Marathon Petroleum Corporation Design Document

**Prepared by:**

**Simeio Solutions**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
|  |  |  |  |
| Aug 16, 2019 | 0.3 | Initial Draft of Design Document | Simeio Solutions |
| Dec 18 , 2019 | 0.5 | Updated Design Document | Simeio Solutions |
| Dec 24,2019 | 0.6 | Updated Design Document | Simeio Solutions |
| Jan 7, 2020 | 0.7 | Updated Design Document sent to MPC team for review and feedback | Simeio Solutions |
| Jan 31, 2020 | 0.8 | Updated Design Document with internal reviews from Architect | Simeio Solutions |
| Feb 20,2020 | 0.9 | Updated Design document with Role imports, Role user and Role access | Simeio Solutions |
| **Feb 28,2020** | **1.0** | **Final version of Design document sent to MPC team for review , feedback & sign off** | **Simeio Solutions** |

Contents

[1 Introduction: 8](#_Toc33292527)

[1.1 Documents scope: 8](#_Toc33292528)

[1.2 Project Overview: 8](#_Toc33292529)

[2 Design and Architecture: 8](#_Toc33292530)

[2.1 Logical Architecture: 8](#_Toc33292531)

[2.2 Physical Architecture 9](#_Toc33292532)

[3 Network Detail: 10](#_Toc33292533)

[4 Saviynt Security Model: 10](#_Toc33292534)

[4.1 Authentication: 10](#_Toc33292535)

[4.2 Authorization: 10](#_Toc33292536)

[5 Identity Imports: 11](#_Toc33292537)

[5.1 Employee onboarding 11](#_Toc33292538)

[5.1.1 Prerequisites 11](#_Toc33292539)

[5.1.2 Connection details 12](#_Toc33292540)

[5.1.3 Attribute Mapping 13](#_Toc33292541)

[5.1.4 Database password encryption and decryption 15](#_Toc33292542)

[5.1.5 Employee Creation: New Hire 16](#_Toc33292543)

[5.1.6 Employee Modification: Update 16](#_Toc33292544)

[5.1.7 Employee Modification: LOA Status 17](#_Toc33292545)

[5.1.8 Employee: Termination 18](#_Toc33292546)

[5.1.9 Employee: Rehire 19](#_Toc33292547)

[5.2 Contractor/Non-Employee onboarding 19](#_Toc33292548)

[5.2.1 User types 19](#_Toc33292549)

[5.2.2 Non-Employee Creation: New Hire 20](#_Toc33292550)

[5.2.3 Non-Employee Conversion: Contractor to Employee -Possibility 1 21](#_Toc33292551)

[5.2.4 Non-Employee Conversion: Contractor to Employee -Possibility 2 21](#_Toc33292552)

[5.2.5 Non-Employee Conversion: Contractor to Employee -Possibility 3 22](#_Toc33292553)

[5.2.6 Non-Employee Conversion: Contractor to Employee -Possibility 4 22](#_Toc33292554)

[6 Target System 23](#_Toc33292555)

[6.1 Active Directory 23](#_Toc33292556)

[6.1.1 Integration Design 23](#_Toc33292557)

[6.1.2 Connection details 23](#_Toc33292558)

[6.1.3 Attribute mapping 26](#_Toc33292559)

[6.1.4 Account provisioning 27](#_Toc33292560)

[6.1.5 Account Access provisioning 32](#_Toc33292561)

[6.1.6 Account Termination 35](#_Toc33292562)

[6.1.7 Account De-provisioning 35](#_Toc33292563)

[6.1.8 Account Re-Hire 38](#_Toc33292564)

[6.1.9 Account Target recon 39](#_Toc33292565)

[6.2 SAP BW2 44](#_Toc33292566)

[6.2.1 Integration Design 44](#_Toc33292567)

[6.2.2 Connection Details 44](#_Toc33292568)

[6.2.3 Attribute mapping 44](#_Toc33292569)

[6.2.4 Account provisioning 45](#_Toc33292570)

[6.2.5 Account Access provisioning 47](#_Toc33292571)

[6.2.6 Account De-provisioning 48](#_Toc33292572)

[6.2.7 Account Re-Hire 49](#_Toc33292573)

[6.2.8 Account Target recon 50](#_Toc33292574)

[6.3 SAP EC2 50](#_Toc33292575)

[6.3.1 Integration Design 50](#_Toc33292576)

[6.3.2 Connection Details 50](#_Toc33292577)

[6.3.3 Attribute mapping 51](#_Toc33292578)

[6.3.4 Account provisioning 51](#_Toc33292579)

[6.3.5 Account Access provisioning 53](#_Toc33292580)

[6.3.6 Account De-provisioning 54](#_Toc33292581)

[6.3.7 Account Re-Hire 55](#_Toc33292582)

[6.3.8 Account Target recon 56](#_Toc33292583)

[6.4 SAP GRD 56](#_Toc33292584)

[6.4.1 Integration Design 56](#_Toc33292585)

[6.4.2 Connection Details 56](#_Toc33292586)

[6.4.3 Attribute mapping 57](#_Toc33292587)

[6.4.4 Account provisioning 57](#_Toc33292588)

[6.4.5 Account Access provisioning 59](#_Toc33292589)

[6.4.6 Account De-provisioning 60](#_Toc33292590)

[6.4.7 Account Re-Hire 61](#_Toc33292591)

[6.4.8 Account Target recon 62](#_Toc33292592)

[6.5 SAP MDD 62](#_Toc33292593)

[6.5.1 Integration Design 62](#_Toc33292594)

[6.5.2 Connection Details 62](#_Toc33292595)

[6.5.3 Attribute mapping 63](#_Toc33292596)

[6.5.4 Account provisioning 63](#_Toc33292597)

[6.5.5 Account Access provisioning 65](#_Toc33292598)

[6.5.6 Account De-provisioning 66](#_Toc33292599)

[6.5.7 Account Re-Hire 67](#_Toc33292600)

[6.5.8 Account Target recon 68](#_Toc33292601)

[6.6 SAP MGD 68](#_Toc33292602)

[6.6.1 Integration Design 68](#_Toc33292603)

[6.6.2 Connection Details 68](#_Toc33292604)

[6.6.3 Attribute mapping 69](#_Toc33292605)

[6.6.4 Account provisioning 69](#_Toc33292606)

[6.6.5 Account Access provisioning 71](#_Toc33292607)

[6.6.6 Account De-provisioning 72](#_Toc33292608)

[6.6.7 Account Re-Hire 73](#_Toc33292609)

[6.6.8 Account Target recon 74](#_Toc33292610)

[6.7 SAP SRD 74](#_Toc33292611)

[6.7.1 Integration Design 74](#_Toc33292612)

[6.7.2 Connection Details 74](#_Toc33292613)

[6.7.3 Attribute mapping 75](#_Toc33292614)

[6.7.4 Account provisioning 75](#_Toc33292615)

[6.7.5 Account Access provisioning 77](#_Toc33292616)

[6.7.6 Account De-provisioning 78](#_Toc33292617)

[6.7.7 Account Re-Hire 79](#_Toc33292618)

[6.7.8 Account Target recon 80](#_Toc33292619)

[6.8 SAP GRC 80](#_Toc33292620)

[6.8.1 Integration Design 80](#_Toc33292621)

[6.8.2 Connection Details 80](#_Toc33292622)

[6.8.3 ARS Access Provisioning 81](#_Toc33292623)

[6.8.4 External Config for SOD Evaluation 83](#_Toc33292624)

[6.9 GOLD Tables 84](#_Toc33292625)

[6.9.1 Connection details 85](#_Toc33292626)

[6.9.2 Attribute Mapping 85](#_Toc33292627)

[6.9.3 Rules 86](#_Toc33292628)

[6.9.4 Job 87](#_Toc33292629)

[6.9.5 Account Provisioning 87](#_Toc33292630)

[6.9.6 Account Update 88](#_Toc33292631)

[6.10 Polaris 89](#_Toc33292632)

[6.10.1 Connection details 89](#_Toc33292633)

[6.10.2 Attribute mapping 90](#_Toc33292634)

[6.10.3 Roles 90](#_Toc33292635)

[6.10.4 Role request approval workflow 93](#_Toc33292636)

[6.10.5 Account provisioning 93](#_Toc33292637)

[6.10.6 Account De-provisioning 94](#_Toc33292638)

[7 Role Management - 94](#_Toc33292639)

[7.1 Enterprise roles: 94](#_Toc33292640)

[7.2 Application roles 94](#_Toc33292641)

[8 Data Migration – 95](#_Toc33292642)

[8.1 Users 95](#_Toc33292643)

[8.2 Access 95](#_Toc33292644)

[8.3 Roles 95](#_Toc33292645)

# Introduction:

## Documents scope:

The Saviynt Architecture and Design document defines and outlines the implementation at Marathon Petroleum Corporation. The combined analysis of the current state environment and as per the requirement document. This document will address and highlight:

* The various identity repositories which exist in the current network and their functionality
* Identifying and designing the overall architecture for all the entities for a functional access governance program
* Setting up three (3) physical environments:
  + Development Environment
  + UAT Environment
  + Production Environment

## Project Overview:

As a part of the overall IAM Risk Management and Security program, MPC has identified specific initiatives they would like to pursue in order to accelerate achievement of key business objectives. MPC has requested Simeio to address Access Governance as part of phase 1 initiative.

The purpose of this project is to ensure that the right individuals have the right access to the right resources at the right times and for the right reasons.

The plan is to implement and use the Saviynt tool. This project will first analyses the existing infrastructure and access given methods and business tools that is used in it and replace the same functionalities with Saviynt without changing the business logic and if any enhancement required then that will be suggested and carried over in the current architecture.

# Design and Architecture:

## Logical Architecture:

The logical architecture describes all high-level system components that are needed to implement the Saviynt Security Manager (SSM). The SSM solution is a web-based application that runs on a relational database and application server.

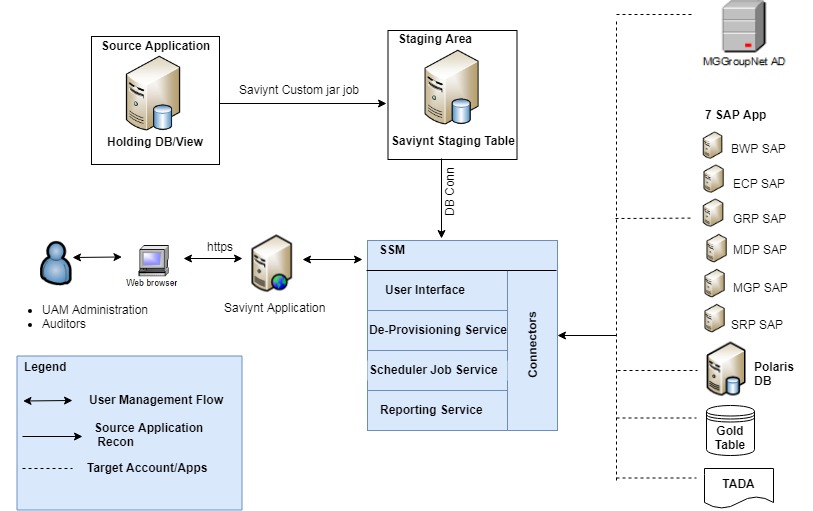
**HR Trusted Feeds:**

1. Success Factor VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL (Employees)

**Target Reconciliation:**

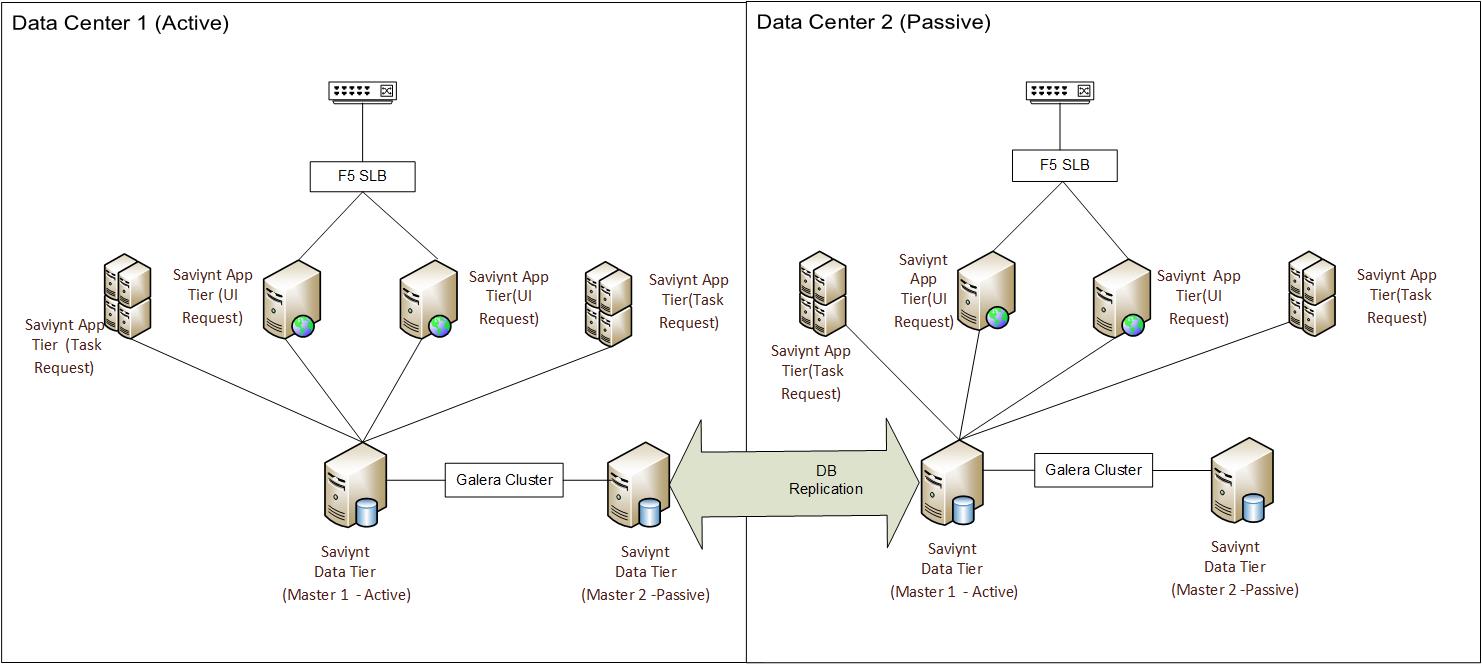
1. Active Directory
2. Polaris DB
3. Gold DB
4. SAP BWP
5. SAP ECP
6. SAP GRP
7. SAP MDP
8. SAP MGP
9. SAP SRP

The diagram below details the logical architecture.



## Physical Architecture

The physical architecture describes the deployment layout of hardware and software components, including integration with network infrastructure. The Saviynt Security Manager will be deployed on separate servers.



# Network Detail:

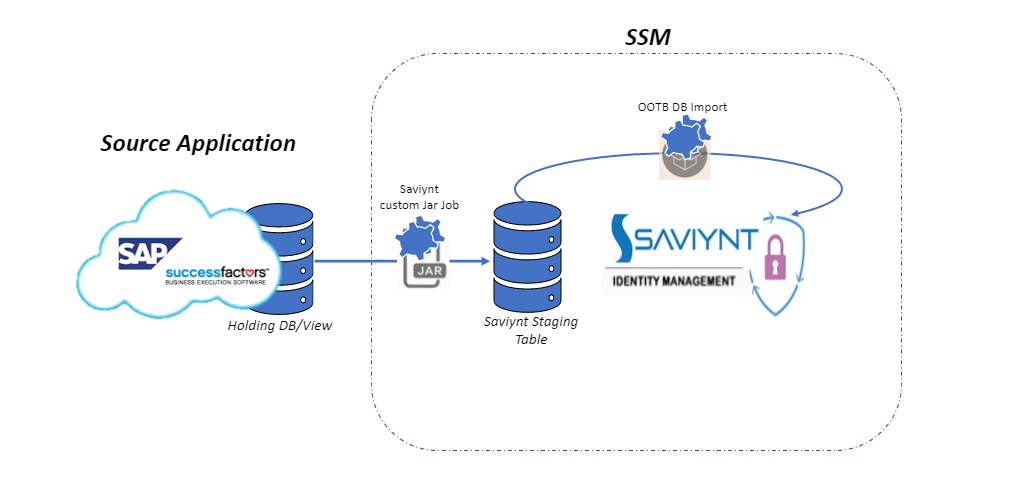
Attached below is the network details.



# Identity Imports:

There are two ways we can import the users into Saviynt

1. Employee onboarding
2. Non-Employee Creation



## Employee onboarding

### Prerequisites

The employee onboarding process contains trusted source as Database view, all the details of employee will be populated from the view source and scheduled job process will create the users in Saviynt.

* **Holding DB/View - VW\_SAVIYNT\_SAP\_RECON, VW\_SAVIYNT\_SAP\_RECON\_INCERMENTAL**  
  **(**which is the combination **of HOLD\_EMPL\_CURRENT** table and **HOLD\_EMPL\_HIRE** table)
* **Saviynt custom Jar Job** - It’s an External Jar job, which invokes custom utility mpc\_sap\_recon.jar. Which loads employee records from **Holding DB/View** to **Saviynt staging table i.e.. mpc\_Saviyntstagetable.**
* **Saviynt staging table -** Intermediate table in Saviynt database which holds all employee records.
* **OOTB DB connector** – Employee record is imported to Saviynt **User** table from staging table using an ootb database connector.

Two job schedulers are required for loading the employee data

|  |  |  |  |
| --- | --- | --- | --- |
| Job type | Job Name | Trigger chain job | Frequency |
| incremental | HRSAPTRUSTEDRECONINCREMENTAL | VIEW\_LOADER\_INCREMENTAL, DBATSAPTRUSTEDRECON | This job will run only for incremental users operations and it runs every 1 hour. |
| Full | HRSAPTRUSTEDRECON | VIEW\_LOADER , DBATSAPTRUSTEDRECON | This job run only once as full recon every weekend Saturday night to sync all the users from source to Saviynt. |

**Mpc\_picavaliable** table is the auto generated table with pic ID for new employees and MUID corresponding to the pic

|  |  |  |  |
| --- | --- | --- | --- |
| PIC | MUID | Reserved | Date Reserved |
| R501 | 1234556 | 1 | 2019-11-27 05:06:31 |

### Connection details

There are 2 database connections involved in this Employee onboarding

1. SAPHRTrustedRecon - OOTB database connector for complete user import from Saviynt staging table

|  |  |
| --- | --- |
| Connection Properties | Value |
| Connection Name | SAPHRTrustedRecon |
| URL | jdbc:mysql://10.220.116.90:3308/Saviyntdb |
| USERNAME | root |
| PASSWORD | \*\*\*\*\*\*\* |
| DRIVERNAME | Com.mysql.jdbc.Driver |
| USERIMPORT |  |

1. SAPHRTrustedReconIncremental - OOTB database connector for Incremental user import from Saviynt staging table

|  |  |
| --- | --- |
| Connection Properties | Value |
| Connection Name | SAPHRTrustedReconIncremental |
| URL | jdbc:mysql://10.220.116.90:3308/Saviyntdb |
| USERNAME | root |
| PASSWORD | \*\*\*\*\*\*\* |
| DRIVERNAME | Com.mysql.jdbc.Driver |
| USERIMPORT |  |

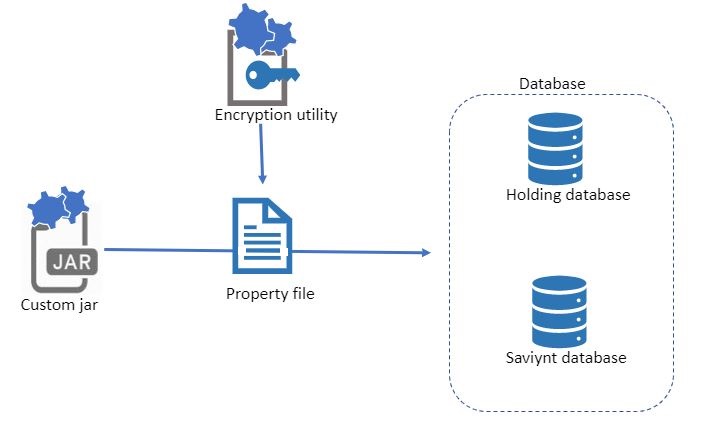
### Attribute Mapping

This is the current attribute mapping from source view to staging table and to Saviynt.

|  |  |  |  |
| --- | --- | --- | --- |
| HR View Attribute | Staging Table Attribute | Saviynt Database Attribute | UI labels |
| Actions\_CHANGEDON | Actions\_CHANGEDON | CustomProperty15 | Actions Changed ON |
| Actions\_BEGDA | Actions\_BEGDA | CustomProperty13 | Actions Beginning Date |
| Actions\_ENDDA | Actions\_ENDDA | CustomProperty14 | Actions End Date |
| calc in code | StartDate | STARTDATE | Start Date |
| by default, its set as 31/12/9999, if the user becomes inactive then current date and time is sent | EndDate | ENDDATE | End Date |
| EmployeeNumber | EmployeeNumber | EMPLOYEEID | Employee Number |
| LegacyEmployeeNumber | LegacyEmployeeNumber | CustomProperty24 | Legacy Employee Number |
| Auto generation from mpc\_picsavaliable table | Muid | CustomProperty3 | MUID |
| ActionType | ActionType | CustomProperty17 | Action Type |
| ActionStatus | ActionStatus | CustomProperty18 | Action Status |
| InitialHireDate | InitialHireDate | CustomProperty16 | InitialHireDate |
| DetailedStatus | DetailedStatus | CustomProperty4 | Detailed Status |
| 1 or 0 | EmployeeStatus | STATUSKEY | status |
| EmployeeType | EmployeeType | EMPLOYEETYPE | Employee Type |
| FirstName | FirstName | FIRSTNAME | First Name |
| LastName | LastName | LASTNAME | Last Name |
| MiddleName | MiddleName | MIDDLENAME | Middle Name |
| Suffix | Suffix | CustomProperty31 | Suffix |
| NickName | NickName | PREFERREDFIRSTNAME | Preferred First Name |
| NameFormat | NameFormat | CustomProperty32 | Name Format |
| BirthDate | BirthDate | CustomProperty2 | Birth Date(MMDD) |
| SSN | SSN | CustomProperty1 | Last 4 of SSN |
| PersonalCity | PersonalCity | CustomProperty27 | Personal City |
| PersonalState | PersonalState | CustomProperty28 | Personal State |
| CompanyCode | CompanyCode | ENTITY | Entity |
| CompanyName | CompanyName | COMPANYNAME | Company Name |
| OrgCode | OrgCode | DEPARTMENTNUMBER | Department Number |
| OrgName | OrgName | DEPARTMENTNAME | Department Name |
| Parent\_OrgName | Parent\_OrgName | CustomProperty34 | Parent Organization Name |
| PersonnelSubArea | PersonnelSubArea | CustomProperty23 | Personnel SubArea |
| CostCenter | CostCenter | COSTCENTER | Cost Center |
| EmployeeGroup | EmployeeGroup | CustomProperty21 | Employee Group |
| EmployeeSubGroup | EmployeeSubGroup | CustomProperty22 | Employee Sub Group |
| PositionCode | PositionCode | JOBCODE | Position Code |
| PositionName | PositionName | TITLE | Title |
| JobCode | JobCode | JOBFUNCTION | Job Function |
| JobName | JobName | JOBDESCRIPTION | Job Description |
| Chief | Chief | CustomProperty19 | Chief |
| UpperManagement | UpperManagement | CustomProperty20 | Upper Management |
| PrimaryLoginId | PrimaryLoginId | USERNAME | User Name |
| AlternateLoginId | AlternateLoginId | CustomProperty5 | Alternate Login |
| SAPAccount | SAPAccount | CustomProperty6 | SAP Account |
| EmailAddress | EmailAddress | EMAIL | Email |
| PhoneNumber | PhoneNumber | PHONENUMBER | Phone Number |
| PhoneExtension | PhoneExtension | CustomProperty25 | Phone Extension |
| MobileNumber | MobileNumber | SECONDARYPHONE | Mobile Number |
| PersonnelArea | PersonnelArea | LOCATION | Location |
| PersonnelAreaDescription | PersonnelAreaDescription | LOCATIONDESC | Location Desc |
| RoomNumber | RoomNumber | CustomProperty26 | Room Number |
| Country | Country | COUNTRY | Country |
| Street | Street | STREET | Street |
| City | City | CITY | City |
| State | State | STATE | State |
| PostalCode | PostalCode | CustomProperty29 | Postal Code |
| Supervisor\_EmployeeNumber | Supervisor\_EmployeeNumber | CustomProperty33 | Supervisor Employee Number |
| Supervisor\_PrimaryLoginId | Supervisor\_PrimaryLoginId | Owner | Supervisor PrimaryLoginId |
| PrimaryOU | PrimaryOU | CustomProperty7 | Primary OU |
| AlternateOU | AlternateOU | CustomProperty8 | Alternate OU |
| InternetAccess | InternetAccess | CustomProperty35 | Internet Access |
|  |  | CustomProperty9 | Vendor Email |
| MailEnabled | MailEnabled | CustomProperty36 | Mail Enabled |

### Database password encryption and decryption

Both Holding database and Saviynt database password in custom jar is encrypted using AES algorithm.



* **Encryption utility** – This is a custom utility used to encrypt the database password, build based on AES algorithm.

Below are the steps to get encrypted password

* 1. Login to any server as root user
  2. Run the below command   
     java –jar /opt/Saviynt/mpc\_hr\_sync/EncryptdbPassword.jar
  3. Enter the password in once it is prompted and hit **ENTER**
* **Property file** – Property file holds the encrypted password. It is placed in the server location /opt/Saviynt/mpc\_sap\_recon/Utility.properties  
  Below is the screenshot of Utility.properties file  
   Note: Use EncryptdbPassword.jar to get encrypted password.
* **Custom jar** – Custom jar fetches the encrypted password from property file and using decrypt method it decrypts the password to build a connection to database.

### Employee Creation: New Hire

The data of hiring a new employee comes from the source view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL from HR and the data is trusted and it can contain some null values.

1. The new user record will not have pic value and the employee number may or may not exist.
2. Our custom jar job First check the new record with EmpNo in the **HOLD\_EMPL\_CURRENT** table and if not found
3. Since **PIC**=NULL, the code logic will pic the record and check for the combination of LastName, Birthdate, SSN in the Saviynt table, if not found.
4. Then this is considered as new user and the new pic is assigned from **mpc\_picavaliable** table to new record and first it is updated in **HOLD\_EMPL\_CURRENT** table with pic and then the same record is inserted into **mpc\_Saviyntstagetable.**
5. Then **SAPHRTrustedRecon** job runs and create the new user with pic in Saviynt table.
6. After successful user creation in IAM, the user profile is available in IAM and the system will send user PIC and password format notification to the user’s manager via email.

### Employee Modification: Update

The update operation of employees comes from the HR source view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL from HR and the data is trusted and it can contain some null values too and some values are updated as per the users.

1. The new user record will not have pic value and the employee number may or may not exist.
2. Since **PIC**=exist, the code logic will search the user in Saviynt table with reference to the pic.
3. If found, then this is considered as update record and the user is updated in **mpc\_Saviyntstagetable.**
4. Then **HRSAPTRUSTEDRECON** job runs and updates the user with corresponding pic in Saviynt table.

Saviynt table

VIEW\_LOADER job

Source View

Staging table

DBATSAPTRUSTEDRECON ootb Connector

### Employee Modification: LOA Status

The update operation of employees comes from the HR source view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL from HR and the data is trusted and it can contain some null values too and some values are updated as per the users.

1. The new user record will not have pic value and the employee number may or may not exist.
2. Since **PIC**=exist, the code logic will search the user in Saviynt table with reference to the pic.
3. If found, then this is considered as update record and the user is updated in **mpc\_Saviyntstagetable.**
4. Then **HRSAPTRUSTEDRECON** job runs and updates the user with corresponding pic in Saviynt table.

|  |  |  |
| --- | --- | --- |
| View status | Saviynt (employee status) | Saviynt (Detailed status) |
| Active | Active | Active |
| LOA\_Active | Active | LOA\_Active |
| LOA\_Inactive | Active | LOA\_Inactive |
| Inactive | Inactive | Inactive |

Saviynt table

VIEW\_LOADER job

Source View

Staging table

DBATSAPTRUSTEDRECON ootb Connector

### Employee: Termination

The update operation of employees comes from the HR source view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL from HR and the data is trusted and it can contain some null values too and some values are updated as per the users.

1. The new user record will not have pic value and the employee number may or may not exist.
2. Since **PIC**=exist, the code logic will search the user in Saviynt table with reference to the pic.
3. If found, then this is considered as update record and the user is updated in **mpc\_Saviyntstagetable.**
4. Then **HRSAPTRUSTEDRECON** job runs and updates the user with corresponding pic in Saviynt table.
5. On an successful update operation the user will be inactive in Saviynt and all his access remains disabled.

|  |  |  |
| --- | --- | --- |
| View status | Saviynt (employee status) | Saviynt (Detailed status) |
| 1 | Active | Active |
| 0 | Inactive | Disabled |

Saviynt table

VIEW\_LOADER job

Source View

Staging table

DBATSAPTRUSTEDRECON ootb Connector

### Employee: Rehire

The update operation of employees comes from the HR source view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL from HR and the data is trusted and it can contain some null values too and some values are updated as per the users.

1. The new user record will not have pic value and the employee number may or may not exist.
2. Since **PIC** or **Employee NO** exist for the record, the code logic will search the user in Saviynt table.
3. If found, then and the user is inactive in Saviynt then this is considered as update operation where the (**source**)Employee status=’Active’ it is updated in **mpc\_Saviyntstagetable** as (**staging**) Employee status=’InActive’ and the record is updated.
4. Then **HRSAPTRUSTEDRECON** job runs and updates the user with corresponding pic in Saviynt table.
5. On a successful update operation, the user will be inactive in Saviynt and all his access will be enabled.

## Contractor/Non-Employee onboarding

### User types

The below are the mandatory attributes that MPC use for all the non-employee types that are going to be defined in Saviynt.

|  |  |
| --- | --- |
| User Types | Mandatory Attributes while Creating  Non-Employees |
| B - Batch ID | First Name |
| Contractor | Last Name |
| Function Outsourced Indefinitely | Organization Name |
| G - Group/Shared | Employee Type |
| Individual Independent Contractor – Self Employed | Last4 digits of ssn  Expiration Date  Manager  Mobile Number |
| Leased Workers | Birthdate (MMDD) |
| Outsourced Project - Billed on Hourly Basis | User Login  Location  Vendor Email |
| Full-Time |  |
| RPA Account |  |
| Speedway store account |  |
| Andeavor Contractor |  |
| Other |  |
| Outsourced Project - Billed on Project Basis |  |
| Test Non-Employee ID |  |

### Non-Employee Creation: New Hire

Users will be created from UI from ARS tab, in this tab only you can create a non-employee.

1. User will be created from UI by any MPC employee.
2. There are mandatory attributes like First Name, Last Name, Employee type, Last 4 Digits of SSN, Birthdate (MMDD), Organization, Location, Email needs to be filled and submitted.
3. After submitting the update rule triggers when the user is created from UI and it updates the **mpcpicavaliable table** as **reserved=1** where the pic= Userlogin of contractor from UI.
4. After successful user creation in IAM, the user profile is available in Saviynt and the system will send user PIC and password format notification to the user’s manager via email.

|  |  |
| --- | --- |
| Mandatory Attributes while Creating Non-Employees | Explanation |
| First Name | Enter First Name |
| Last Name | Enter Last Name |
| Manager | If Requester is manager then this option will show you with the drop down of REQUESTER, REQUESTER’s MANAGER, REQUESTERS MANAGER’s MANAGER, REQUESTERS BELOW MANAGER |
| Organization Name | Based on the manager selected his department will be visible. |
| Employee Type | Select Employee Type |
| Last4 digits of ssn | Enter SSN |
| Mobile Number | Enter 10-digit mobile number |
| Birthdate (MMDD) | Enter Birthdate |
| User Login | Select the auto generated login |
| Location | Select the location |
| Vendor Email | Enter the external email |

### Non-Employee Conversion: Contractor to Employee -Possibility 1

The contractor is converted to employee. the pic he uses while he is a contractor and if HR assigns same pic to source while he is hired.

1. All User record will be picked up from the view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL.
2. As per the code logic the taken record is searched with employee no in HOLD\_EMPL\_CURRENT\_TABLE if not found then.
3. The same record is searched in Saviynt table with the combination of Lastname, Birthdate, SSN if all 3 matches with the source record.
4. And the pic from the same source record is compared with the Saviynt ,if the above combination and the pic matches for the same user, then this is considered as correct record.
5. then this record is updated in **HOLD\_EMPL\_CURRENT** table with pic from Saviynt and the same record is inserted in **mpc\_Saviyntstagetable** as employee.
6. Then **HRSAPTRUSTEDRECON** job runs and updates the contractor with employee for corresponding pic in Saviynt table.

### Non-Employee Conversion: Contractor to Employee -Possibility 2

The contractor is converted to employee. he has a pic while he is a contractor and if HR didn’t assign pic to source while he is hired.

1. All User record will be picked up from the view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL.
2. As per the code logic the taken record is searched with employee no in HOLD\_EMPL\_CURRENT\_TABLE if not found then.
3. The same record is searched in Saviynt table with the combination of Lastname, Birthdate, SSN if all 3 matches with the source record.
4. And if the source record didn’t have the pic, then the pic from the Saviynt is assigned to the user.
5. then this record is updated in **HOLD\_EMPL\_CURRENT** table with pic from Saviynt and the same record is inserted in **mpc\_Saviyntstagetable** as employee.
6. Then **HRSAPTRUSTEDRECON** job runs and updates the contractor with employee for corresponding pic in Saviynt table.

### Non-Employee Conversion: Contractor to Employee -Possibility 3

The contractor is converted to employee, the pic he uses while he is a contractor and if HR assigns different pic to source while he is hired.

1. All User record will be picked up from the view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL.
2. As per the code logic the taken record is searched with employee no in HOLD\_EMPL\_CURRENT\_TABLE if not found then.
3. The same record is searched in Saviynt table with the combination of Lastname, Birthdate, SSN if all 3 matches with the source record.
4. And the pic from the same source record is compared with the Saviynt, if the above combination and the pic didn’t match for the same user, then this is considered as exceptional record.
5. then this record is updated in **HOLD\_EMPL\_CURRENT** table with pic from Saviynt and the same record is inserted in **mpc\_Saviyntstagetable** as employee with exception flag=1.
6. Then **HRSAPTRUSTEDRECON** job runs this record is not loaded into Saviynt table since this is exceptional, and it is sent in email as a report.

### Non-Employee Conversion: Contractor to Employee -Possibility 4

The contractor is converted to employee. He has a pic while he is a contractor and if HR didn’t assign pic to source while he is hired.

1. All User record will be picked up from the view VW\_SAVIYNT\_SAP\_RECON/VW\_SAVIYNT\_SAP\_RECON\_INCREMENTAL.
2. As per the code logic the taken record is searched with employee no in HOLD\_EMPL\_CURRENT\_TABLE if not found then.
3. The same record is searched in Saviynt table with the combination of Lastname, Birthdate, SSN if not found.
4. The same record is searched in Saviynt table with the combination of Lastname, birthdate and if found then this is considered as exceptional record.
5. then this record is updated in **HOLD\_EMPL\_CURRENT** table with pic from Saviynt and the same record is inserted in **mpc\_Saviyntstagetable** as employee with exception flag=1.
6. Then **HRSAPTRUSTEDRECON** job runs this record is not loaded into Saviynt table since this is exceptional, and it is sent in email as a report.

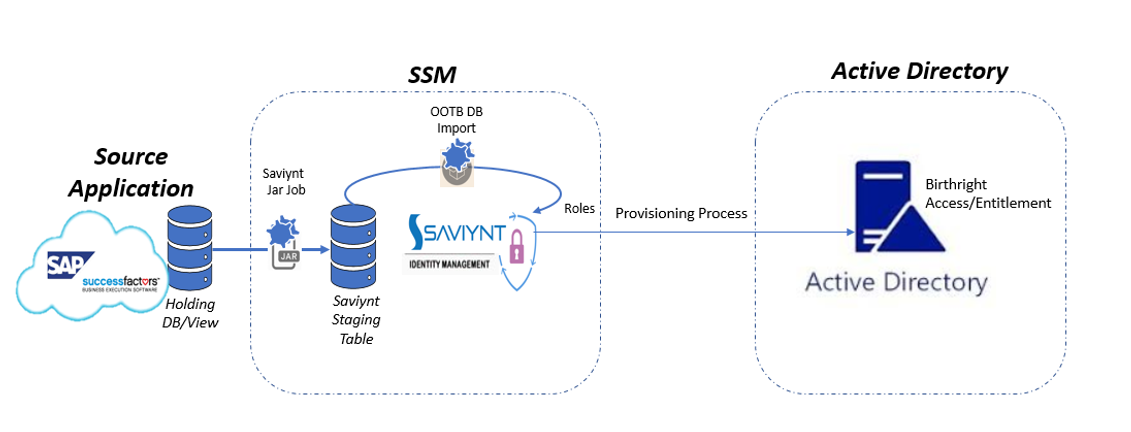
# Target System

## Active Directory

MPC has one instances of the Active Directory systems. This application would be considered as the Target systems which would be enabled for the Account reconciliation and Account Provisioning / de-Provisioning.

SSM uses the Active Directory OOTB connector for the Account reconciliation.

### Integration Design



### Connection details

This is the connection details for the Active Directory connector.

|  |  |
| --- | --- |
| Connection Properties | Value |
| URL | ldap://<Hostname>:<port>  ldaps://MTS1000.MTESTNET.COM:636 |
| USERNAME | Service Username  CN=IAM\_AD,OU=Batch,OU=Security,DC=mtestnet,DC=com |
| LDAP\_OR\_AD | AD  AD |
| SEARCHFILTER | <LDAP OU for the user and group search>  DC=mtestnet,DC=com |
| OBJECTFILTER | <Any filter>  (&(objectClass=person)(objectClass=user)(sAMAccountName=\*)) |
| ACCOUNT\_ATTRIBUTE | Attribute Mapping.  [CUSTOMPROPERTY1::cn#String,  STATUS::userAccountControl#number,  CUSTOMPROPERTY30::userAccountControl#String,  CUSTOMPROPERTY2::userPrincipalName#String,  CUSTOMPROPERTY28::primaryGroupID#String,  LASTLOGONDATE::lastLogon#millisec,  DISPLAYNAME::name#String,  CUSTOMPROPERTY25::company#String,  CUSTOMPROPERTY20::employeeID#String,  CUSTOMPROPERTY3::sn#String,  COMMENTS::distinguishedName#String,  CUSTOMPROPERTY4::homeDirectory#String,  LASTPASSWORDCHANGE::pwdLastSet#millisec,  CUSTOMPROPERTY5::co#String,  CUSTOMPROPERTY6::employeeNumber#String,  CUSTOMPROPERTY7::givenName#String,  CUSTOMPROPERTY8::title#String,  CUSTOMPROPERTY9::telephoneNumber#String,  CUSTOMPROPERTY10::c#String,  DESCRIPTION::description#String,  CUSTOMPROPERTY11::uSNCreated#String,  VALIDTHROUGH::accountExpires#millisec,  CUSTOMPROPERTY12::logonCount#String,  CUSTOMPROPERTY13::physicalDeliveryOfficeName#String,  UPDATEDATE:whenChanged#date,  CUSTOMPROPERTY14::extensionAttribute1#String,  CUSTOMPROPERTY15::extensionAttribute2#String,  CUSTOMPROPERTY16::streetAddress#String,  CUSTOMPROPERTY17::mailNickname#String,  CUSTOMPROPERTY18::department#String,  CUSTOMPROPERTY19::countryCode#String,  NAME:sAMAccountName#String,  CUSTOMPROPERTY21::manager#String,  CUSTOMPROPERTY22::homePhone#String,  CUSTOMPROPERTY23::mobile#String,  CREATED\_ON::whenCreated#date,  ACCOUNTCLASS::objectClass#String,  ACCOUNTID::distinguishedName#String,  CUSTOMPROPERTY24::userAccountControl#String,  CUSTOMPROPERTY27::objectSid#Binary,  RECONCILATION\_FIELD::CUSTOMPROPERTY26,  CUSTOMPROPERTY26::objectGUID#Binary,  CUSTOMPROPERTY29::st#String] |
| ENTITLEMENT\_ATTRIBUTE | memberOf |

### Attribute mapping

This is the current attribute mapping from Saviynt to target

|  |  |
| --- | --- |
| Saviynt Attributes | Active Directory Attributes |
| user login | cn |
| firstname or preferredfirstname | givenname |
| last name | sn |
| middle name | middlename |
| password | unicodepwd |
| displayname | displayname |
| mailenabled (customproperty36) | extentionattribute13 |
| primaryou | ou |
| [username@mgroupnet.com](mailto:USERNAME@mgroupnet.com) | userprinciplename |
| username | samaccountname |
| Set as 0 (User must change Password at Next logon) | pwdlastset |
| manageraccount?.accountid | manager |
| company name | company |
| department name | department |
| (standard ad field, based on user and ou mapping) | distinguishedname |
| employeetype | division |
| legacyemployeenumber (customproperty24) | employeeid |
| employeenumber (employeeid) | employeenumber |
| static value after m:\ provisioning | homedirectory |
|  | homedrive |
| middlename (middle initial) | initials |
| username | mailnickname |
| mobilenumber (secondaryphone) | mobile |
| username | name |
| postalcode (customproperty29) | postalcode |
| state | st |
| street | streetaddress |
| phonenumber | telephonenumber |
| title | title |
| muid (customproperty3) | extensionattribute6 |
| city | l |
| Roomnumber (customproperty26) | physicalDeliveryOfficeName |
| Samaccountname (customproperty6) | uid |
| Wel\_2M$+MMDD (customproperty2) | unicodePwd |

### Account provisioning

* Saviynt uses Birthright provisioning method for user AD account provisioning
* As per requirement we have defined two technical rules in SSM for Active Directory target
  + Technical Rule – **MB\_AD Birthright Rule**

Condition: **Users. Status = ‘Active’**

Birthright Role - **MGROUPNET AD Basic Account Provisioning**

Birthright Access/Entitlement **- CN=ManagedUsers,OU=FDY Groups,OU=Delegated,DC=mtestnet,DC=com**

* + Technical Rule – **MB\_AD Internet Access Rule**

Condition: **Users.Usertype = ’EMPLOYEE’ OR Users.InternetAccess = ‘1’**

Birthright Role -  **MGROUPNET AD Internet Access**

Birthright Access/Entitlement- **CN=MTESTNET PRX PROXY WWW,OU=MTESTNET,OU=People,DC=mtestnet,DC=com**

#### Account creation

* As soon as user satisfy active directory birthright then SSM will trigger AD birthright process.
* As per defined AD birthright access ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioning process Create AD account and Add Birthright Access to the user account in AD target
* New account created for user with ‘Manually Provisioned’ status in Saviynt
* Set default password as per Marathon password policy I.e. Wel\_2M$DDMM
* After User Provisioned to AD then Email Notification will send the user’s Manager With User Name and Password fields
* Create Account JSON :

|  |
| --- |
| {  "cn":"${user.username}",  "company":"${user.companyname}",  "department":"${user.departmentname}",  "displayname":"${user.displayname}",  "employeeID":"${user.customproperty24}",  "employeeNumber":"${'0'+user.employeeid}",  "extensionAttribute13":"${if(user.customproperty36 == '1'){'1'}else{'0'}}",  "extensionAttribute6":"${user.customproperty3}",  "givenName":"${user.preferedFirstName}",  "initials":"${if(user.displayname == null){user.middlename.substring(0,1).toUpperCase()+'.'}else{user.preferedFirstName}}",  "l":"${user.city}",  "manager": "${managerAccount?.accountID}",  "mobile":"${user.secondaryPhone}",  "physicalDeliveryOfficeName":"${user.customproperty26}",  "postalCode":"${user.customproperty29}",  "sAMAccountName":"${user.username}",  "sn":"${user.lastname}",  "st":"${user.state}",  "streetAddress":"${user.street}",  "telephoneNumber":"${user.phonenumber}",  "title":"${user.title}",  "uid":"${user.customproperty6}",  "unicodePwd":"${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}",  "pwdLastSet": "0",  "objectClass":[ "top","person", "organizationalPerson", "user" ]  } |

#### Account update

* We have Created ‘**User Update Rule’** in Saviynt to update AD attributes in target
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘**Update Account**’ task for AD
* User attribute Changes like:
  + Username
  + Firstname
  + Lastname
  + Middlename
  + Customproperty36(Mail Enabled)
  + Companyname
  + DepartmentNumber
  + Customproperty40(Mobile)
  + Customproperty43(Postal Code)
  + State
  + Street
  + Phonenumber
  + Customproperty7(Primary OU)
* Execute the Provisioning job to push the update to the target
* Update Account JSON:

|  |
| --- |
| {  "cn":"${user.username}",  "company":"${user.companyname}",  "department":"${user.departmentname}",  "displayname":"${user.displayname}",  "employeeID":"${user.customproperty24}",  "employeeNumber":"${'0'+user.employeeid}",  "extensionAttribute13":"${if(user.customproperty36 == '1'){'1'}else{'0'}}",  "extensionAttribute6":"${user.customproperty3}",  "givenName":"${user.preferedFirstName}",  "initials":"${if(user.displayname == null){user.middlename.substring(0,1).toUpperCase()+'.'}else{user.preferedFirstName}}",  "l":"${user.city}",  "manager": "${managerAccount?.accountID}",  "mobile":"${user.secondaryPhone}",  "physicalDeliveryOfficeName":"${user.customproperty26}",  "postalCode":"${user.customproperty29}",  "sAMAccountName":"${user.username}",  "sn":"${user.lastname}",  "st":"${user.state}",  "streetAddress":"${user.street}",  "telephoneNumber":"${user.phonenumber}",  "title":"${user.title}",  "uid":"${user.customproperty6}",  "unicodePwd":"${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}",  "pwdLastSet": "0",  "objectClass":[ "top","person", "organizationalPerson", "user" ]  } |

#### LOA Use Cases

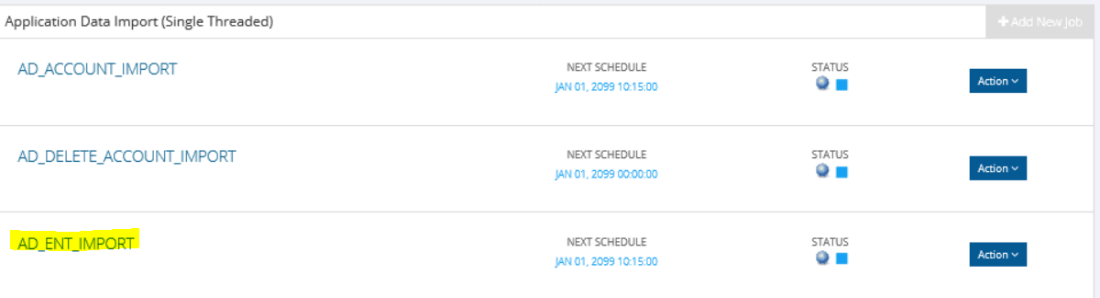
* When the ‘Detailed status’(CustomProperty4) flag updates then we need have changes to be performed on account
* ‘User Update rules’ will be trigger according to the updates in ‘Detailed status’ fields
* And it creates tasks accordingly (Disable account and Enable Account)
* After provisioning job status of the user account will get change.

|  |  |  |
| --- | --- | --- |
| View Status | Saviynt Status | AD |
| Active | Active | Active |
| LOA Active | Active | Active |
| LOA Inactive | Active | Inactive |
| Inactive | Inactive | Inactive |

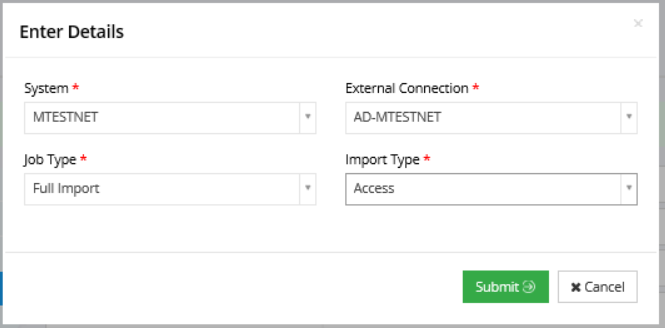
### Account Access provisioning

#### AD group management

* User should not able to submit active directory group/access request within SSM
* All group/access added through different roles
* To Pull all groups from AD to SSM, we need to trigger a job - Application Data Import (Single Threaded)



* Need to provide job parameter like System, Connection Name, Job Type, Import Type

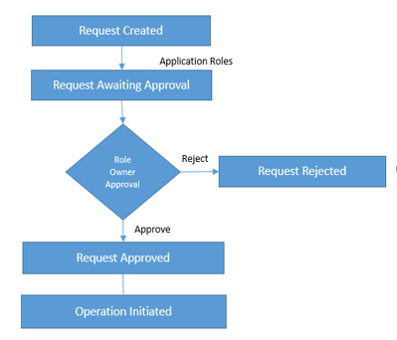


* Group Import Mapping JSON

|  |
| --- |
| {  "importGroupHierarchy": "true",  "entitlementTypeName": "Group",  "performGroupAccountLinking": "true",  "incrementalTimeField": "whenChanged",  "groupObjectClass": "(objectclass=group)",  "mapping": "memberHash:member\_char,customProperty1:sAMAccountType\_char,customProperty2:instanceType\_char,customProperty3:uSNCreated\_char,customProperty4:groupType\_char,customProperty5:dSCorePropagationData\_char,customProperty12:dn\_char,customProperty13:cn\_char,lastscandate:whenCreated\_date,customProperty15:managedBy\_char,entitlement\_glossary:description\_char,description:description\_char,displayname:name\_char,customProperty9:name\_char,customProperty10:objectCategory\_char,customProperty11:sAMAccountName\_char,entitlement\_value:distinguishedName\_char,entitlementid:distinguishedName\_char,customProperty14:objectClass\_char,updatedate:whenChanged\_date,customProperty17:distinguishedName\_char,RECONCILATION\_FIELD:customProperty17,customProperty18:objectGUID\_Binary",  "activeGroupPossibleValues": [  "active",  "a",  "l",  "TRUE"  ]  } |

#### Approval workflow

* When the user requested for any role then as per role approval workflow request will submit to role owner
* Once the role owner approves the request ‘Add Access’ task will create in Saviynt
* After provisioning job, role will add to the user along with different role access
* Role Owner Approval Workflow:



### Account Termination

* Created ‘user update rule’ to disable the user account in Saviynt
* We have added below condition like:
* When Users.Status =0 AND Users.Status is Updated
* Then Trigger ‘Disable User Account task’
* On user Disable this rule will execute which will create the Disable account task for account
* After provisioning job, it will Disable the account in AD i.e UserAccountControl Status =514
* Disable account JSON:

|  |
| --- |
| {  "moveUsertoOU": "OU=Disabled Users,DC=mtestnet,DC=com",  "deleteAllGroups": "No",  "userAccountControl": "514"  } |

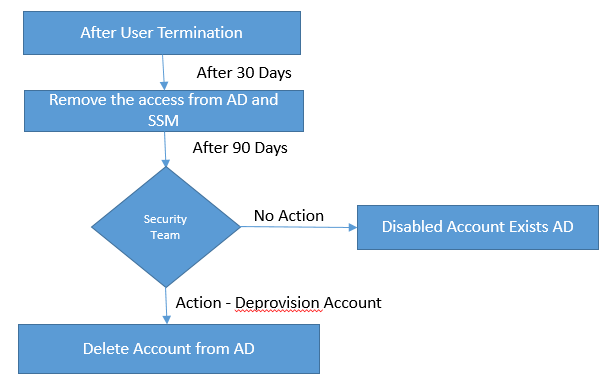
* Once the user account terminated, user’s active directory account will be moving into disabled users OU.

OU=Disabled Users,DC=mtestnet,DC=com

### Account De-provisioning

User Active Directory Account Deprovisioning process contains below 3 Steps

|  |  |
| --- | --- |
| User Status/Days | Action on AD account from Saviynt on Term Users |
| Inactive | Only Disable user AD account |
| Inactive -After 30 Days | Remove Associated Entitlements/access from user AD account |
| Inactive -After 90 Days | Send Email notification of Analytics Report to Security team to act on the 90 days AD accounts |
| Inactive- After 90 Days (Action by Security team) | Based on Security Team action remove AD account from target end |

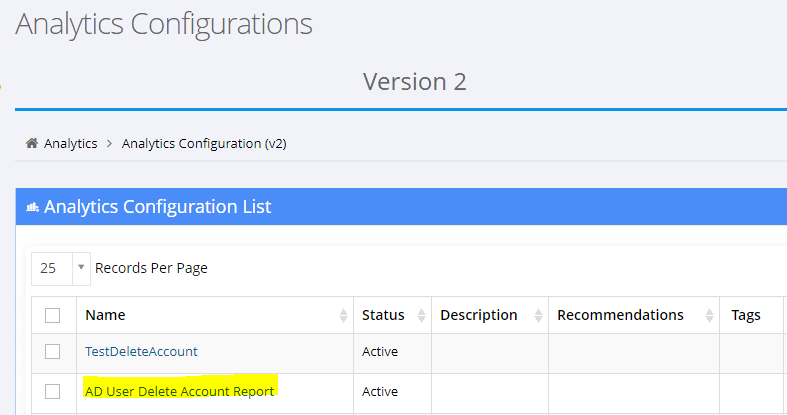


1. As soon as user Inactive then Terminate/Disable User AD account in target

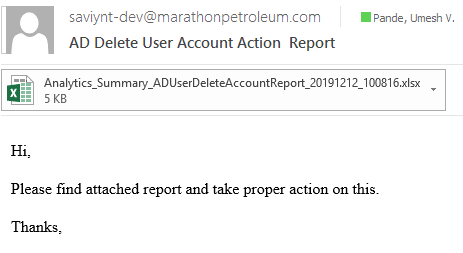
* Once the user disabled after 30 days account should lose the groups attached to it and ‘remove access task’ will be launch after disable user and disable account through an update of CUSTOMPROPERTY50 in the Saviynt
* Which we are doing through database import job
* After that we don’t want to delete active directory user account after 90 days
* Security team will be going to decide whether they want to delete user AD account from AD target or not
* To perform this process, we have created Analytic report in SSM which will going to send delete account information based on 90 days criteria through email notification to security team.
* Security Team will be going to act on these AD account by login into SSM.
* Remove Account Action Json:

|  |
| --- |
| {  "removeAction": "${if((new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(new Date())).getTime()-new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(user.enddate)).getTime())/(1000\*60\*60 \* 24)>=90) {'DELETE'} else {'SUSPEND'}}",  "deleteAllGroups":"${if((new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(new Date())).getTime()-new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(user.enddate)).getTime())/(1000\*60\*60 \* 24)>=30 && (new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(new Date())).getTime()-new java.text.SimpleDateFormat("yyyy-MM-DD").parse(new java.text.SimpleDateFormat("yyyy-MM-DD").format(user.enddate)).getTime())/(1000\*60\*60 \* 24)<=90){'Yes'} else {'No'}}"  } |

* Active Directory users delete account Analytic report
* We are creating this report for users those are disabled in the past 90 days



* Email Notification for



### Account Re-Hire

|  |  |
| --- | --- |
| User Status/Days/Action | Action on AD account from Saviynt on Rehire User |
| Inactive – Rehire Before 30 Days | Process will Enable user in Saviynt and enable respective AD account in target |
| Inactive - Rehire After 30 Days | Process will Enable user in Saviynt and enable respective AD account in target along with all access/groups |
| Inactive - Rehire After 90 Days | Process will Enable user in Saviynt and Create respective AD account in target if it’s not existed along with all birthright access/groups |
| Inactive - Rehire After 90 Days | Process will Enable user in Saviynt and Enable respective AD account in target if it exists along with all access/groups |

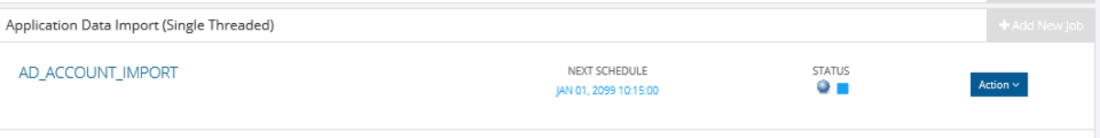
* If the user re-joins the MPC, based on his end date certain actions will take place
* Created ‘user update rule’ in Saviynt(Users.Status = ‘1’ AND Users.Status is Updated)
* If the User re-joins the MPC before 30 days ‘Enable Account’ task will create in SSM
* After provisioning job, it will activate the account in AD
* If the User re-joins the MPC after 30 days ‘Enable Account’, 2 ‘add access’ task will create in SSM
* After provisioning job, it will activate the account in AD along with access
* If the User re-joins the MPC after 90 days and account not exists in AD then ‘New Account’, 2 ‘add access’ task will create in SSM
* After provisioning job, New account will create in AD along with access
* If the User re-joins the MPC after 90 days and account exists in AD then ‘Enable Account’, 2 ‘add access’ task will create in SSM
* Once the account is enabled, Account OU will move based on the Primary OU
* New Account will create in Respective OU

Enable Account JSON:

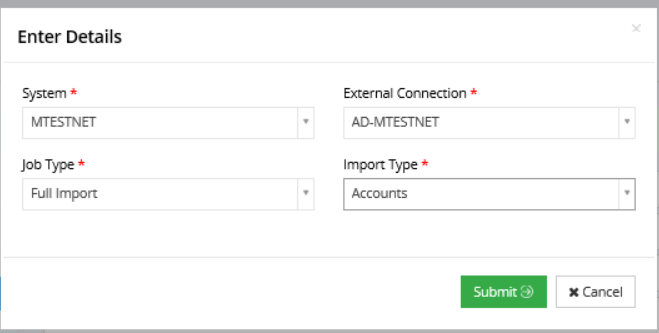
|  |
| --- |
| {  "USEDNFROMACCOUNT": "YES",  "MOVEDN": "YES",  "REMOVEGROUPS": "No",  "ENABLEACCOUNTOU": "${if(user.customproperty7 != null && user.customproperty7 != ''){user.customproperty7+',DC=mtestnet,DC=COM'} else {'OU=FDY,OU=People,DC=mtestnet,DC=COM'}}",  "AFTERMOVEACTIONS": {  "userAccountControl": "512"  }  } |

### Account Target recon

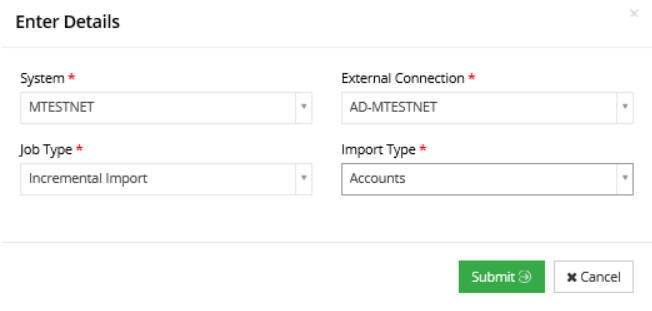
* We have configured target recon process in SSM
* We can do Incremental as well as Full import to recon active directory target user accounts into SSM
* Created Target recon job to import Accounts from AD
* Based on the Correlation rule accounts will sync with SSM



* Full Account Import



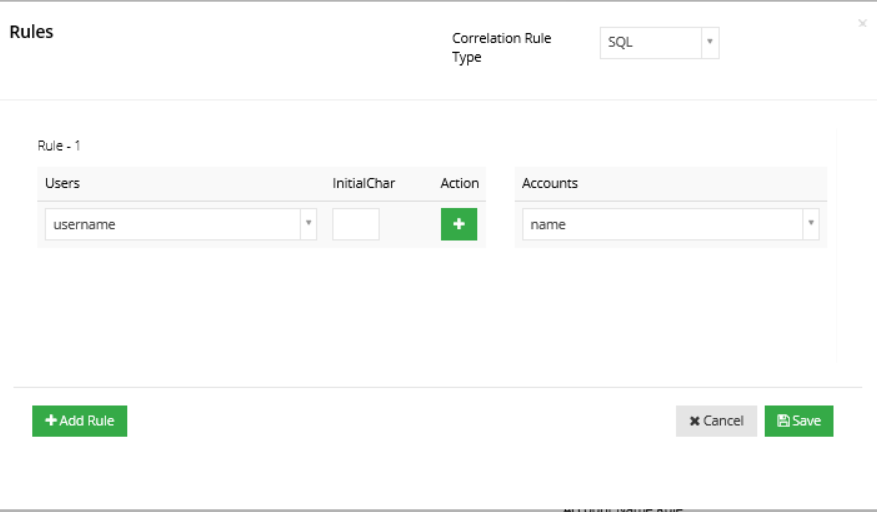
* Incremental Account Import



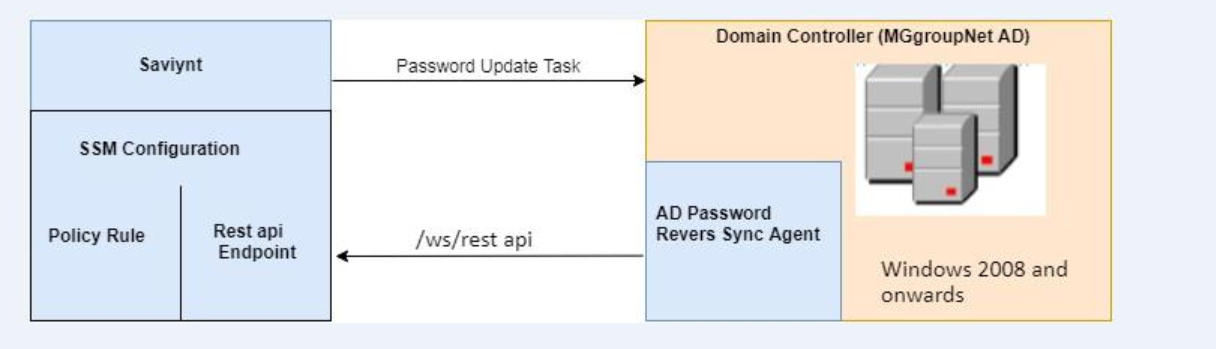
|  |  |
| --- | --- |
| Saviynt Attributes | AD Attributes |
| CUSTOMPROPERTY1 | cn |
| STATUS | userAccountControl |
| CUSTOMPROPERTY30 | userAccountControl |
| CUSTOMPROPERTY2 | userPrincipalName |
| CUSTOMPROPERTY28 | primaryGroupID |
| LASTLOGONDATE | lastLogon |
| DISPLAYNAME | name |
| CUSTOMPROPERTY25 | company |
| CUSTOMPROPERTY20 | employeeID |
| CUSTOMPROPERTY3 | sn |
| COMMENTS | distinguishedName |
| CUSTOMPROPERTY4 | homeDirectory |
| LASTPASSWORDCHANGE | pwdLastSet |
| CUSTOMPROPERTY5 | co |
| CUSTOMPROPERTY6 | employeeNumber |
| CUSTOMPROPERTY7 | givenName |
| CUSTOMPROPERTY8 | title |
| CUSTOMPROPERTY9 | telephoneNumber |
| CUSTOMPROPERTY10 | c |
| DESCRIPTION | description |
| CUSTOMPROPERTY11 | uSNCreated |
| VALIDTHROUGH | accountExpires |
| CUSTOMPROPERTY12 | logonCount |
| CUSTOMPROPERTY13 | physicalDeliveryOfficeName |
| UPDATEDATE | whenChanged |
| CUSTOMPROPERTY14 | extensionAttribute1 |
| CUSTOMPROPERTY15 | extensionAttribute2 |
| CUSTOMPROPERTY16 | streetAddress |
| CUSTOMPROPERTY17 | mailNickname |
| CUSTOMPROPERTY18 | department |
| CUSTOMPROPERTY19 | countryCode |
| NAME | cn |
| CUSTOMPROPERTY21 | manager |
| CUSTOMPROPERTY22 | homePhone |
| CUSTOMPROPERTY23 | mobile |
| CREATED\_ON | whenCreated |
| ACCOUNTCLASS | objectClass |
| ACCOUNTID | distinguishedName |
| CUSTOMPROPERTY24 | userAccountControl |
| CUSTOMPROPERTY27 | objectSid |
| RECONCILATION\_FIELD | CUSTOMPROPERTY26, |
| CUSTOMPROPERTY26 | objectGUID |
| CUSTOMPROPERTY29 | st |
| CUSTOMPROPERTY40 | mail |

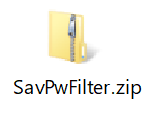
#### Correlation link

* Username (In SSM) -------> name(In AD)



**6.2.0 AD Password Reverse sync**

* We have configured ‘AD Password Revers Sync Agent’ on Active Directory Domain Controller
* Password Revers Sync agent will be going to sync AD password with Saviynt
* This agent users Rest Webservice to sync AD password with Saviynt
* AD password revers sync process flow:
* Password Filter Zip is shared along with the document.

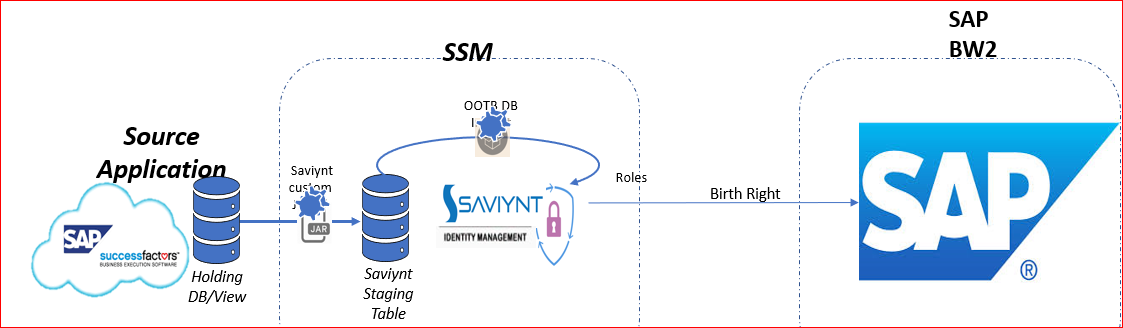


## SAP BW2

MPC has SAP BW2 as of one of the target systems. In this application SSM will provision all Employee to the BW2 along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | sapbw2.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | BW2 |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3620 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |  |
| --- | --- | --- |
| Saviynt Attributes | SAP BW2 Attributes | |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP BW2 attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP BW2
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Access provisioning

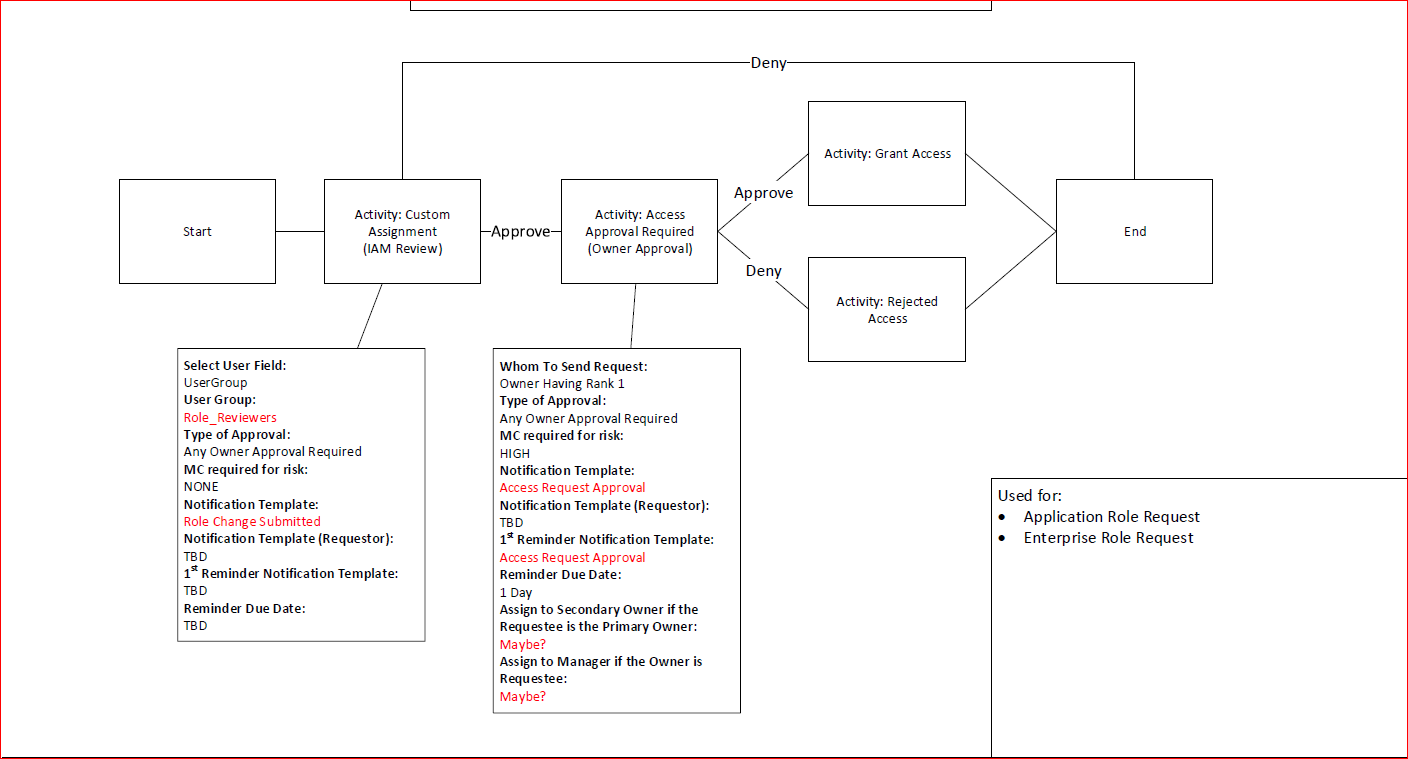
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in BW2
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPBWPIMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

#### Correlation link

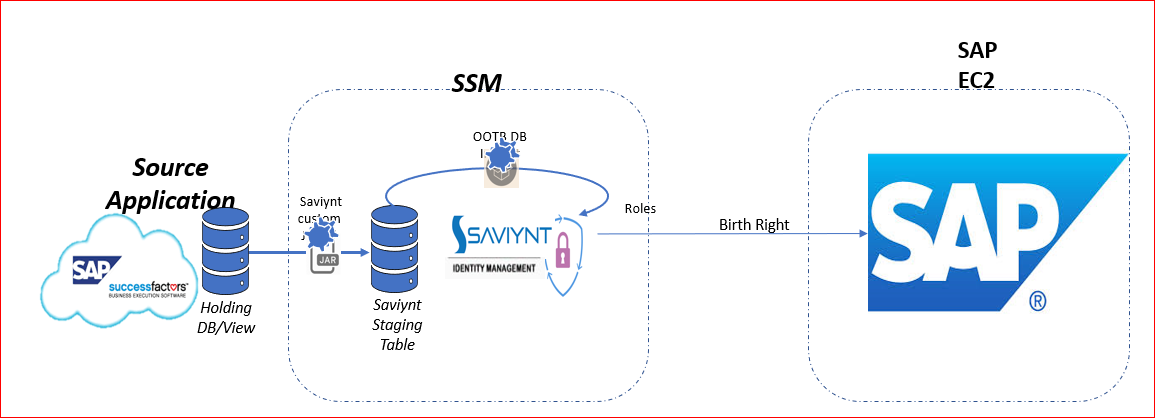
* Customproperty6(SAP Account ID) AccountName

## SAP EC2

MPC has SAP EC2 as of one of the target system. In this application SSM will provision all Employee to the EC2 along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | sapec2.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | EC2 |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3600 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |
| --- | --- |
| Saviynt Attributes | SAP EC2 Attributes |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  },  "PARAMETER": {  "CVR": "100",  "UGR": "10"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP EC2 attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP EC2
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  },  "PARAMETER": {  "CVR": "100",  "UGR": "10"  }    } |

### Account Access provisioning

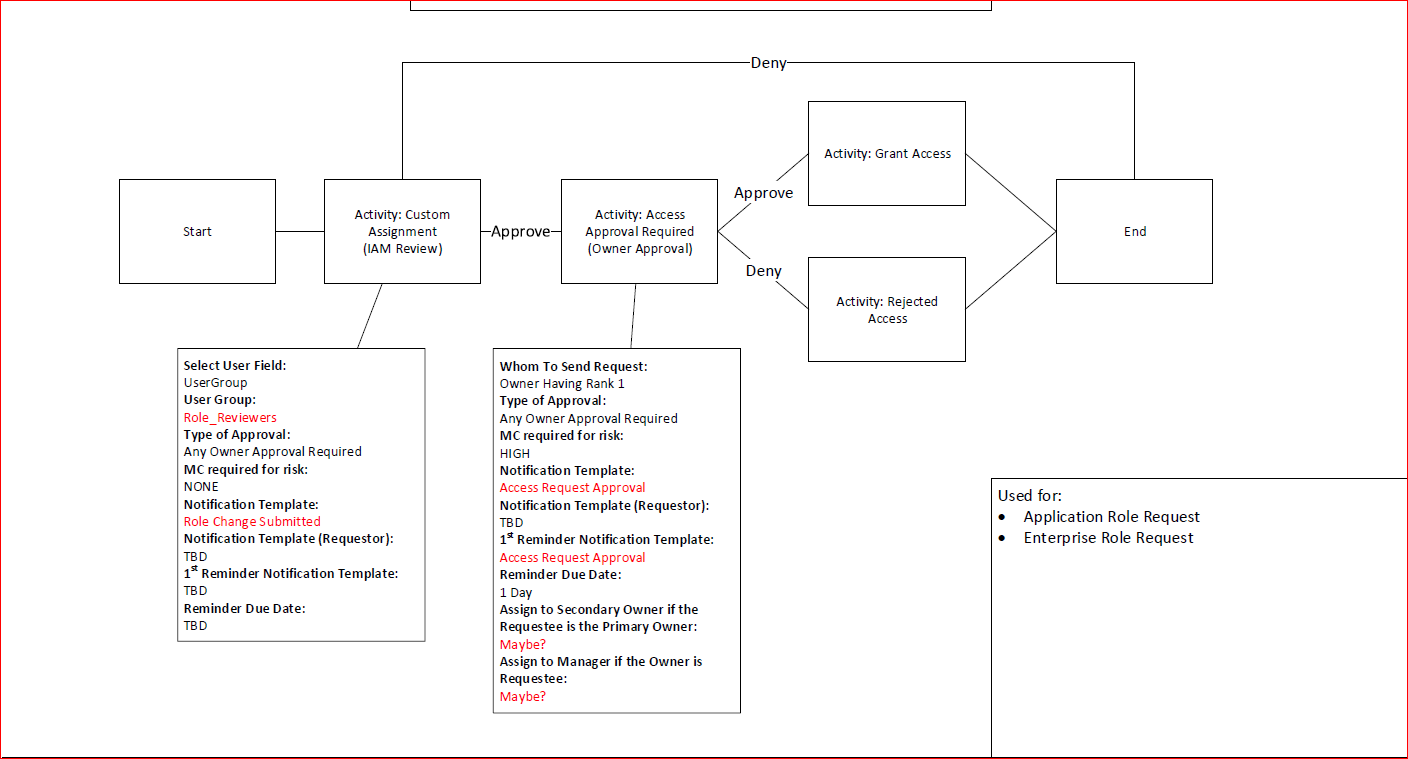
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in EC2
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPEC2IMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

#### Correlation link

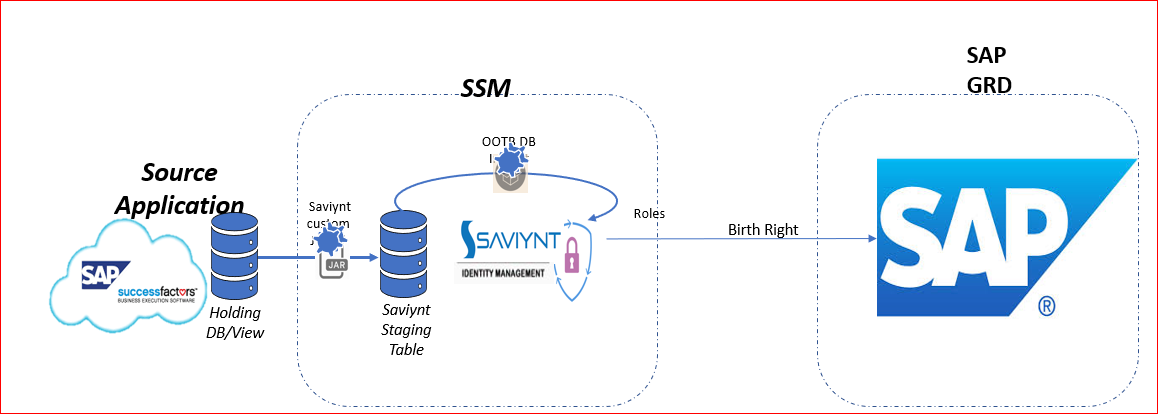
* Customproperty6(SAP Account ID) AccountName

## SAP GRD

MPC has SAP GRD as of one of the target system. In this application SSM will provision all Employee to the GRD along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | grd.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | GRD |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3620 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |
| --- | --- |
| Saviynt Attributes | SAP GRD Attributes |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP GRD attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP GRD
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Access provisioning

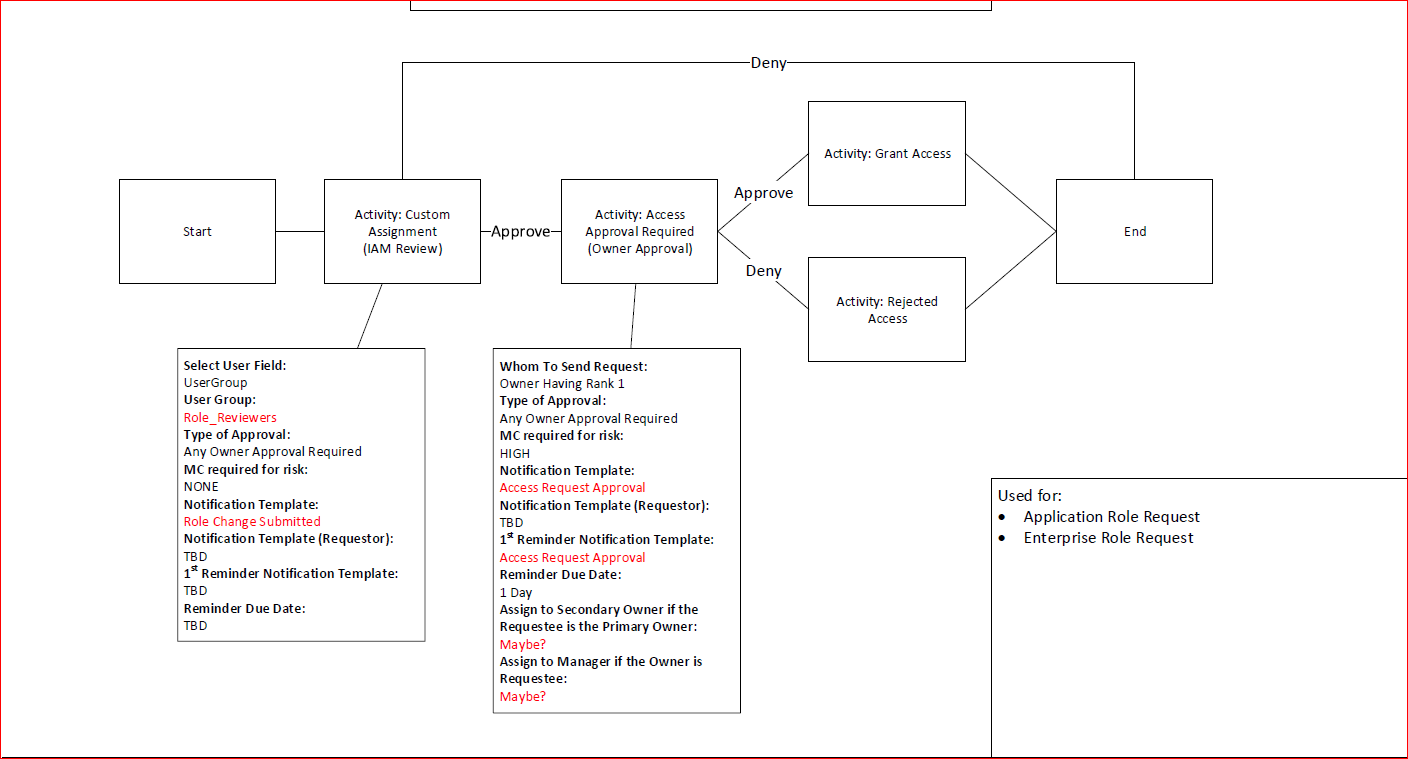
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in GRD
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPGRDIMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

#### Correlation link

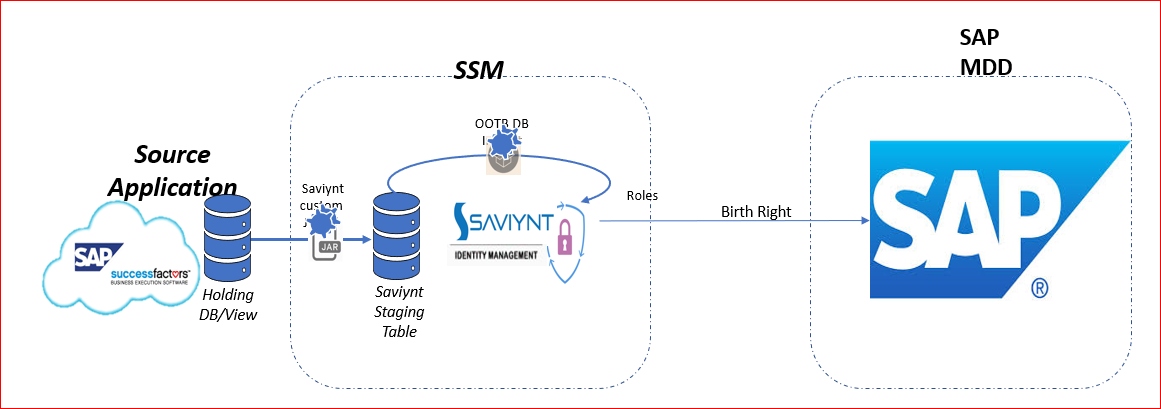
Customproperty6(SAP Account ID) AccountName

## SAP MDD

MPC has SAP MDD as of one of the target system. In this application SSM will provision all Employee to the MDD along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | sapmdd.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | MDD |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3600 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |
| --- | --- |
| Saviynt Attributes | SAP MDD Attributes |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP MDD attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP MDD
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Access provisioning

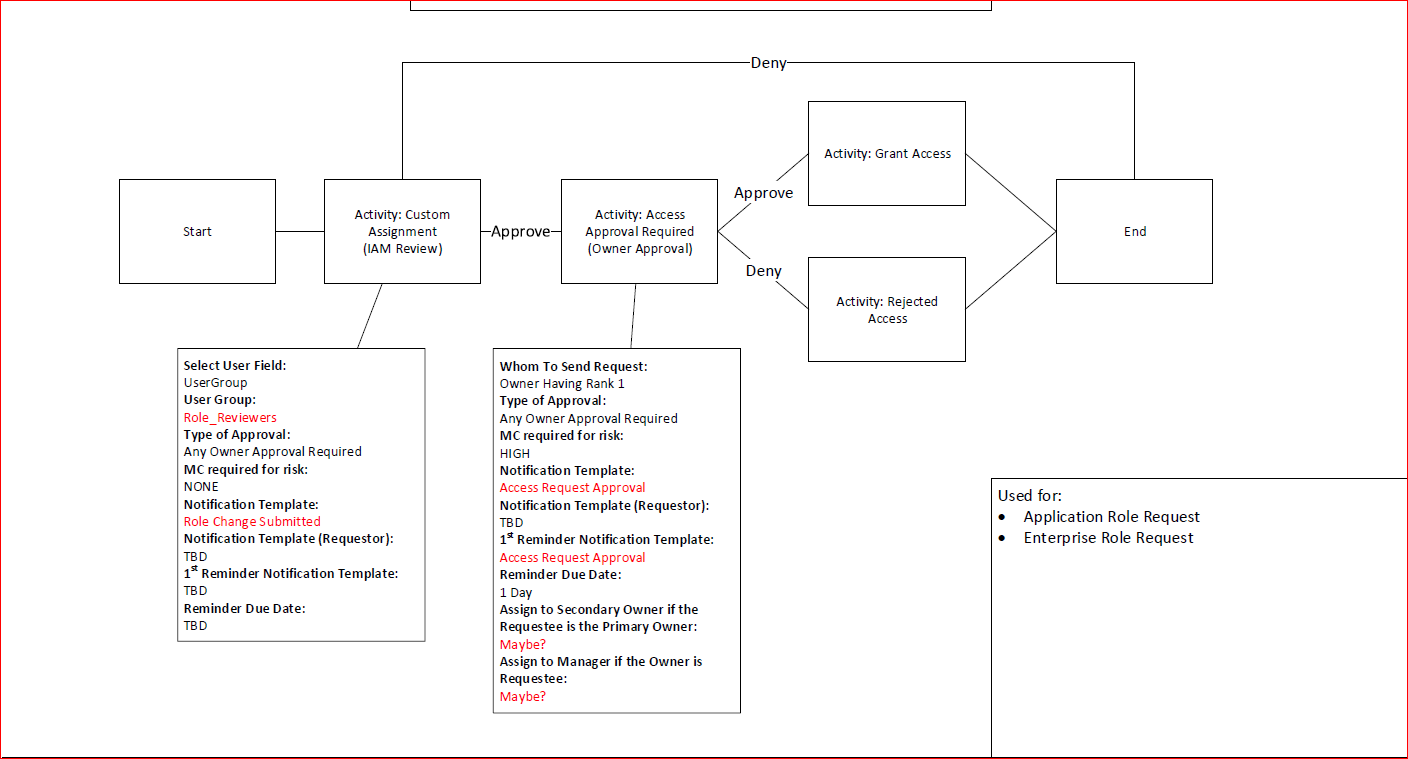
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in MDD
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPMDDIMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

#### Correlation link

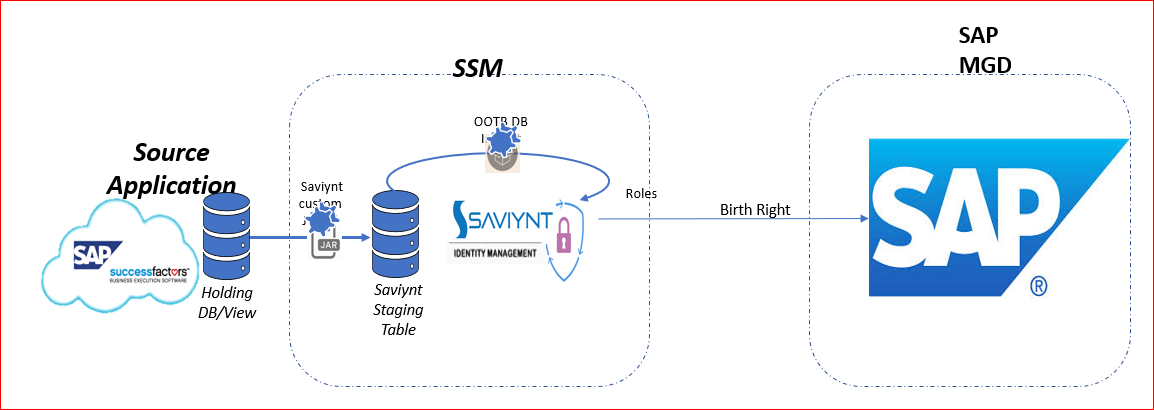
Customproperty6(SAP Account ID) AccountName

## SAP MGD

MPC has SAP MGD as of one of the target system. In this application SSM will provision all Employee to the MGD along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | sapmgd.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | MGD |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3600 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |
| --- | --- |
| Saviynt Attributes | SAP MGD Attributes |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP MGD attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP MGD
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Access provisioning

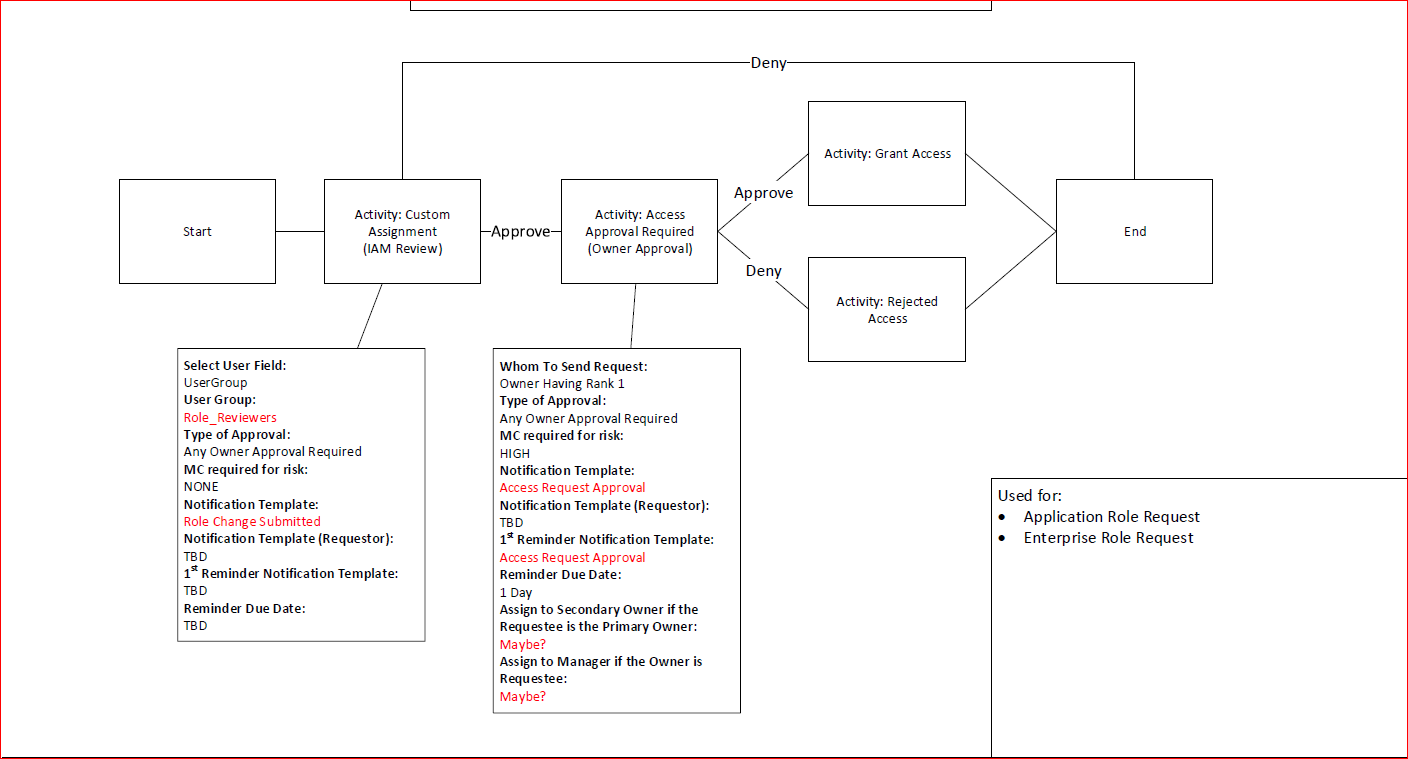
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in MGD
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPMGDIMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

#### Correlation link

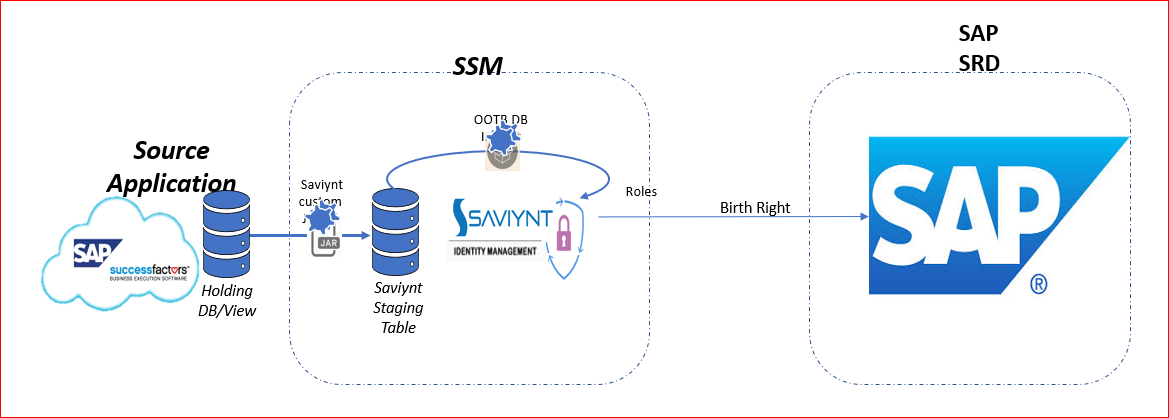
Customproperty6(SAP Account ID) AccountName

## SAP SRD

MPC has SAP SRD as of one of the target system. In this application SSM will provision all Employee to the SRD along with access roles.

SSM uses the Out of the box SAP connector to perform all the joiner/Mover/Leaver use cases.

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Value |

|  |  |
| --- | --- |
| URL | sapsrd.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | SRD |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3600 |

### Attribute mapping

* Attribute mapping for source to target

|  |  |
| --- | --- |
| Saviynt Attributes | SAP SRD Attributes |
| First Name | | First Name |
| Last Name | | Last Name |
| Email | | E\_MAIL |
| Country | | NAMCOUNTRY |
| Department | | Department |
| Room Number | | ROOM\_NO\_P |
| Telephone Number | | TEL1\_NUMBR |
| User Password | | Password |

### Account provisioning

#### Account creation

* Saviynt uses BirthRight access provisioning method for account provisioning
* BirthRight Role - DEV-Employee-SAP-Basic Account Provisioning
* Technical Rule - (Users. Status=‘Active’) AND (Users. User Type=Employee)
* SAP birthright process trigger ‘**New Account**’ and ‘**Add Access**’ tasks in Saviynt
* Provisioned SAP account and birthright Access to the new employee with ‘Manually Provisioned’ status in Saviynt
* Create SAP account and Add Birthright Role to the target
* Set default password
* Create Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }  } |

#### Account update

* Created ‘**User Update Rule’** in Saviynt to update SAP SRD attributes
* This ‘User Update Rule’ triggers when user attributes updated in Saviynt and it will create ‘Update Account’ task for SAP SRD
* User attribute Changes like :

First/Last Name

Email

Country

Department

Telephonenumber

* Execute the Provisioning job to push the update to the target
* Update Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Access provisioning

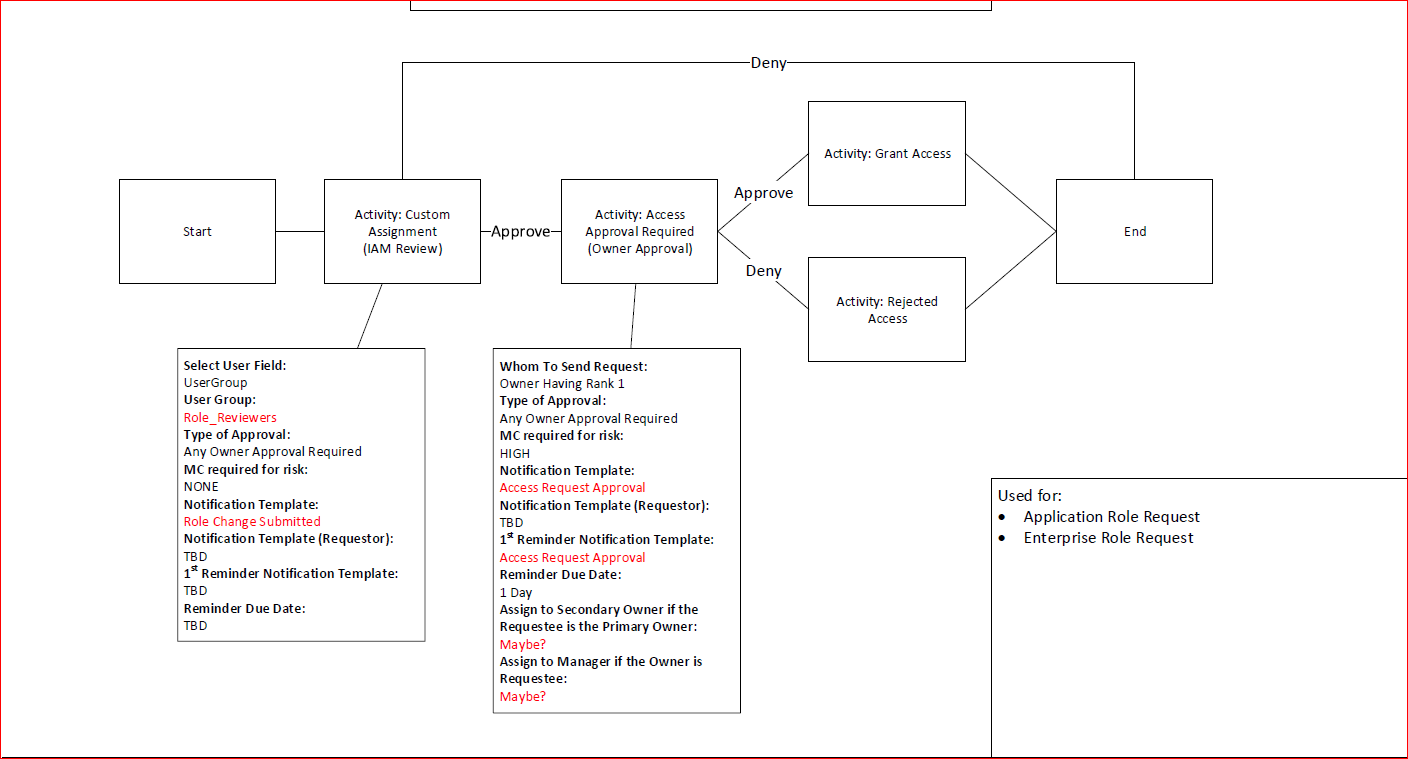
#### SAP Role management

Any changes on Role will be taken care by SSM. Below are the role types currently SSM is managing.

* Application Role
* Enterprise Role
* In SSM this can be requested under ARS tab -> Create/Modify Role
* Role Modification Approval workflow
* It has two level approval

First level – User Group (Role Reviewer)

Second level – Role Owner



#### Access Request

Under this section we are covering application/enterprise access role access request for end user

* This request can be raised under ARS tab
* This request is associated with GRC SOD evaluation
* For more details please refer section [6.8](#_SAP_GRC)

### Account De-provisioning

* Created ‘Terminate SAP Account Rule’ in Saviynt
* On user terminated this rule will execute which will create the deprovision task for account
* After provisioning job, it will lock the account in SAP
* Terminate Account JSON

|  |
| --- |
| {ENDDATE} |

### Account Re-Hire

* Created ‘Re-hire SAP Account Rule’ in Saviynt
* On user re-hire this rule will execute which will create the deprovision task for account
* After provisioning job, it will activate the account in SRD
* Re-hire Account JSON

|  |
| --- |
| {  "ADDRESS": {  "LASTNAME": "${user.lastname}",  "FIRSTNAME": "${user.firstname}",  "E\_MAIL": "${user.email}",  "DEPARTMENT": "${if(user.departmentname !=null){user.departmentname} else {''}}",  "NAMCOUNTRY": "${if(user.country !=null){user.country} else {''}}",  "COMM\_TYPE": "INT",  "TEL1\_NUMBR": "${if(user.phonenumber !=null){user.phonenumber} else {''}}",  "TEL1\_EXT": "${if(user.customproperty25 !=null){user.customproperty25} else {''}}",  "ROOM\_NO\_P": "${if(user.customproperty26 !=null){user.customproperty26} else {''}}"  },  "LOGONDATA": {  "USTYP": "A",  "GLTGB": "99991231",  "GLTGV": "${if(user.startdate!=null){new java.text.SimpleDateFormat('yyyyMMdd').format(user.startdate)}else {new java.text.SimpleDateFormat('yyyyMMdd').format(new Date())}}",  "TZONE": "EST",  "CLASS": "END\_USER"  },  "PASSWORD": {  "BAPIPWD": "${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  },  "UCLASSX": {  "UCLASS": "X"  }    } |

### Account Target recon

* Created Target recon job to import data from below SAP tables
* Job – SAPSRDIMPORT
* Tables - USR02, AGR\_TEXTS, AGR\_USERS

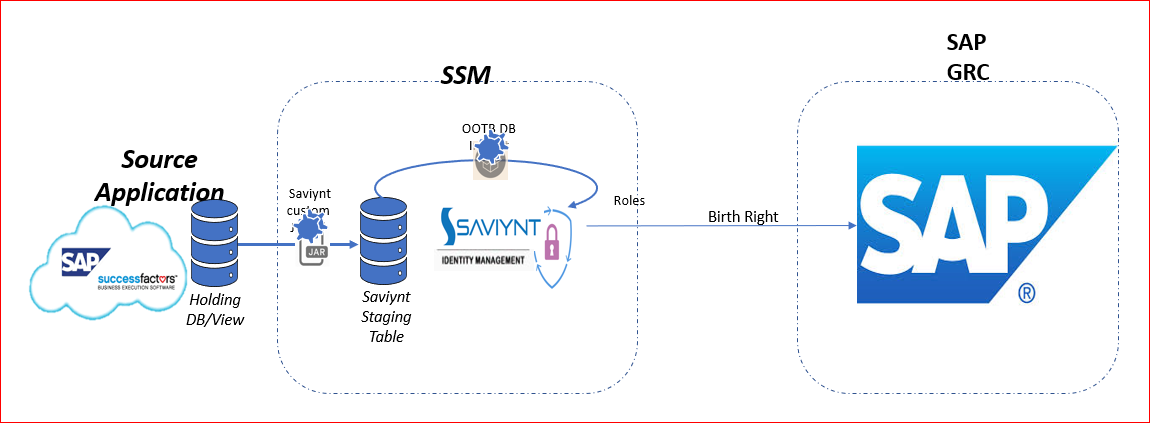
#### Correlation link

Customproperty6(SAP Account ID) AccountName

## SAP GRC

* The connection type is SAP and this connection is attached to the Security System directly.
* Automating SOD detective reporting by using SAP GRC automation

### Integration Design



### Connection Details

Following are the connection parameters configured in SSM

|  |  |
| --- | --- |
| Connection Properties | Values |
| URL | grd.mgroupnet.com |
| Username | IAM\_SAP\_D |
| Group | IDM\_RFC |
| R3 Name | GRD |
| Client Number | 100 |
| System Number | 00 |
| Message Server Port | 3620 |

### ARS Access Provisioning

GRC supports below role types for SOD evaluation. Role can be built the same way, Saviynt will support GRC simulation call outs when a user is requesting for an application/enterprise role

* Application Role
* Enterprise Role

##### **Request Submission**

* The requestor submits an access request for the endpoint associated with the Security System.
* System makes a call to GRC using the External Risk Connection. This happens through BAPI

[GRAC\_IDM\_RISK\_WOUT\_NO\_SERVICES] to GRC by passing username and list of SAP Roles to

be simulated as part of the request.

* The SOD evaluation call (External SOD) happens after the request is submitted. The requester

can then see the violation when the response is received from GRC.

* Since the Internal evaluation (Internal SOD) happens within Saviynt, there is no wait time and

the requestor will view the violations soon after the request submission.

* GRC will simulate SAP SOD violations and provide a response if any SOD violations will arise

along with severity / risk details (in parallel, Saviynt will evaluate for potential cross app /

multiple SOD rule violations)

* The configuration for retires at the endpoint level for External Risk Connection determines the

number of times the webservice call will be sent to the GRC. System will wait for a response till

all the retries are exhausted.

* If there is no response from GRC after all the retries, the request is auto-rejected and the

requester will receive an email notification about the failure.

* Till the time we get a response from GRC, the request stays in Pending state and does not go for

approval. After receiving the response from GRC, the request goes for approval. Approver is

then presented with two tabs – Internal SOD and External SOD.

* Internal SOD shows the evaluations that occur within Saviynt. This can be within the access on

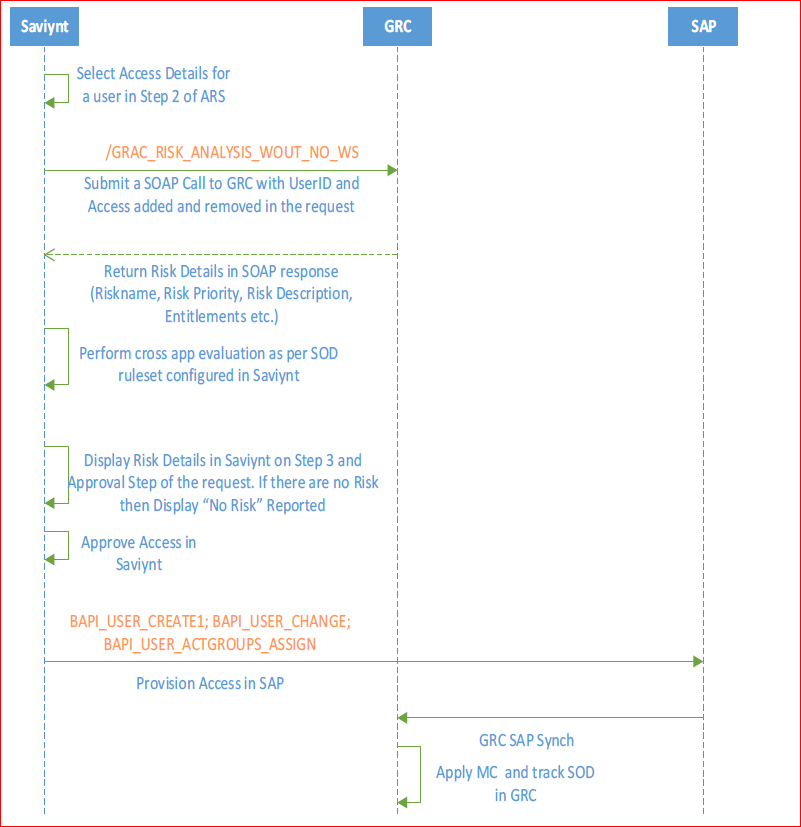
base endpoint (ECC) or it can also include cross application SODs.

* External SOD shows the evaluations that are retrieved using the External Risk SOD connection.

Logic to evaluate SOD risk is completely dependent on SAP GRC. Saviynt will only display the Risk

details returned from SAP GRC.

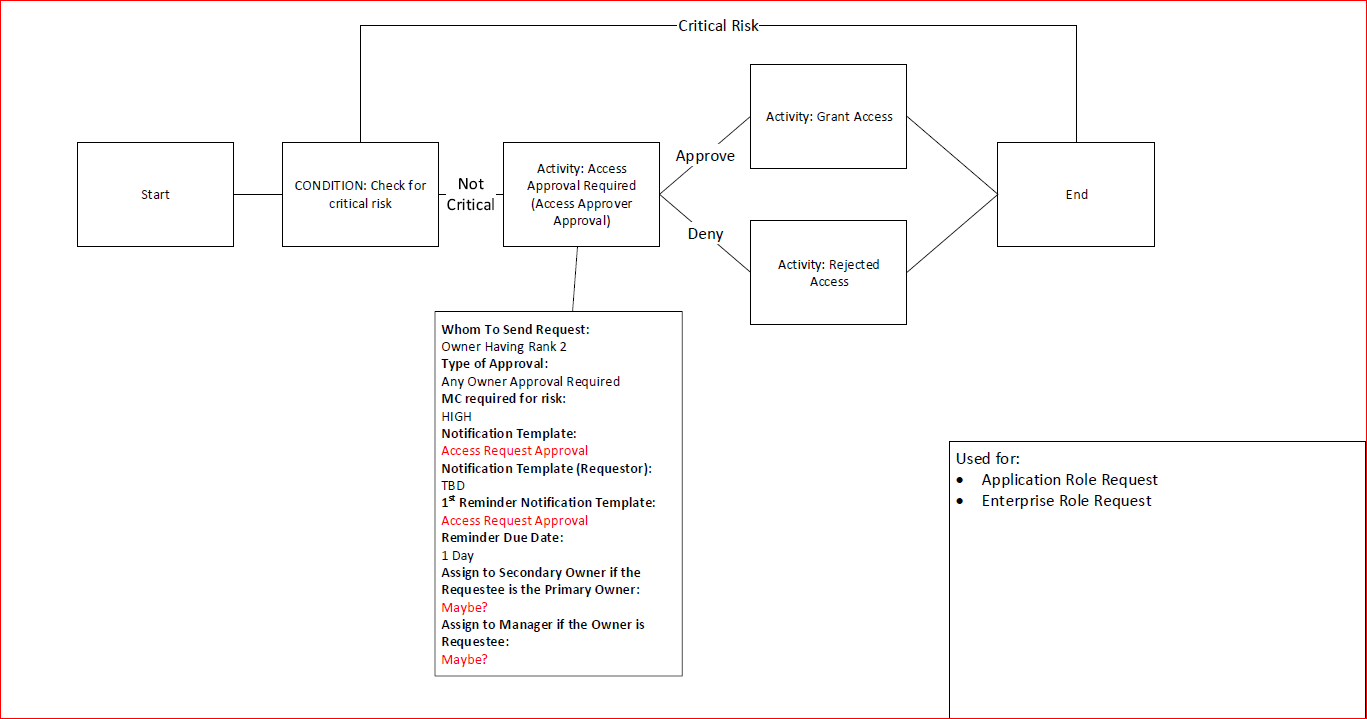
* The rest of the steps follow the existing request process as per the workflow.



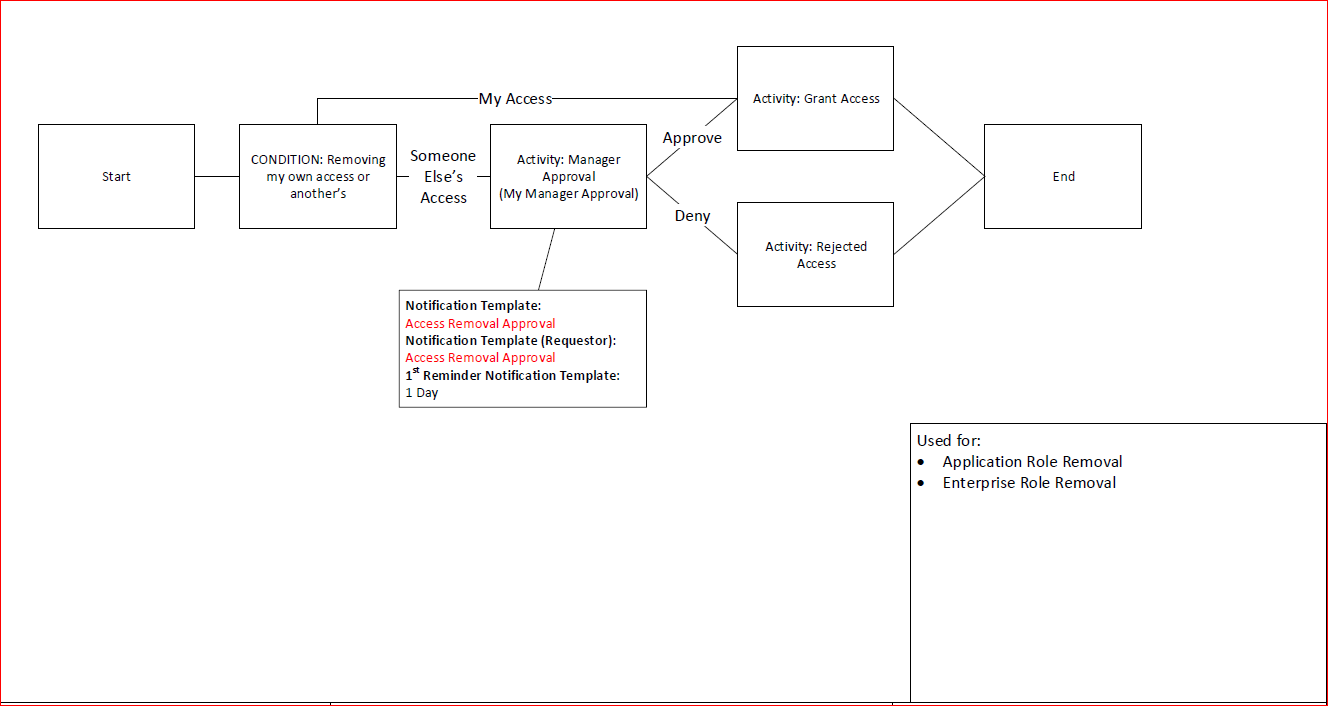
##### **Request Approval Workflow**

Below is the Workflow currently we have configured in SSM for access request addition/removal approval.

* **Add Access WF**



* **Remove Access WF**



### External Config for SOD Evaluation

Configured the below JSON for external SOD evaluation

|  |
| --- |
| {  "GRAC\_IDM\_RISK\_WOUT\_NO\_SERVICES": {  "RULE\_SET\_ID": "MARATHON",  "REPORT\_FORMAT": "2",  "HIT\_COUNT" :"100000"  }  } |

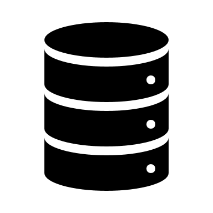
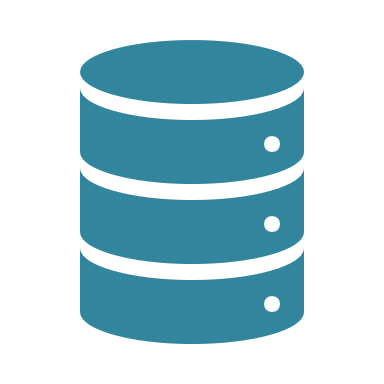
## GOLD Tables

The Gold db is the sync process which is designed to sync the users from Saviynt to Gold tables. This process is designed in such a way that if a user is active in Saviynt his details will be synced in gold db via simple DB connector in Saviynt. The application will synchronize any changes between Saviynt and the Gold DB Tables

Create user from import/UI

**Gold-DB dbo.user\_id**

Saviynt DB

Update user from import/UI

Saviynt Gold DB Connector

### Connection details

This is the connection details

|  |  |
| --- | --- |
| Connection Properties | Value |
| Connection Name | Gold-DB |
| URL | jdbc:sqlserver://common.dev.db.mgroupnet.com |
| USERNAME | IAM\_ADMIN |
| PASSWORD | \*\*\*\*\*\*\* |
| DRIVERNAME | com.microsoft.sqlserver.jdbc.SQLServerDriver |
|  | This above connection is as per QA |

### Attribute Mapping

This is the current attribute mapping from Saviynt to target

|  |  |
| --- | --- |
| Saviynt Attributes | Gold DB table |
| Username | UserId |
| CustomProperty5(Alternate login ID) | AltUserId |
| CustomProperty6(SAP Account} | SapAccount |
| CustomProperty3(MUID) | Muid |
| EMPLOYEEID | EmplNo |
| LastName | LastName |
| FirstName | FirstName |
| MidName | MidName |
| PREFERREDFIRSTNAME | NickName |
| DisplayName | DisplayName |
| CustomProperty33(Supervisor Employee ID ) | SupvEmplNo |
| Owner | SupvUserId |
| EMPLOYEETYPE | UserType |
| STATUSKEY | UserStatus |
| CustomProperty19(Chief) | IsSupervisor |
| ENDDATE | ExpirationDate |
| CustomProperty37(Contractor Company) | ContractorCompany |
| ENTITY | CoNo |
| COMPANYNAME | CoName |
| DEPARTMENTNUMBER | OrgCode |
| DEPARTMENTNAME | OrgName |
| TITLE | PositionName |
| COSTCENTER | CostCenter |
| LOCATION | SapPersArea |
| LOCATIONDESC | SapPersAreaDesc |
| EMAIL | EmailAddress |
| Calculated by EMAIL | EmailAddressRoot |
| CustomProperty26(Room No) | OfficeRoomNo |
| STREET | OfficeAddress |
| CITY | OfficeCity |
| STATE | OfficeState |
| CustomProperty29(PostalCode) | OfficePostalCode |
| PHONENUMBER | OfficePhoneNo |
| CustomProperty25(Extension) | OfficePhoneExt |
| SECONDARYPHONE | MobileNo |
| '' | FaxNo |
| '' | PagerNo |
| 'Saviynt' | UpdateBy |
| Calculated | UpdateDate |

### Rules

The below is the rules that is triggered when a user is created/updated in Saviynt

|  |  |  |
| --- | --- | --- |
| Trigger condition | Trigger Name | Details |
| Create | SqlGoldDB-create users’ rule | This rule is assigned to user whenever the status key =’1’ and this is checked as birthright rule. If the status key =’0’ the birthright rule will not be removed. |
| update | SQLGoldDB update rule Import | This rule triggers whenever there is update happens on any of the above Saviynt attributes also when the user name is not null. |
| update | SQLGoldDB update rule UI | This rule triggers whenever there is update happens on any of the above Saviynt attributes also when the user name is not null. |

### Job

This instance that is created under WSRETRY job will run for GOLD-DB.

|  |  |  |
| --- | --- | --- |
| Job | Instance Name | Frequency |
| Provisioning Job (WSRETRYJOB) | GoldDB | This job will run only for gold db tasks for create/update account operations and it runs every 2 hours. |

### Account Provisioning

Gold DB (**IAMGoldTables.dbo.user\_id**) is the target system where Saviynt talks to it by OOB Db connector to push all the account creation information’s to the target.

1. Whenever a new user is created either from UI or from import the technical rule will be triggered which will call the gold-db end point and it will trigger a create task for account creation in pending state.
2. Gold db account is provisioned to the user in the manually provisioned state as part of birth right.
3. If the birth right condition fails, the provisioned account will not be reverted back.
4. A gold db instance job runs the new account request this will connect to target and insert the record with sysdatetime.

### Account Update

1. Whenever an update operation happens either from UI or from import the two update rule will be triggered which will call the gold-db end point and it will create a update task for account update in pending state.
2. This update operations triggers when there is an update happened in above mentioned Saviynt attributes.
3. A gold db instance job runs the update account request this will connect to target and update the record based on the username.
4. On any condition the user record will not be deleted from the target, if there are inactive users found in Saviynt the status will be updated as Disabled in target too with the sysdatetime.

## Polaris

Polaris is an Oracle database, used by database users. For Saviynt this is requestable via enterprise roles a person who have Polaris application provisioned will be eligible to request the roles in Polaris. In this target system we used to create dba users and assign access to the created users, so those users will have the rights to use the database based on the groups assigned to them.

Any employee or contractor can login to Saviynt and request for Polaris.

The requesting users must have AD account assigned to them already.

The request can be made by self or for others.

Once the request is made the request will go for role owner approval, after they approved the role will be assigned to the user.

### Connection details

|  |  |
| --- | --- |
| Connection Properties | Value |
| Connection Name | DB-Polaris-test |
| URL | jdbc:oracle:thin:@Polaris.test.db.Mgroupnet.com:1521/POLARIS\_TEST |
| USERNAME | IAMT\_ORA\_DB |
| PASSWORD | \*\*\*\*\*\*\* |
| DRIVERNAME | oracle.jdbc.driver.OracleDriver |
|  | This above connection is as per QA |

### Attribute mapping

This is the current attribute mapping from Saviynt to target

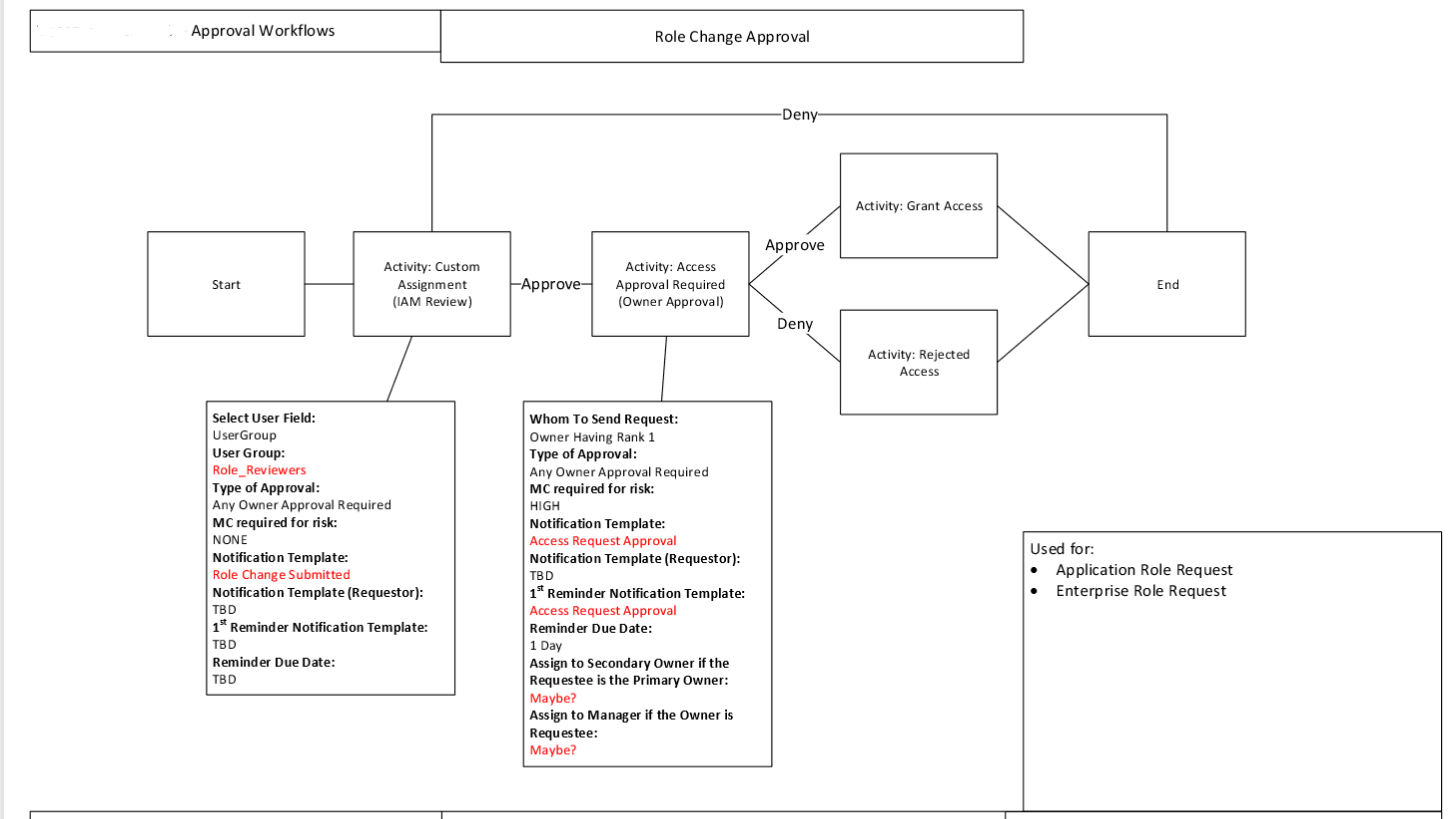
|  |  |
| --- | --- |
| Saviynt Attributes | Polaris DB |
| Username | UserId |
| password | password |

### Roles

The below are the roles from Saviynt and the roles that is mapped to the target.

|  |  |
| --- | --- |
| Saviynt Roles | Polaris DB groups |
| Polaris MPL Hydraulic and Power Engineering | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_POWER |
| POLARIS\_POWER\_U |
| POLARIS\_ALL\_READ |
| Polaris IT Application Support | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_IT |
| POLARIS\_ALL\_READ |
| Polaris MPL Application Support Administration | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ADMIN |
| SHIFT\_ADMIN |
| Polaris MPL BO and P Analysis | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_FINANCIAL |
| POLARIS\_FINANCIAL\_U |
| Polaris MPL Crude Movement Analyst | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_CRUDACCT |
| POLARIS\_CRUDACCT\_U |
| Polaris Prod Oracle |
| POLARIS\_SCHED |
| POLARIS\_SCHED\_U |
| POLARIS\_ALL\_READ |
| Polaris MPL ES and R Analyst | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_OC |
| POLARIS\_OC\_U |
| Polaris MPL Field Operations | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_REMOTE |
| POLARIS\_REMOTE\_U |
| POLARIS\_ALL\_READ |
| Polaris MPL Financial Accounting | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_FINANCIAL |
| POLARIS\_FINANCIAL\_U |
| Polaris MPL Measurement and Proving Support | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_MEASURE |
| POLARIS\_MEASURE\_U |
| Polaris MPL Operations Center | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_OC |
| POLARIS\_OC\_U |
| SHIFT\_ANALYST |
| Polaris MPL Operations Center Admin | SHIFT\_ADMIN |
| Polaris MPL Operations Center Specialist | SHIFT\_SPECIALIST |
| Polaris MPL Operations Process Support | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_OC |
| POLARIS\_OC\_U |
| Polaris MPL Product Movement Analyst | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_PRODACCT |
| POLARIS\_SCHED |
| POLARIS\_PRODACCT\_U |
| POLARIS\_SCHED\_U |
| POLARIS\_ALL\_READ |
| Polaris MPL Product and Crude Movement Scheduling | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_SCHED |
| POLARIS\_SCHED\_U |
| POLARIS\_ALL\_READ |
| Polaris MPL SCADA Analyst | FDY\_USER |
| POLARIS\_USER |
| POLARIS\_ALL\_READ |
| POLARIS\_SCADA\_READ\_WRITE |
| SHIFT\_ADMIN |
| CM - MDG Approve Customer Change Request for Polaris - MPL | MGroupNetBasic~CN=APG\_PRD\_MDG\_DISPLAY\_CUSTOMER,OU=E Portal Role PRD,OU=Delegated,DC=mgroupnet,DC=com |
| SAPUM-PRD-MDP~MD: Display customer change requests and details for Polaris - Pipeline |
| AR - Display Sensitive Z9AR tables ZTF\_POLARIS | SAPUM-PRD-ECP~Display Sensitive Z9AR tables (ZTF\_POLARIS) |

### Role request approval workflow



### Account provisioning

The Polaris is a requestable enterprise roles for the end user. Any non-employee or employee can login to Saviynt and select enterprise roles and request for Polaris access.

1. Any employee or contractor can login to Saviynt and request for Polaris.
2. The requesting users must have AD account assigned to them already.
3. The request can be made by self or for others.
4. Once the request is made the request will go for role owner approval, A notification will be triggered to the role owners regarding the pending request.
5. After approving the request , a notification will be triggered to the requester saying the approval is done.
6. Then the role is assigned for the user in Saviynt and in target the corresponding groups is assigned to the user.

|  |  |
| --- | --- |
| Json operation | query |
| CREATEACCOUNTJSON | {  "CreateAccountQry": [  "Create user ${user.username} IDENTIFIED BY ${if(user.customproperty2 !=null){'Wel\_2M$'+user.customproperty2}}"  ]  } |

### Account De-provisioning

Any non-employee or employee can login to Saviynt and navigate to accounts/roles tab and raise removal request for Polaris access.

1. Any non-employee or employee can login to Saviynt and request for Polaris.
2. The request must be made by self.
3. Once the request is made the request will go for role owner approval, A notification will be triggered to the role owners regarding the pending request.
4. After approving the request, a notification will be triggered to the requester saying the approval is done.
5. Then the role is removed for the user in Saviynt and in target the user is deleted and all his roles are dropped.

|  |  |
| --- | --- |
| Json operation | query |
| DELETEACCOUNTJSON | {  "DeleteAccountQry" : [  "Drop user ${user.username} CASCADE" ]  } |

# Role Management - Pooja

## Enterprise roles:

Role from OIM will be brought in Saviynt as ‘Enterprise Role’ Role type as part of a onetime Import activity. Since Saviynt does not have concept of Access Policies, entitlements which are part of Access Policies in OIM will be directly imported inside Enterprise Roles.

Enterprise Role import will be structured in three parts

1. Basic Role Definition and Entitlements Import
2. Role Members/User import
3. User Group Import (To replicate Role Approval roles in OIM)

Following attributes have been identified to be Imported for

1. **Role Definition/Role Structure**.

|  |  |  |
| --- | --- | --- |
| **OIM Role Attribute** | **Saviynt Role Attribute** | **Additional Details** |
| Role Name | Role Name |  |
| Display Name | Display Name |  |
| Role Description | Role Description |  |
| User Defined Tag | Custom Property1 |  |
| Risk | Risk |  |
| Category | Role Type Defaulted to ‘Enterprise Role’ |  |
| Audit Objective |  |  |
| Risk Level | Custom Property2 | 7 (High - OIM)  High 5 (Medium – OIM) – Medium  3 (Low – OIM) - Low |
| Approver Role | Custom Property3 |  |
| Approver Role Description Field | Custom Property4 |  |
| Requestable | Requestable | 1 – True  0 – False |
| No such field in OIM | Status | Defaulted to 1  1 depicts Active Role in Saviynt |
|  | Role Owner This Tab will carry the Role Owner user information (First entry from Approval Role Description) | Role Owner is currently maintained in the description field of |

1. **Role - Entitlements**

|  |  |  |
| --- | --- | --- |
| **OIM Role-Entitlement Attribute** | **Saviynt Role- Entitlement Attribute** | **Additional Details** |
| Role Name | Role Name |  |
| Entitlement Value  POC\_FIELD\_VALUE | Entitlement Value |  |
| POC\_FIELD\_NAME | Entitlement Type | **OIM POC Field Name**  'UD\_ECPSAPRL\_USERROLE'  'UD\_BWPSAPRL\_USERROLE'  'UD\_MGPSAPRL\_USERROLE'  'UD\_SRPSAPRL\_USERROLE'  'UD\_GRPSAPRL\_USERROLE'  'UD\_MDPSAPRL\_USERROLE'  are equivalent to ‘saprole’ in Saviynt  **\*Polaris and AD ent. type to be added** |
| Resource Name  OBJ\_Name | Security System | OIM    SAP  -------------------------------------------------------  SRP SAP Resource Object SAP SRD     MGP SAP Resource Object SAP MGD    GRP SAP Resource Object SAP GRD    BWP SAP Resource Object SAP BW2    MDP SAP Resource Object SAP MDD    ECP SAP Resource Object SAP EC2 |
| Resource Name  OBJ\_Name | End Point | OIM    SAP  -------------------------------------------------------  SRP SAP Resource Object  SRDCLNT100  MGP SAP Resource Object MGDCLNT100  GRP SAP Resource Object GRDCLNT100  BWP SAP Resource Object  BW2CLNT100  MDP SAP Resource Object MDDCLNT100 ECP SAP Resource Object EC2CLNT100 |

1. **Role-Users/Members**

|  |  |  |
| --- | --- | --- |
| **OIM Role-User Import** | **Saviynt Role- User Import** | **Additional Details** |
| Role Name | Role Name |  |
| Members | Users |  |

**Notes:**

* While Import Test/Obsolete/Deleted Roles does not need an import. List of roles to be reviewed with MPC.
* RA – Approval roles information will be imported in Saviynt for approval purpose, but these roles will not be imported as such.

**Role Modification**

Role Modification will be auto approved if the Requestor is same as Role Owner, if not Approval Request will go Role Owner for Approval.

**Role Access Request**

Role Request will be initiated through Saviynt ARS Module and will go to all the Approvers of the Roles – Including Role Owner (Rank1) and Other approvers (Rank2). Upon Approval Tasks will be initiated for the users for fulfillment action.

## Application roles

## We are treating all roles as Enterprise roles.

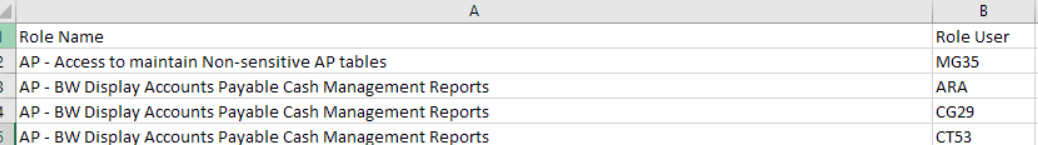
# Data Migration

Simeio has performed through analysis on SAV vs UI Import option including testing these with the Sample OIM data. After analysis and issues encountered Simeio has decided to go with UI import option to import OIM Roles data to Saviynt.

## Users

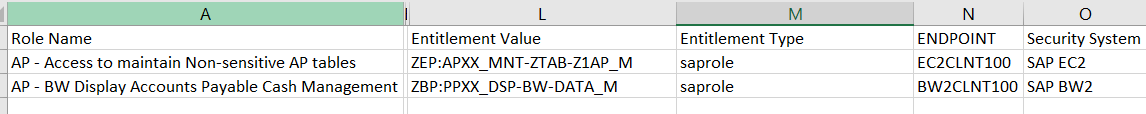
Feed file for User data will comprise of Role Name and User/Member PIK ID. Upload Role Association Option will be used from Saviynt SSM UI.

Sample Feed File Screenshot



## Access

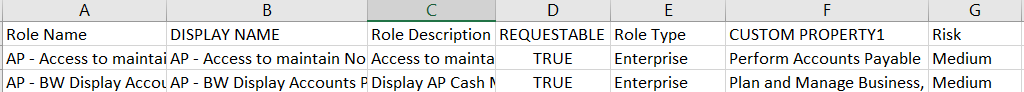
Access information – which lists the entitlements and are part of the Role will be imported along with the Role Basic Structure.

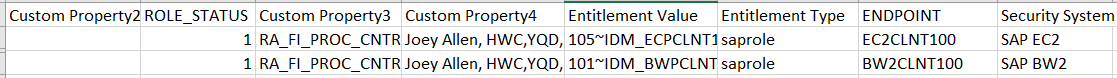


Role access can be requested via ARS module in Saviynt.

## Roles

Role Definition which includes the OIM Role Screen - Attributes and Catalog Attributes (Listed in Section 6) will be imported leveraging UI Upload Option in Saviynt





**Note:** Data Massaging will be performed on Entitlement Value before it is imported into Saviynt so that the data is compatible with Saviynt Import

Once the Import is complete – Role will be created in system; Entitlements and Users will be mapped to existing Users and Entitlements already imported in system.

## User Group Import

User Group will be used to replace the Approval Role functionality in Saviynt. User Group members will be Role Owners and other approvers.

Sample Import file:

