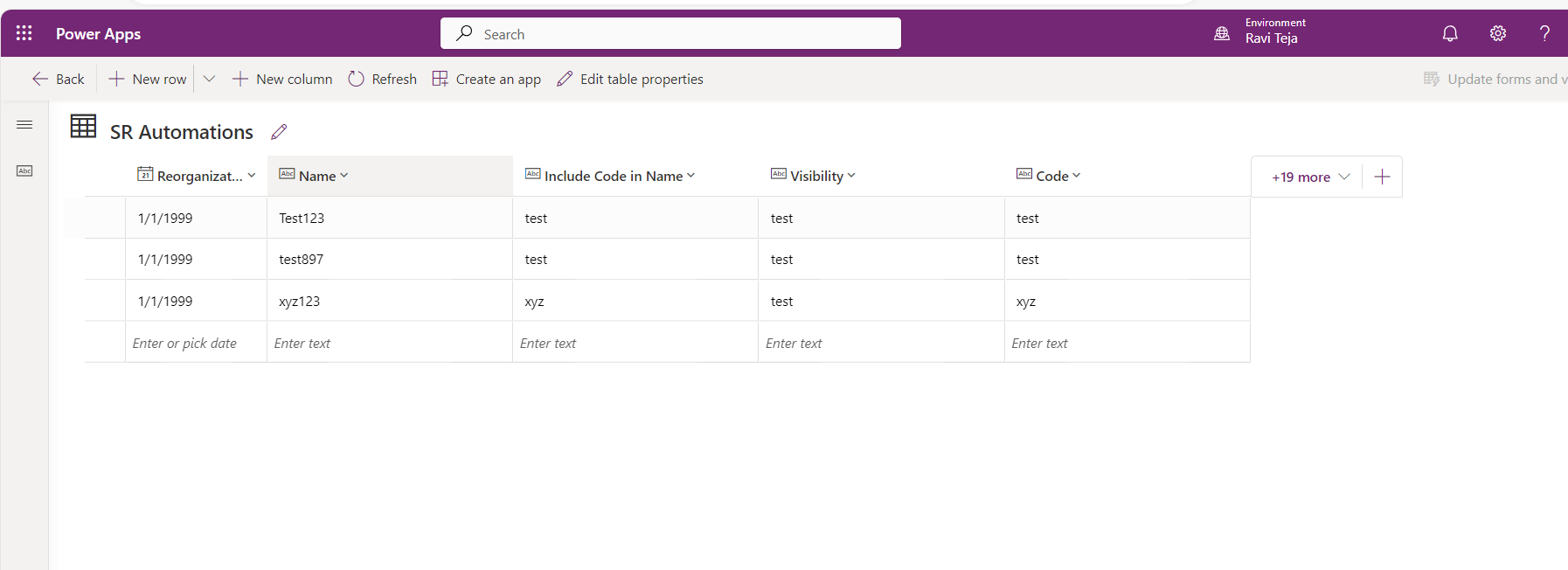
|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Remarks** |
| HTTP Connector | Tenant, Audience, client\_id, credentials\_Type,Secret | ServiceNow team will provide |

## Dataverse Connector for Storing the ServiceNow Ticket details in Dataverse Table

After Extracting Cost Center ServiceNow ticket details, store the ticket details in Microsoft Dataverse table for historical purpose.



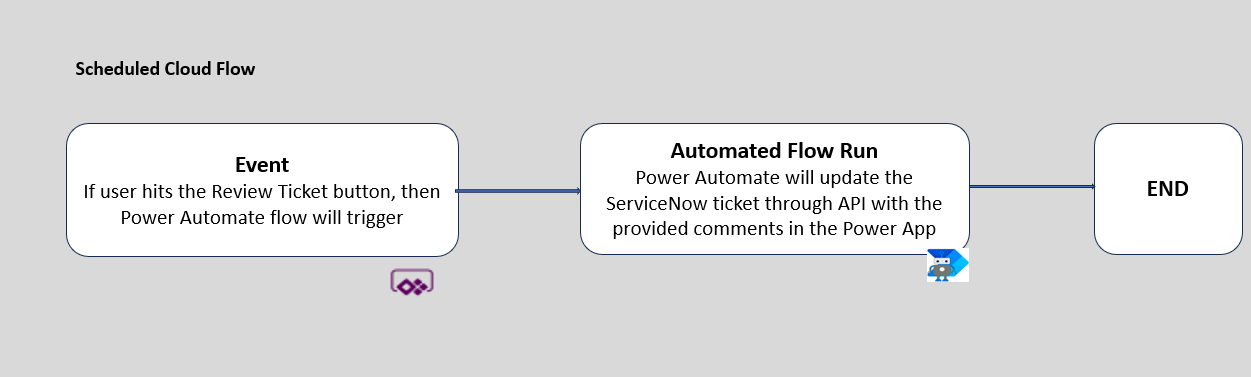
***Populate the Cost Center ServiceNow Ticket details from Dataverse Table to the Power App***

******

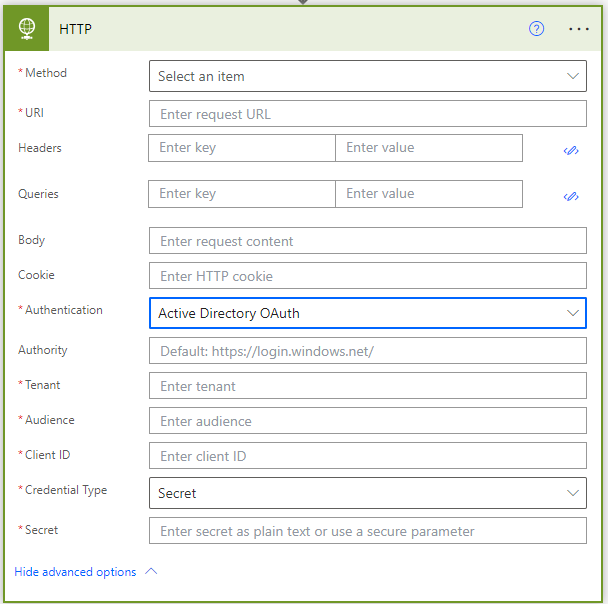
* Connect the Cost Center ServiceNow ticket details Dataverse table to the Power App as a Data Source.
* Power App UI having all the Cost Center ServiceNow ticket details along with each ticket details having Create Cost Center Button.
* Users need to review it and click on Create Cost Center Button.

**If inadequate details are present in the ticket user will update the comments and hit on Review ticket.**

* Once if user hits on Review ticket, Power App will trigger the Power Automate flow and then Power Automate will the call the ServiceNow API and it will update the ServiceNow ticket with provided comments.

******

***Fig: Update the ServiceNow ticket through API***



**Method: POST** method is used to update the ServiceNow ticket in ServiceNow.

**URI:** The URI would be the endpoint provided by the Workday API for creating a cost center. It might be something look like ‘**https://your-instance.service-now.com/api/now/table/incident**’**’**.

**Headers:**

Content-Type: Content-Type: application/json

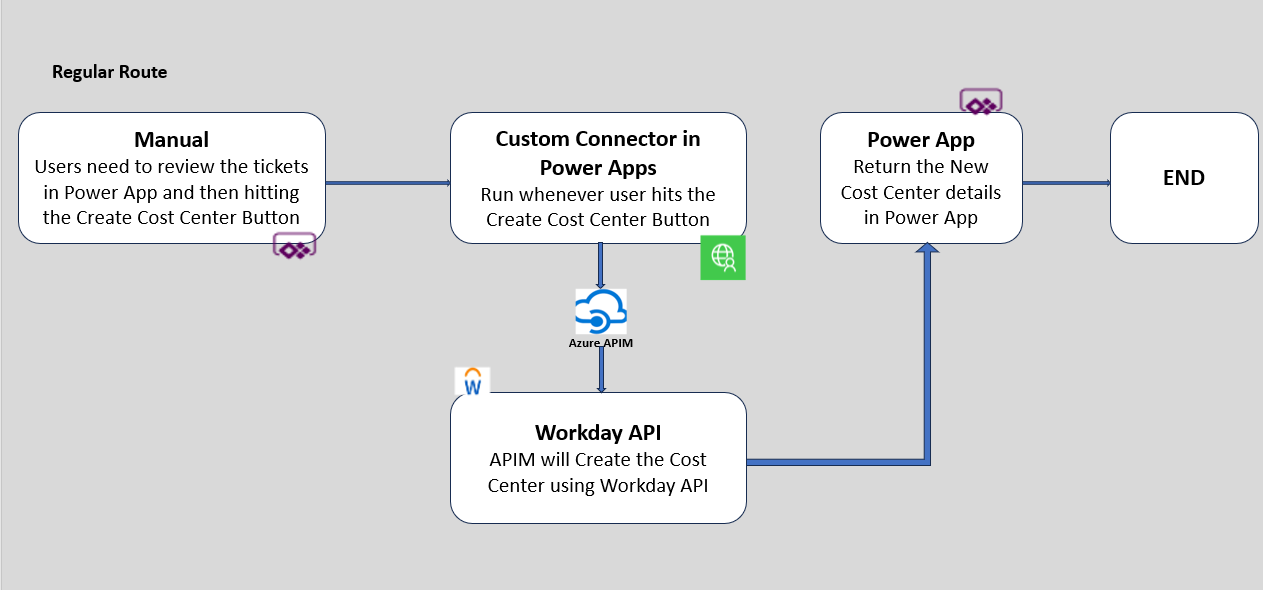
Authorization: Access token

**Request Body:**

The request body would have the details of the cost center that the user wants to create.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Remarks** |
| HTTP Connector | Tenant, Audience, client\_id, credentials\_Type,Secret | ServiceNow team will provide |

***Creation of Cost Centers by using Workday API through Azure APIM***

******

***Fig: Creation of Cost Center by using Workday API through Custom Connector in Power Apps***

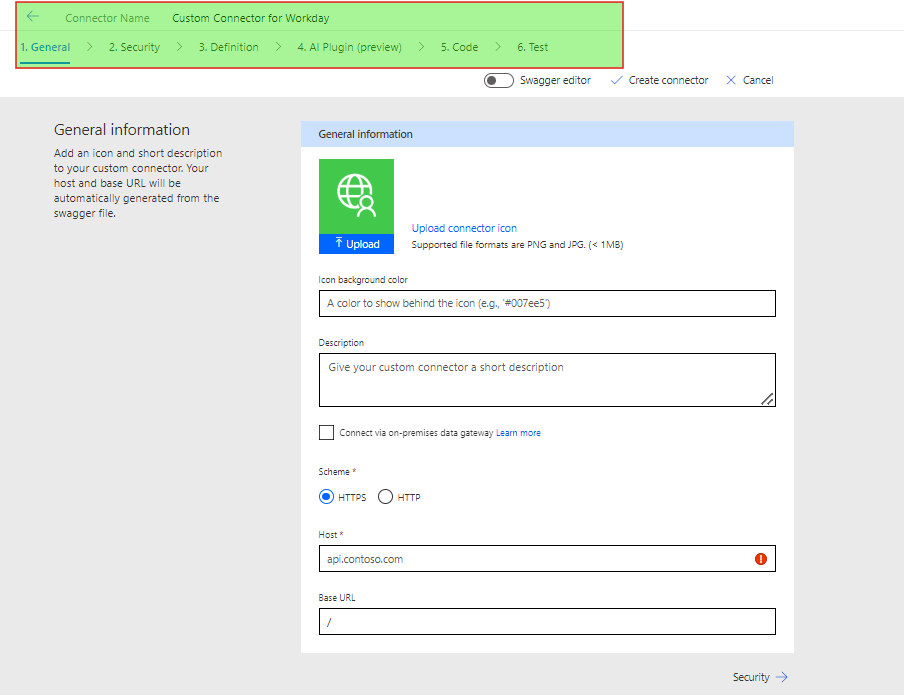
* In Power Apps, after hitting the Cost Center Create button, it will call the APIM through Custom Connector.
* APIM will Create the Cost Center using Workday API in Workday.
* Workday API PUT\_COST\_CENTER will create the Cost Center in Workday and returns the Cost Center Name.

For Creation of Custom Connector, we should have to follow 6 steps which are shown in the below figure. Out of 6, 4 steps are mandatory,

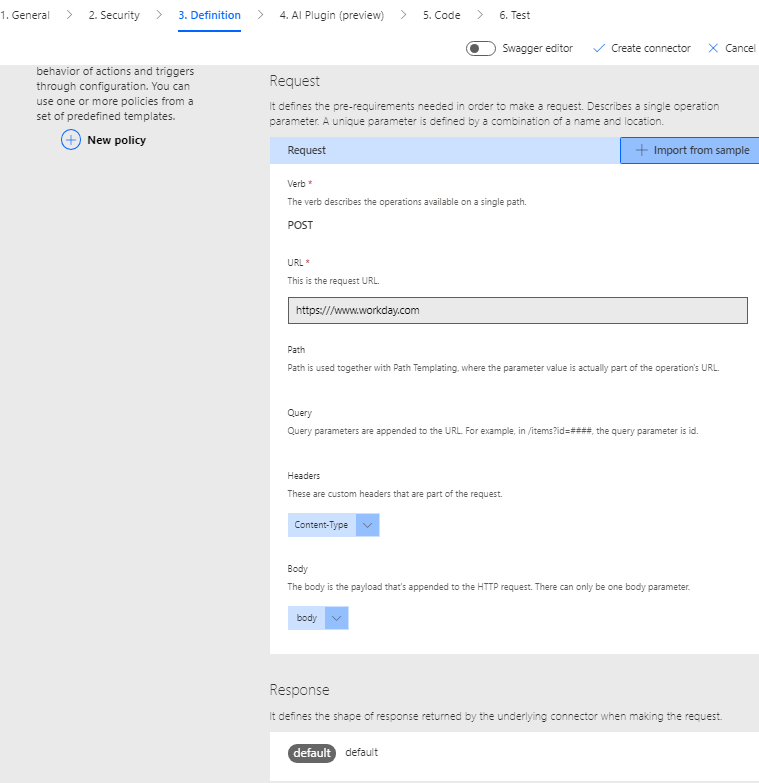
1. General: It contains the API URL provided by the APIM team.
2. Security: It needs to define by API Team.

|  |  |  |  |
| --- | --- | --- | --- |
| **Area** | **Parameter** | **Value** | **Remarks** |
| General | Operational ID | User defined | Required later in the canvas app to call Custom Connector |
| Request | Header | Content-Type | APIM team will provide |
| Body | client\_id, client\_credentials, grant\_type, Scope | APIM team will provide |
| Response | Bearer Token | Output to the request |

1. Definition: It has two areas: General and Request
2. Test: Test the functionality of the custom connector.

****

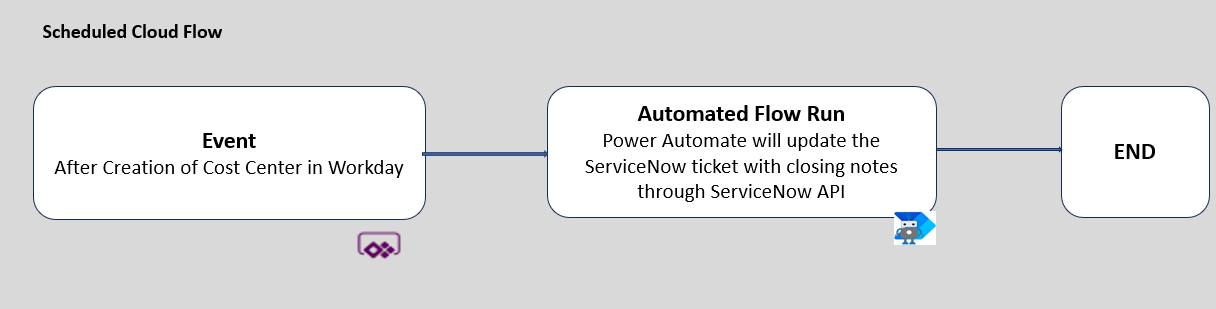
**Fig: Custom Connector in Power Apps**



**Fig: Definition**

***Power Automate will update the ServiceNow ticket with closing notes through ServiceNow API***

* After Creation of Cost Center, Power Automate will close the ServiceNow ticket with Closing notes through ServiceNow API.



**Fig: Update the ServiceNow tickets through ServiceNow API using Power Automate**

# Data:

NA

**Regulatory Requirements:**

# Security

## Risks and Mitigation

<Brief overview on how the different stakeholders interact with the solution, what are the risks associated and the mitigation actions for those risks>

**Solution Metrics**

## License Estimation:

## Triggers:

When a service request is created in ServiceNow.

## Business

<List all know business exceptions, per task, that are expected for the solution. Each exception should include details such as the task, exception description, affected applications and manual actions needed to solve issue either by Business or Support teams>

## Technical

<List all know system exceptions, per task, that are expected for the solution. Each exception should include details such as the task, exception description, affected applications and manual actions needed to solve issue or to alert the person/team that can solve the issue>