



**MID301: MidPoint Deployment: First steps**

# Introduction & Course Goals

What you can expect

# Course Goals

- **Understand** how “First Steps Methodology” helps you to deploy midPoint
- **Learn in iterations**, try and extend previous knowledge
- Start using midPoint by connecting your first source and target system

## Course Goals (2)

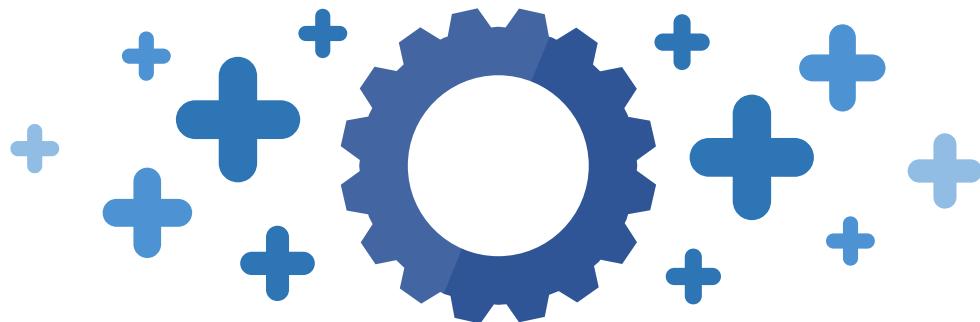
- Use simulations to allow safe configuration deployment
- Understand the concept of Resources and Connectors
- Configure resources using GUI and wizard

# Course Goals (3)

- Import data from resource
- Use Reconciliation with resources
- Clean-up data in resources (orphaned accounts etc.)
- Automate the provisioning from source system to target system through midPoint
- Prepare exceptions and data override for incorrect source system data

# What's Not Included

- No midPoint installation
- No container configuration
- No XML configuration language
- No version management
- **Will be covered in other courses**



## What's Not Included (2)

- No migration from earlier versions, starting with 4.8
- Migration from 4.4 might require additional work (e.g. resource migration)



# Course Map

## Module 1

Planning Your Deployment Project

## Module 2

Connecting Source System

## Module 3

Importing Source Data

## Module 4

Connecting Target System

## Module 5

Target System Integration

## Module 6

Importing Usernames From Target System

# Course Map (2)

## Module 7

Enabling Provisioning  
to Target System

## Module 8

Automating  
Integration

## Module 9

Overriding Incorrect  
Data

# Module 1

Planning Your Deployment Project

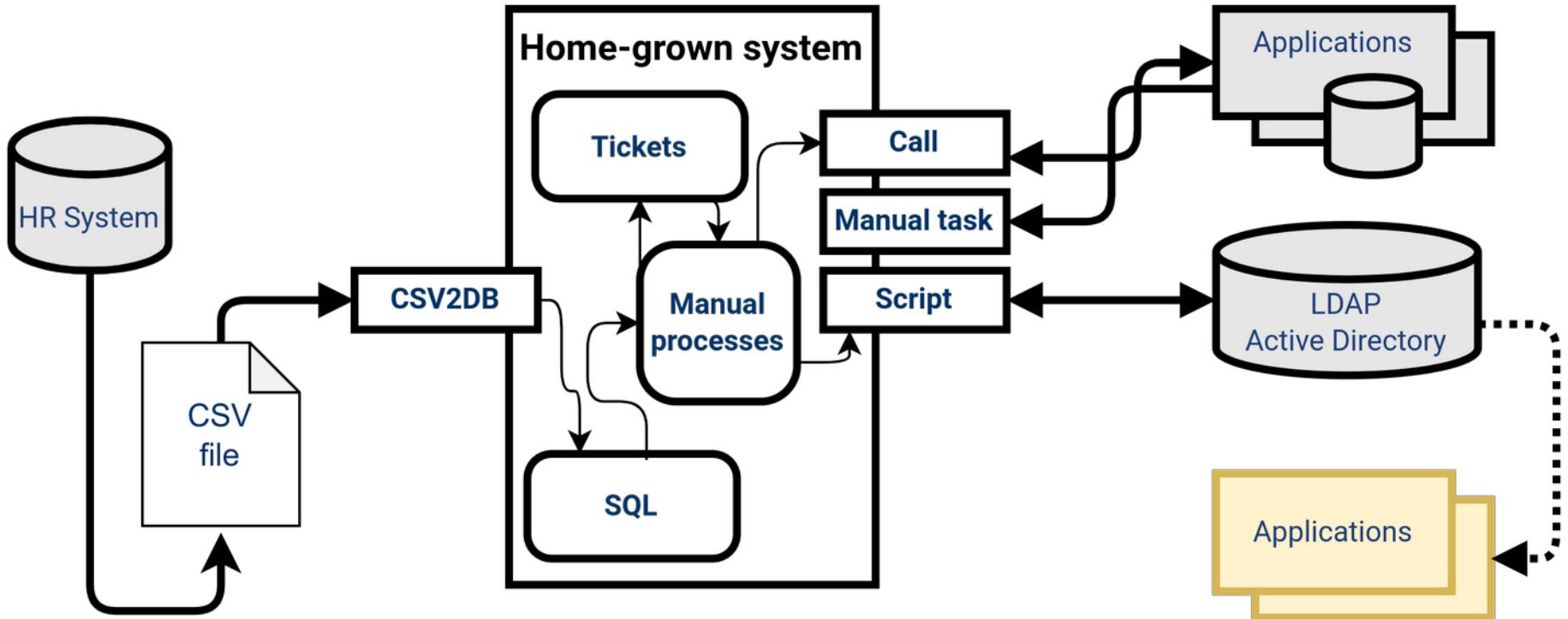
# Existing Situation

- Provisioning is currently partially implemented using a home-grown solution for some target systems
- Source system: HR application (exported CSV file); includes non-IT personnel
- Some target systems are managed by their administrators using tickets
- Usernames are created by AD administrators (**jsmith** convention, appending number of not unique, manually)
- AD username is used in all other systems

## Existing Situation (2)

- No self-service, no roles, no role request process
- AD groups are used in AD for access control
- Most target systems use AD for authentication
  - ⓘ No SSO configuration within this training
- Home-grown solution ... has grown out of control
  - “Do not touch mode”, (original author retired)

# Existing Situation: Architecture (Home-grown System)



# HR Application: Show Users

- List of HR records with ability to modify

Demo HR app   Register user **Show users**   Monday, October 9, 2023 at 5:54:26 AM PDT

## Show users

| <b>ID</b> | <b>First name</b> | <b>Surname</b> | <b>Art name</b> | <b>Emp type</b> | <b>Job</b> | <b>Employee number</b> | <b>Locality</b>     | <b>Country</b>   | <b>Status</b> | <b>Action</b>   |
|-----------|-------------------|----------------|-----------------|-----------------|------------|------------------------|---------------------|------------------|---------------|---|
| 1         | Geena             | Green          |                 | FTE             | 124#CEO    | 1001                   | Small Red Rock City | _loc:Rocky State | In            |  <b>Modify</b>   |
| 2         | Ana               | Lopez          |                 | FTE             | 125#CFO    | 1002                   | Hot Lava City       | _loc:Lava State  | In            |  <b>Modify</b> |

# HR Application: Register User

Demo HR app **Register user** Show users

- Register (new) record
- Some fields are mandatory

## Register user

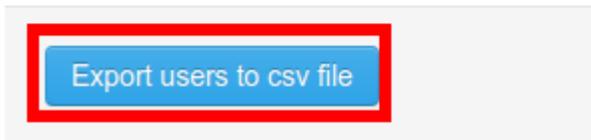
Please fill information below

|                 |                          |
|-----------------|--------------------------|
| First name      | <input type="text"/>     |
| Surname         | <input type="text"/>     |
| Artname         | <input type="text"/>     |
| Employee number | <input type="text"/>     |
| Locality        | <input type="text"/>     |
| Country         | <input type="text"/>     |
| Job             | <input type="text"/>     |
| EmpType         | <input type="text"/> FTE |
| Status          | <input type="text"/> In  |

**Register user**

# HR Application: Export to CSV File

- HR data can be exported to CSV file
- File is stored in application server, available to midPoint using docker volume



**Number of users in database: 44**

You can export data from list to .csv format

Actual export path: `/var/opt/hr/export.csv`

**Successfully exported**

# HR Application: Export to CSV File (2)

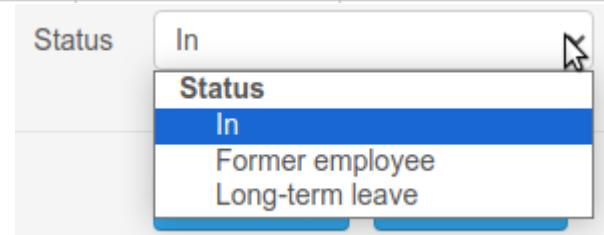
- Attributes exported: **empnum**, **firstname**, **surname**, **artname**, **emptytype**, **job**, **status**, **locality**, **country**

| empnum | firstname | surname | artname | emptytype | job                   | status          | locality            | country         |
|--------|-----------|---------|---------|-----------|-----------------------|-----------------|---------------------|-----------------|
| 1001   | Geena     | Green   |         | FTE       | 124#CEO               | In              | Small Red Rock City | Icl:Rocky State |
| 1002   | Ana       | Lopez   |         | FTE       | 125#CFO               | In              | Hot Lava City       | Icl:Lava State  |
| 1003   | Jimmy     | Taylor  |         | FTE       | 107#Junior Consultant | Former employee | Small Red Rock City | Icl:Rocky State |
| 1004   | Peter     | Hunter  |         | FTE       | 910#HR Consultant     | In              | White Stone City    | Icl:Stone State |
| 1005   | Emanuel   | Young   |         | FTE       | 120#Senior Specialist | Former employee | Hot Lava City       | Icl:Lava State  |
| 1006   | Martin    | Knight  |         | FTE       | 121#Junior Specialist | In              | Hot Lava City       | Icl:Lava State  |
| 1007   | Diane     | Davis   |         | FTE       | 107#Junior Consultant | In              | Hot Lava City       | Icl:Lava State  |
| 1008   | Elisabeth | Mason   |         | FTE       | 191#Accountant        | In              | Small Red Rock City | Icl:rocky state |

# HR Application: Data Content

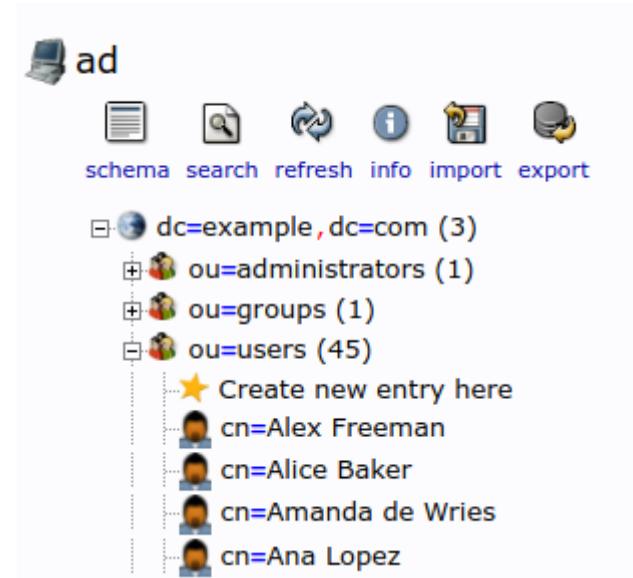
- Employees only (no contractors)
- Non-IT personnel included (should not have IT accounts)
- Attribute **status**: In / Long-term leave / Former employee (should be reflected in target system accounts statuses – only “In” having active accounts)

| empnum | firstname | surname   | artname | emptytype | job                                   | status | locality            | country             |
|--------|-----------|-----------|---------|-----------|---------------------------------------|--------|---------------------|---------------------|
| 8000   | Janet     | Garner    |         | PTE       | 899#Cleaning & Maintenance Specialist | In     | Hot Lava City       | Icl:Lava State      |
| 8001   | Ben       | Goosehead |         | PTE       | 899#Cleaning & Maintenance Specialist | In     | Hot Lava City       | Icl:Lava State      |
| 8002   | Maria     | Alvarez   |         | PTE       | 899#Cleaning & Maintenance Specialist | In     | Small Red Rock City | loc:Rocky State     |
| 8003   | Monica    | Mendez    |         | PTE       | 899#Cleaning & Maintenance Specialist | In     | Fast River City     | rlc:Two River State |



# Active Directory: Data Content

- **cn** of DN is created manually as user's Given Name + Family Name (but must be unique)
- **uid (*sAMAccountName*)** is created manually in **jsmith** convention (but must be unique)
  - ⓘ Some accounts (deliberately) don't match the convention
- ⓘ We are simulating AD with OpenLDAP



# Active Directory: Data Content “Errors”

| DN: cn=Alex Freeman,ou=users,dc=example,dc=com |                                   |
|--|-----------------------------------|
| Attribute Description                          | Value                             |
| <i>objectClass</i>                             | <i>inetOrgPerson (structural)</i> |
| <b>cn</b>                                      | Alex Freeman                      |
| <b>sn</b>                                      | Freeman                           |
| displayName                                    | Alex Freeman                      |
| employeeNumber                                 | 1010                              |
| givenName                                      | Alex                              |
| l  | Fast River City                   |
| st   | Two River State                   |
| <b>uid</b>                                     | afreeman                          |
| userPassword                                   | SSHA hashed password              |

| DN: cn=Geena Green,ou=users,dc=example,dc=com |                                   |
|---|-----------------------------------|
| Attribute Description                         | Value                             |
| <i>objectClass</i>                            | <i>inetOrgPerson (structural)</i> |
| <b>cn</b>                                     | Geena Green                       |
| <b>sn</b>                                     | Green                             |
| displayName                                   | Geena Green                       |
| employeeNumber                                | 1001                              |
| givenName                                     | Geena                             |
| l   | Small Red Rock City               |
| st  | Rocky State                       |
| <b>uid</b>                                    | geena                             |
| userPassword                                  | SSHA hashed password              |

- ⓘ We are simulating AD with OpenLDAP

# Main Goals

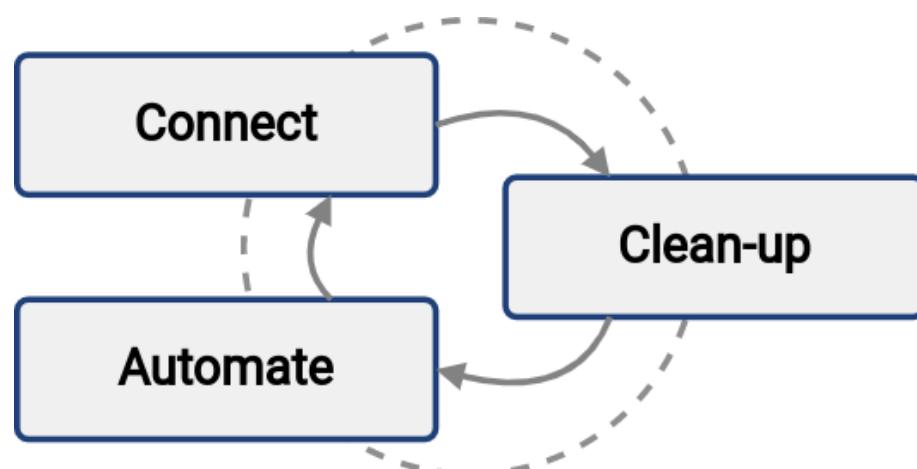
- Switch from home-grown solution for provisioning target systems using scripts to open-source provisioning and governance system (midPoint)
- Connect more target systems
- Centralize IdM and IGA

# Approach for Main Goals

- Safe migration from existing solution
- No unexpected data deletion or modification in target systems
- Smaller steps, iterations
- Use GUI whenever possible
- Utilize midPoint Simulations

# First Steps Methodology

- Simplified midPoint deployment methodology
- Quick deployment of simple midPoint configurations
- *Iterative* identity management program
- Docs: [First Steps Methodology](#)



# First Steps Methodology (2)

- **Connect**

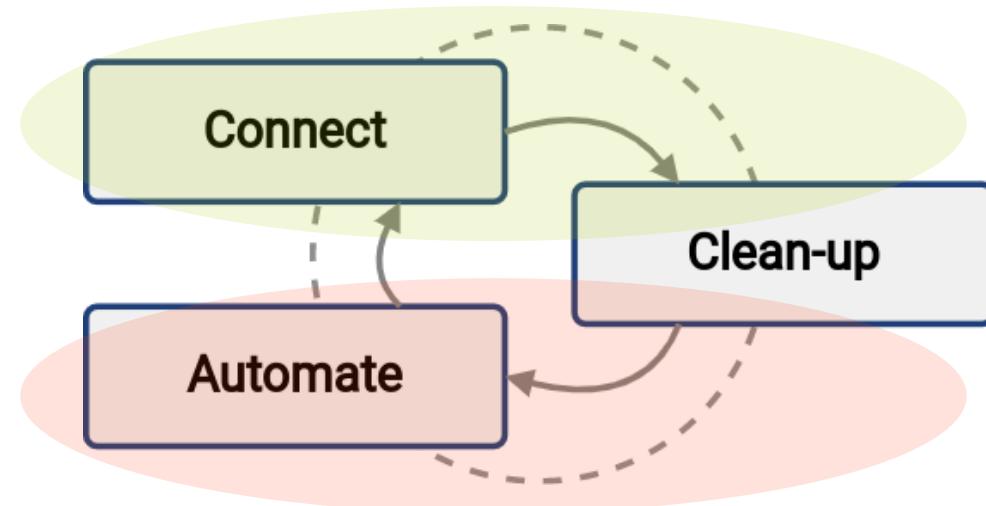
- Connect new system(s) to the solution.  
Read/analyze data

- **Clean-up**

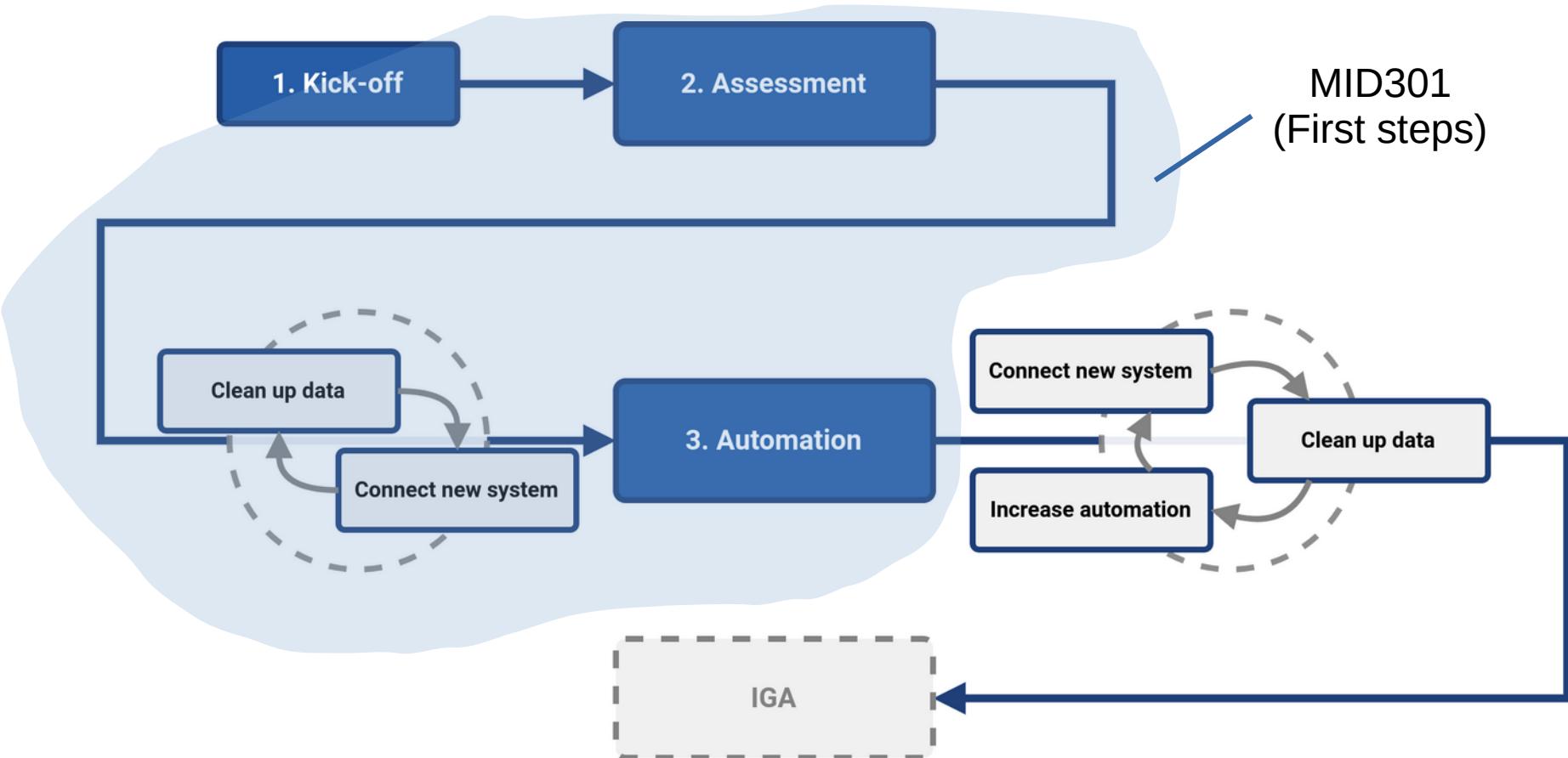
- Improve data quality. Correlate, resolve orphaned accounts, identify data errors

- **Automate**

- Speed up the processes, improve efficiency.  
On-boarding, data updates, off-boarding



# First Steps Methodology vs First Steps Training



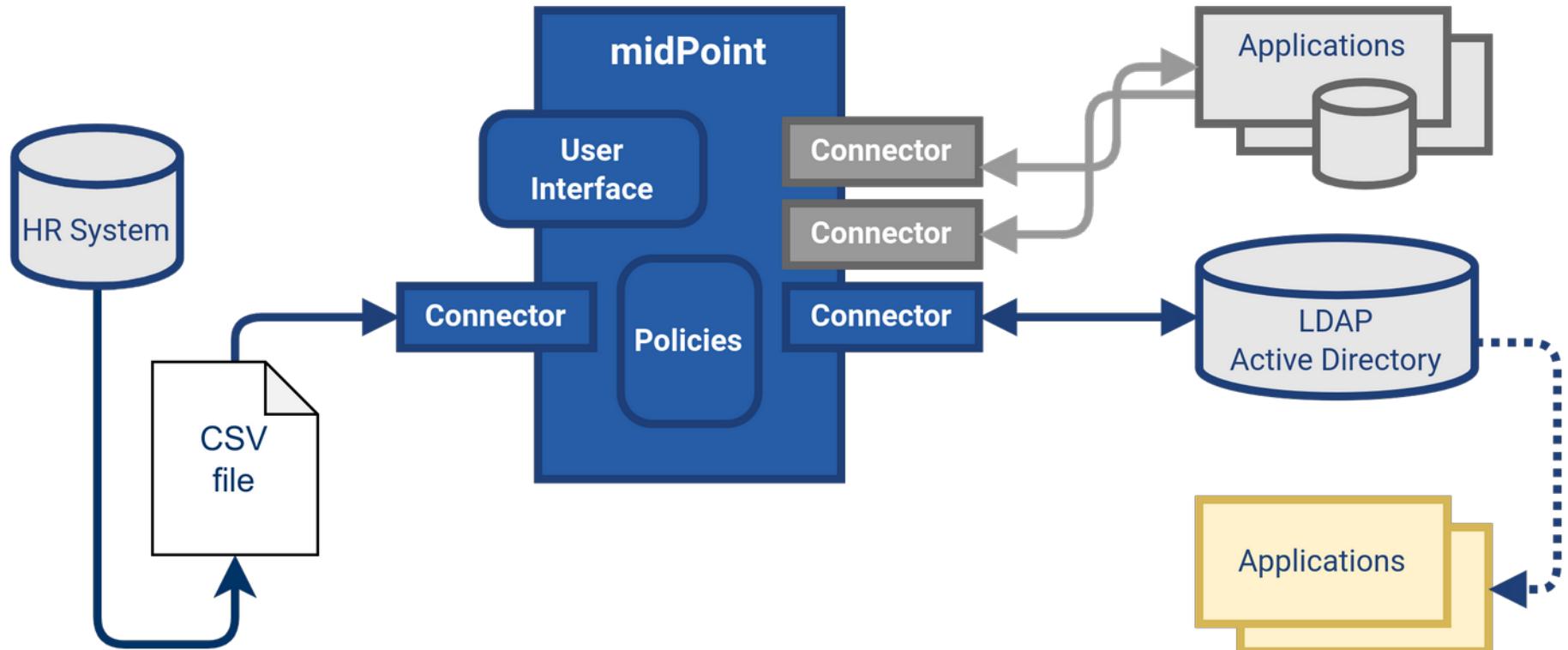
# Utilizing First Steps Methodology

| # | Step                       | Description, goals   |
|---|----------------------------|--|
| 1 | Connect Source System (HR) | We will connect the source system using CSV file and preview data                        |
| 2 | Import Source Data         | We will import data from source system, create users in midPoint                         |
| 3 | Connect Target System      | We will connect the target system (AD) using a resource template and preview data        |
| 4 | Target System Integration  | We will correlate existing accounts to midPoint users (representing HR data)             |
| 5 | Import Usernames           | We will import usernames from AD to midPoint as they are used for all other applications |

# Utilizing First Steps Methodology (2)

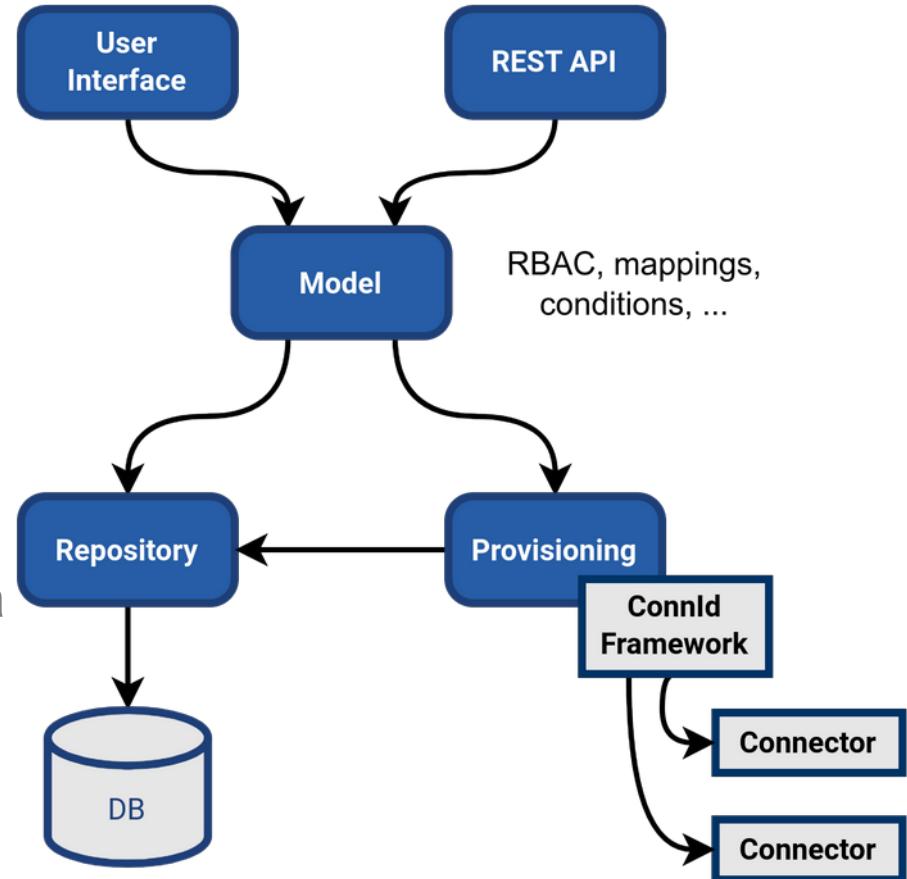
| # | Step                                 | Description, goals   |
|---|--------------------------------------|--|
| 6 | Enable Provisioning to Target System | We will prepare AD resource for provisioning from midPoint, checking what would be done using simulations  |
| 7 | Automate Integration                 | We will automate the AD account provisioning based on HR data in regular intervals.<br>We will start generating midPoint usernames on our own.<br>On-boarding, off-boarding and modifications will be automated. |
| 8 | Override Incorrect Data              | We will make sure we can override incorrect data from HR if needed   |

# Migration to midPoint: New Architecture



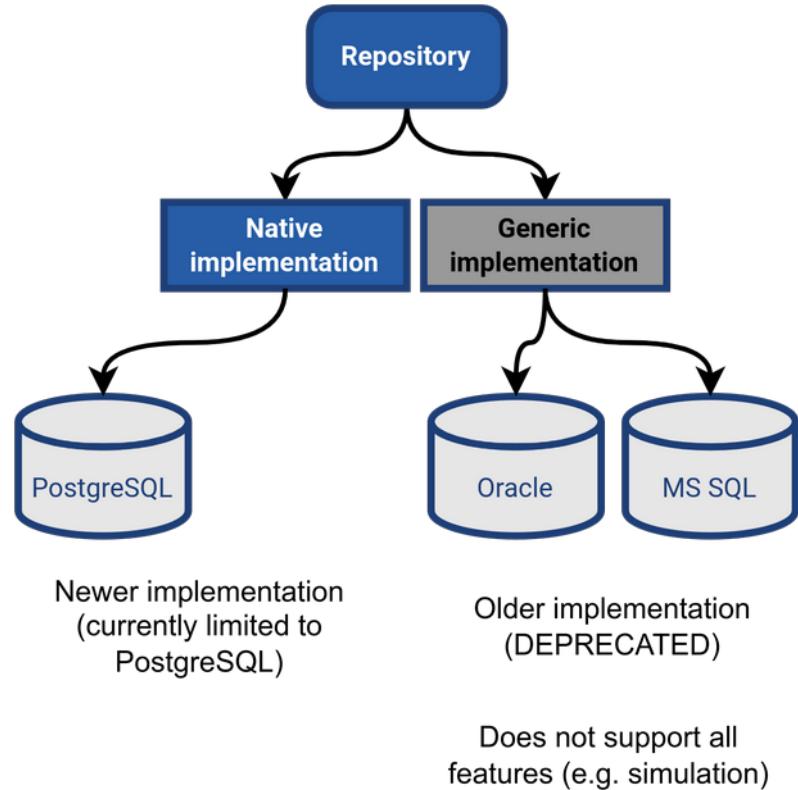
# midPoint: Basic Architecture

- Java web application
- Embedded Tomcat, runs as a standalone process
- Small number of components
- Uses XML/JSON to represent internal data
  - ⓘ We will not use this during the training



# midPoint Repository

- MidPoint needs a DB repository to store its configuration and identities
- Audit log is by default stored in the same repository
- Native repository (PostgreSQL)
- Generic repository (Oracle, MS SQL)
  - Deprecated, limited features



# Containerized Environment Introduction

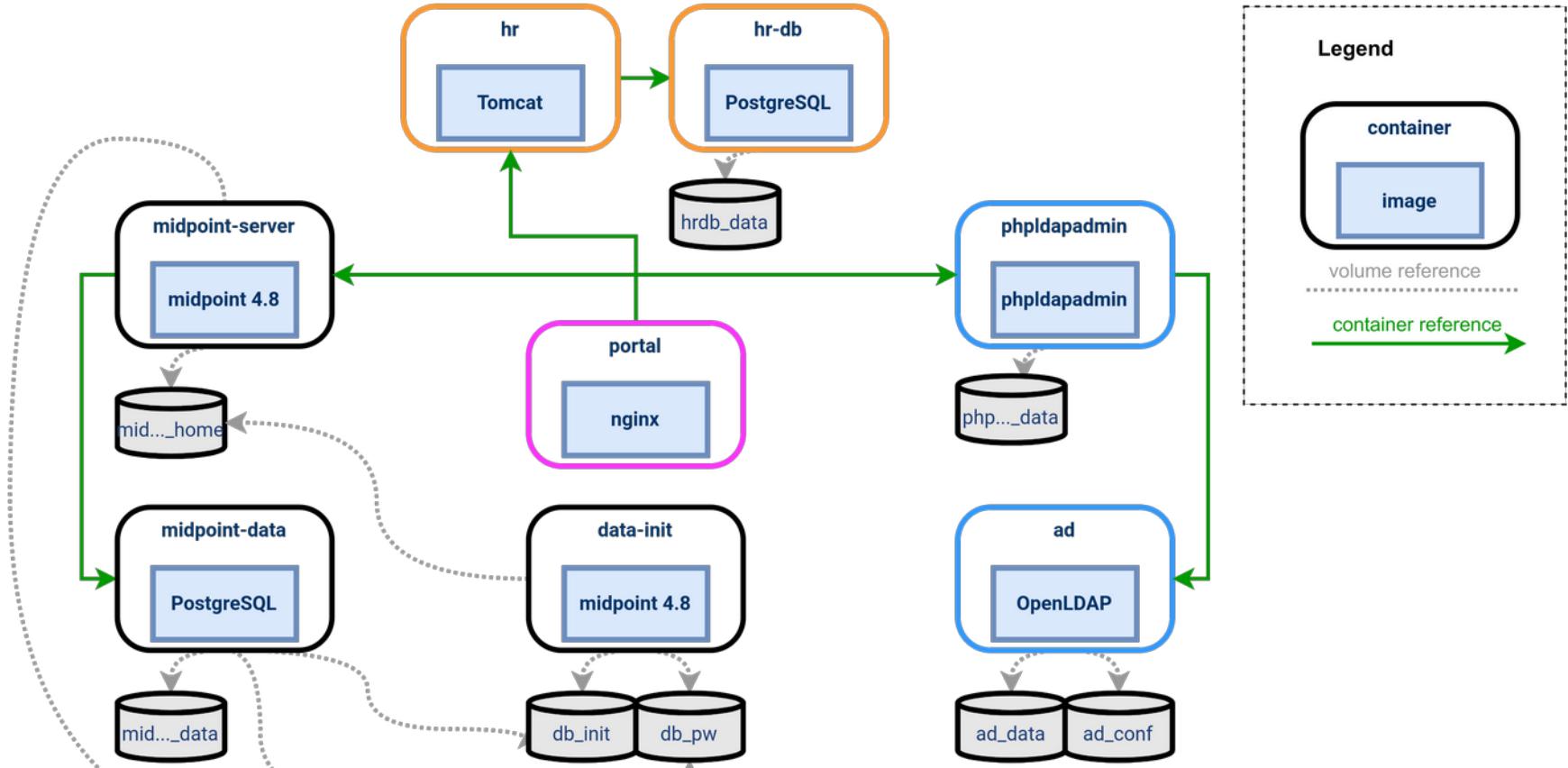
- The training is based on docker containers
  - Not just midPoint!
- Lightweight (vs virtual machine)
- Reproducible
- Isolated
- Each container serves one purpose



# Containerized Environment Introduction (2)

- Docker image: equivalent of OS/applications
  - Docker container: running instance of docker image
  - Docker volume: persistent storage accessible between containers and host
  - ⚠ We will not install the environment
    - 1) MidPoint server 4.8 LTS
    - 2) MidPoint DB repository (PostgreSQL)
    - 3) “AD” (simulated by OpenLDAP)
    - 4) LDAP browser (phpLdapAdmin)
    - 5) HR application (Tomcat)
    - 6) HR DB repository (PostgreSQL)
    - 7) Portal (Nginx)
- + 2 more data initialization containers

# Containerized Environment Architecture



# Module 1: Labs

LAB 1-1: Inspect Your Environment

# Module 1: Self-assessment

- TODO

# Module 1: Summary

- Utilizing First Steps Methodology will allow iterative and safe midPoint deployment
- This training will utilize docker containers

# Module 1

End of module

# Module 2

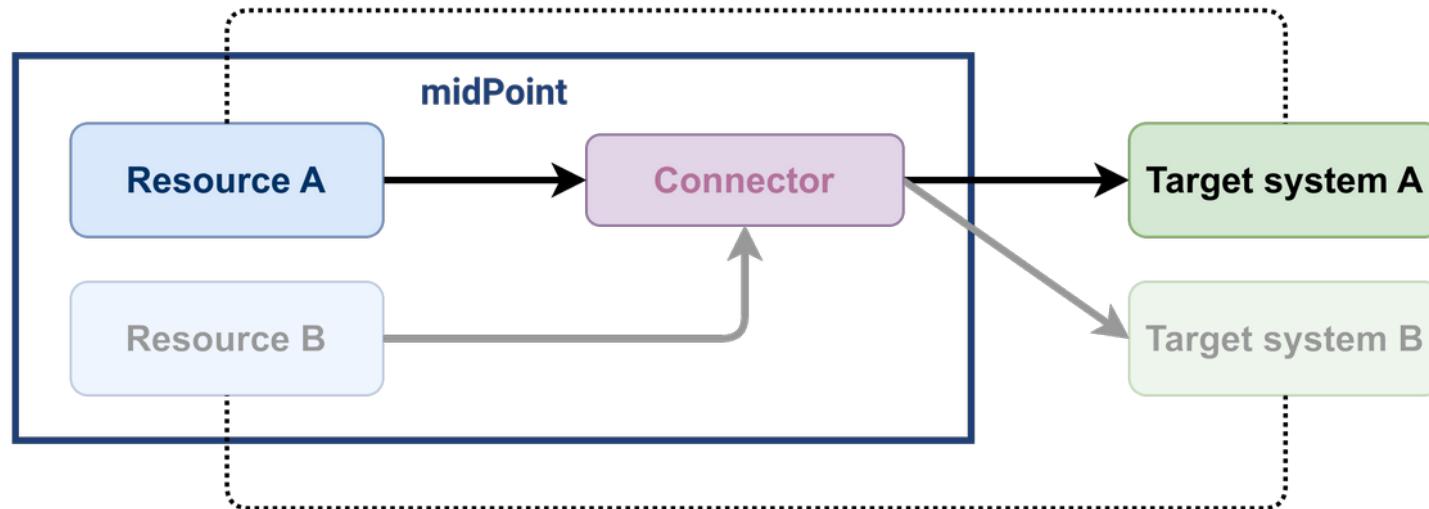
Connecting Source System

# Target Systems, Resources and Connectors

- **Target system** is an application / IS in your environment
  - HR System, Student database, ERP System, Corporate Directory, ...
- **Resource** is a midPoint object
- MidPoint connects to target systems configured as resources using **Identity Connectors**

# Target Systems, Resources and Connectors (2)

- Resource is conceptually equivalent to the source/target system
- One connector can be used for multiple source/target systems of the same type



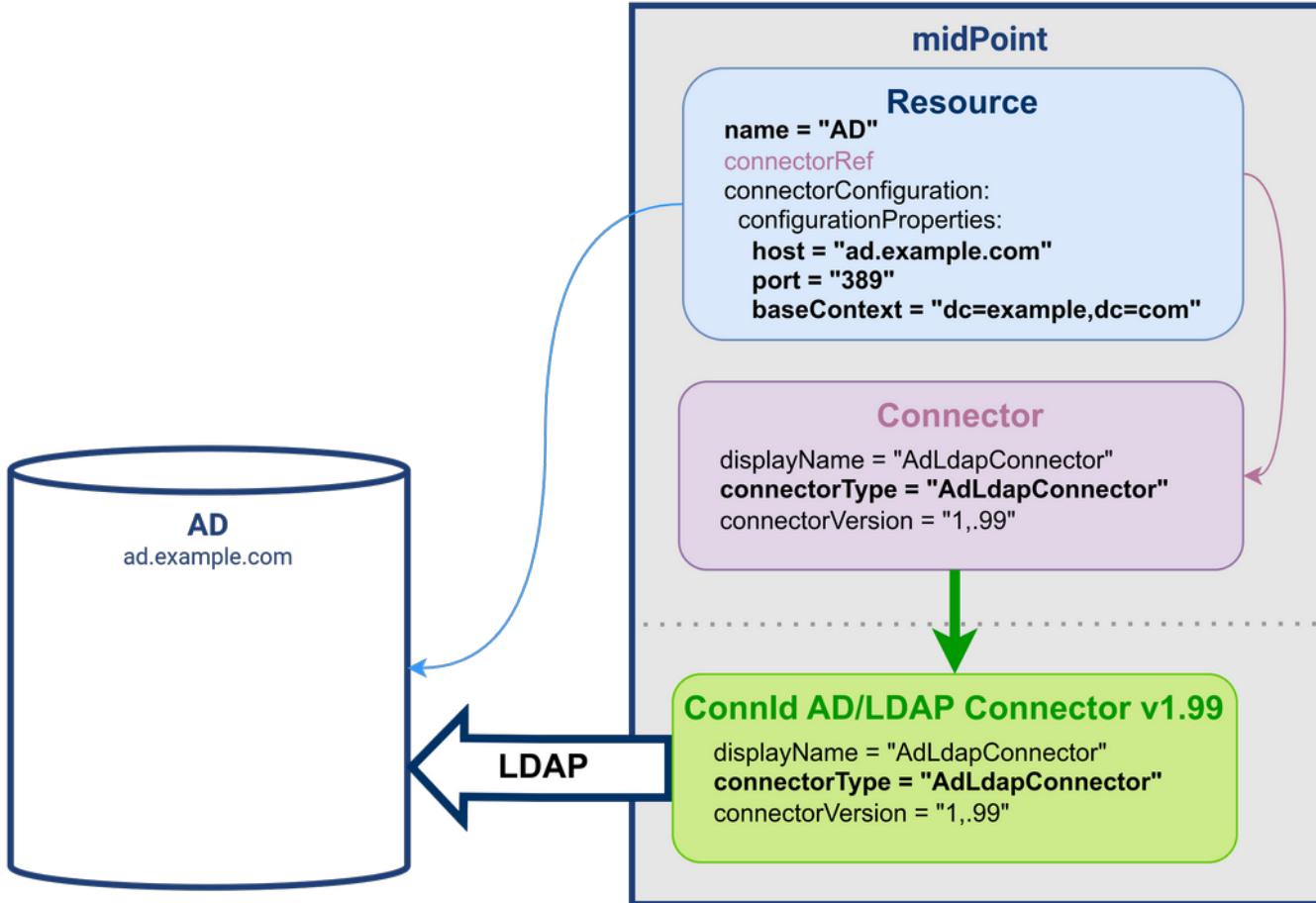
# Resource

- MidPoint representation of the source/target system
- Connection configuration including connector reference
- Resource schema (supported objects and attributes)
- Schema handling (attribute mappings, behavior)
- Capabilities
- (and more)

# | (Identity) Connectors

- Used for communication between midPoint and target system
- Similar to “Device Drivers”
- One connector can be used for several resources of the same type

# Resources and Connectors

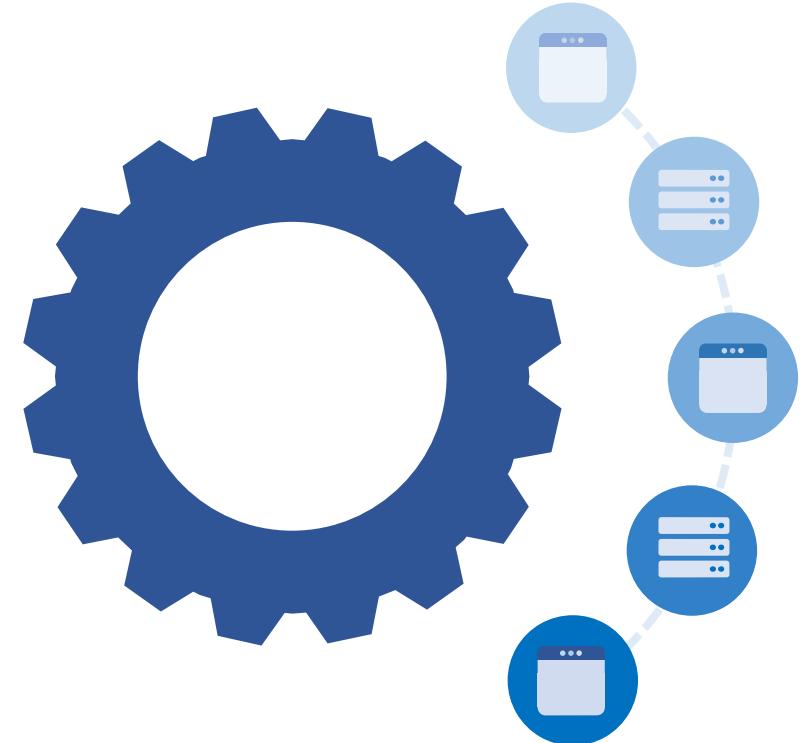


# ConnId Connectors

- ConnId Framework: <https://github.com/Tirasa/ConnId>
- Connectors are maintained independently in several projects
  - Evolveum: <https://github.com/Evolveum>
- Docs: [Identity Connectors](#)

# Built-in Connectors

- LDAP Connector
  - AD Connector
  - CSV Connector
  - DBTable Connector
- ⚠ Custom connectors may be installed anytime.



# Resource Configuration Properties

- **Connector-specific** parameters (how to connect to the target system using specified connector)
  - Host, Port, Username, Password, . . .
  - JDBC URL, Password, . . .
- Available parameters are defined by the connector developer

The screenshot shows the 'Configuration' tab of a resource configuration interface. It includes tabs for 'Configuration', 'Connector pool', 'Timeouts', and 'Results handlers'. Under the 'Configuration' tab, there are several input fields:

- Field delimiter: ,
- Quote: "
- File path \*: /opt/midpoint/var/resources/export.csv (highlighted with a red box)
- Unique attribute name: empnum (highlighted with a red box)

A 'Show empty fields' button is also present. At the bottom right, a large blue button labeled 'export.csv' is highlighted with a blue box. Below the interface, a list of CSV columns is shown in a box:

```
"empnum", "firstname", "surname", "status", "locality", . . .
```

# Resource Schema

- Objects (object classes) and their attributes supported on the resource by the selected connector
- Attribute type
- Single/multi-valued
- Mandatory/optional
- Detected by the connector

Object classes

Object class  AccountObjectClass (HR person) X

<< < 1 > >>

**Details**

Display name HR Person

Description -

Kind i account

Intent i default

Native obj. class i \_\_ACCOUNT\_\_

Default i

**Attributes**

| Name     | Display name | Native attribute name | Min/max occurs | Order | Returned by default      |
|----------|--------------|-----------------------|----------------|-------|--------------------------|
| empnum   | empnum       | empnum                | 1/1            | 100   | <input type="checkbox"/> |
| surname  | surname      | surname               | 0/1            | 120   | <input type="checkbox"/> |
| locality | locality     | locality              | 0/1            | 130   | <input type="checkbox"/> |
| status   | status       | status                | 0/1            | 140   | <input type="checkbox"/> |

# Source System (CSV File Exported from HR)

- Employee records (one row per employee)
- Active, temporarily inactive and former employees
- Data is never deleted
- Unique field: `empnum` (employee number); no username
- Note: not all HR fields will be used in this training

# Source System (CSV File Exported from HR) (2)

- Non-IT personnel included (but should be excluded from processing)

| <b>Id</b> | <b>First name</b> | <b>Surname</b> | <b>Art name</b> | <b>Emp type</b> | <b>Job</b>                            | <b>Employee number</b> | <b>Locality</b>     | <b>Country</b>       | <b>Status</b> | <b>Action</b>  |
|-----------|-------------------|----------------|-----------------|-----------------|---------------------------------------|------------------------|---------------------|----------------------|---------------|--|
| 40        | Frederick         | Smith          |                 | FTE             | 400#Environment Adaptation Specialist | 1040                   | Hot Lava City       | _lcl:Lava State      | In            |  <a href="#">Modify</a>   |
| 41        | Janet             | Garner         |                 | PTE             | 899#Cleaning & Maintenance Specialist | 8000                   | Hot Lava City       | _lcl:Lava State      | In            |  <a href="#">Modify</a>   |
| 42        | Ben               | Goosehead      |                 | PTE             | 899#Cleaning & Maintenance Specialist | 8001                   | Hot Lava City       | _lcl:Lava State      | In            |  <a href="#">Modify</a>   |
| 43        | Maria             | Alvarez        |                 | PTE             | 899#Cleaning & Maintenance Specialist | 8002                   | Small Red Rock City | _loc:Rocky State     | In            |  <a href="#">Modify</a>   |
| 44        | Monica            | Mendez         |                 | PTE             | 899#Cleaning & Maintenance Specialist | 8003                   | Fast River City     | _ric:Two River State | In            |  <a href="#">Modify</a> |

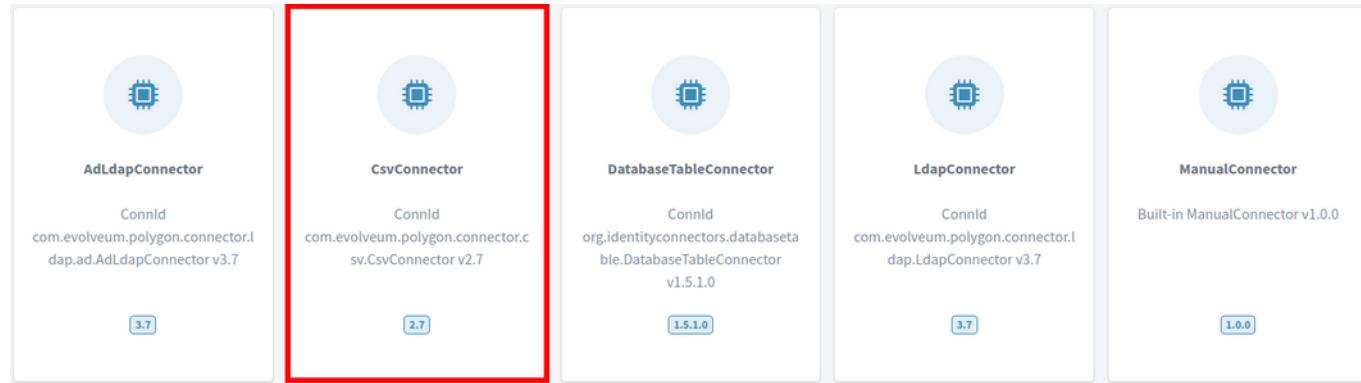
# Create Resource

- Resource wizard
- **From scratch**
- Inherit template
- Copy from template



# Creating Resource From Scratch, Step I.

- Select connector



# Creating Resource From Scratch, Step II.

- Specify basic information
  - “Name”
  - “Lifecycle state” is Proposed by default
    - (will be explained later)

**Basic information about the resource**

Fill in basic information about your resource and for more optional settings click on "Show empty fields" button

The screenshot shows a configuration interface for a resource. At the top, it says "Basic information about the resource" and "Fill in basic information about your resource and for more optional settings click on 'Show empty fields' button". Below this, there are three main input fields: "Name" with the value "HR", "Description" (an empty text area), and "Lifecycle state" with the value "Proposed". At the bottom of the form, there is a blue button labeled "Show empty fields". At the very bottom, there are two navigation buttons: "Back" and "Next: Configuration".

Basic information

Name

Description

Lifecycle state

Show empty fields

← Back    Next: Configuration →

# Creating Resource From Scratch, Step III.

- Specify connection information
  - CSV: “File path”
- MidPoint will use this information to connect to the target system and show more configuration parameters

**Establish a connection**

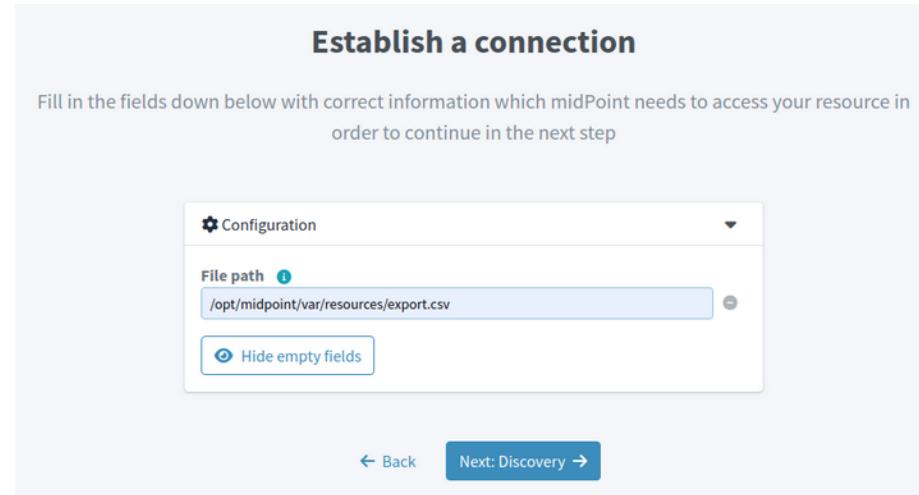
Fill in the fields down below with correct information which midPoint needs to access your resource in order to continue in the next step

**Configuration**

File path [i](#)  
`/opt/midpoint/var/resources/export.csv`

Hide empty fields

[← Back](#) [Next: Discovery →](#)



# Creating Resource From Scratch, Step IV.

- Specify additional information
  - CSV: at least “Unique attribute name”
  - CSV: Also check field delimiters
- MidPoint will use this information during communication with the system

**MidPoint discovery**

Your resource has gone through the discovery process and midPoint found following parameters for you that you can configure

The screenshot shows a configuration interface for a CSV file. It includes fields for Multivalue delimiter (set to comma), Field delimiter (set to semicolon), Quote (set to double quotes), User password attribute name (empty), Name attribute (empty), and Unique attribute name (empty). A 'Show empty fields' button is also present. Navigation buttons at the bottom include 'Back' and 'Next: Schema'.

Configuration

Multivalue delimiter   
Field delimiter   
Quote "  
User password attribute name   
Name attribute   
Unique attribute name   
  
  
[← Back](#) [Next: Schema →](#)

# Creating Resource From Scratch, Step V.

- Check schema of object types
  - CSV: no changes are needed
- MidPoint will use this information ask connector to generate schema for the selected object classes

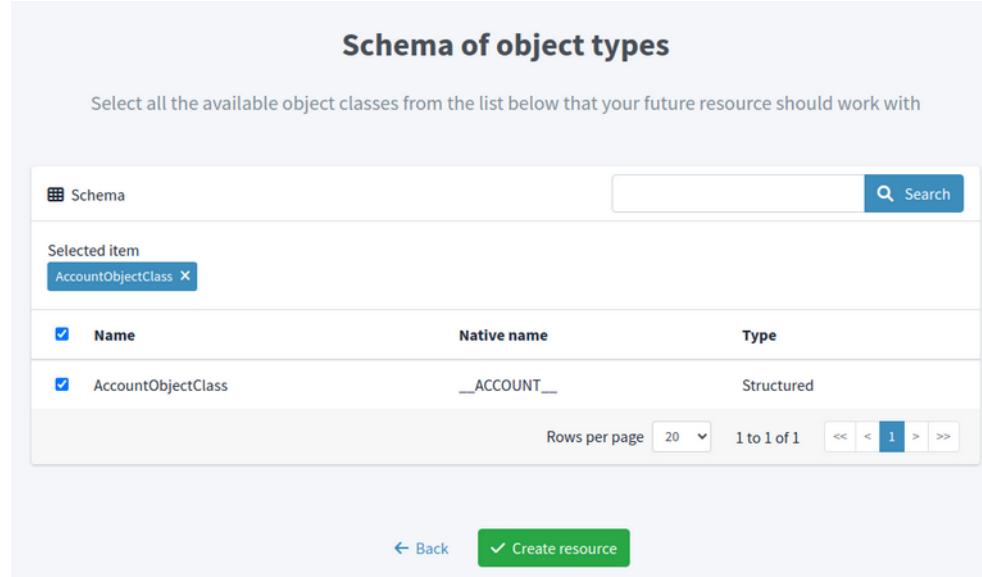
**Schema of object types**

Select all the available object classes from the list below that your future resource should work with

| Name   | Native name | Type       |
|--|-------------|------------|
| <input checked="" type="checkbox"/> AccountObjectClass | __ACCOUNT__ | Structured |

Rows per page: 20 | 1 to 1 of 1 | << < 1 > >>

Back 



# Creating Resource From Scratch, Step VI.

- Resource wizard completed creation of resource
- Further configuration is required for object types (see later)

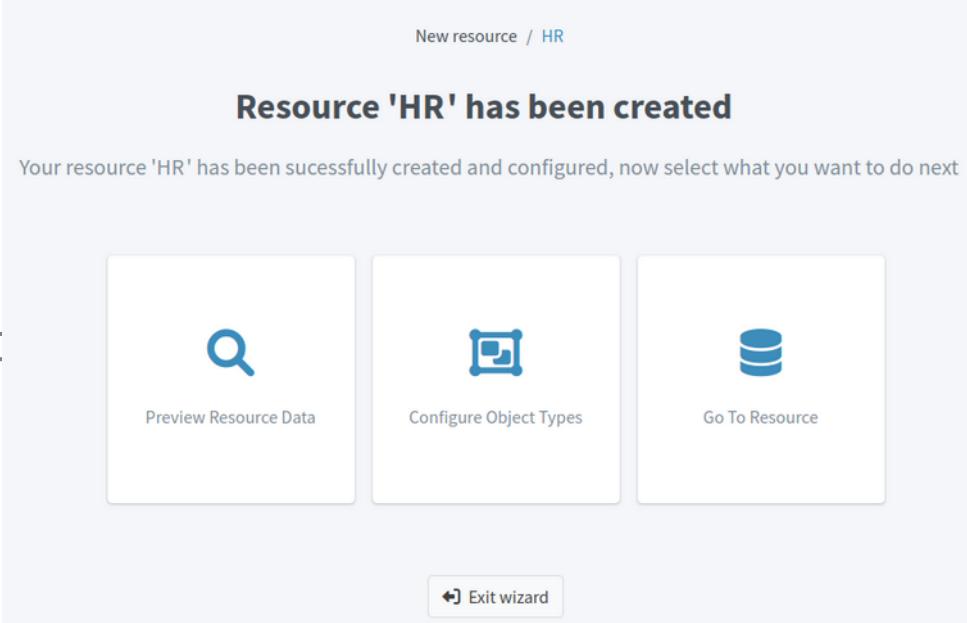
New resource / HR

**Resource 'HR' has been created**

Your resource 'HR' has been sucessfully created and configured, now select what you want to do next

[!\[\]\(3d5e02bd2d7d668db80c79c8b7906e3d\_img.jpg\) Preview Resource Data](#)[!\[\]\(b5b02390dbe8bb0f3da6850b533e20ce\_img.jpg\) Configure Object Types](#)[!\[\]\(be8e16413e41a4e243ff9fdcd9b236d5\_img.jpg\) Go To Resource](#)

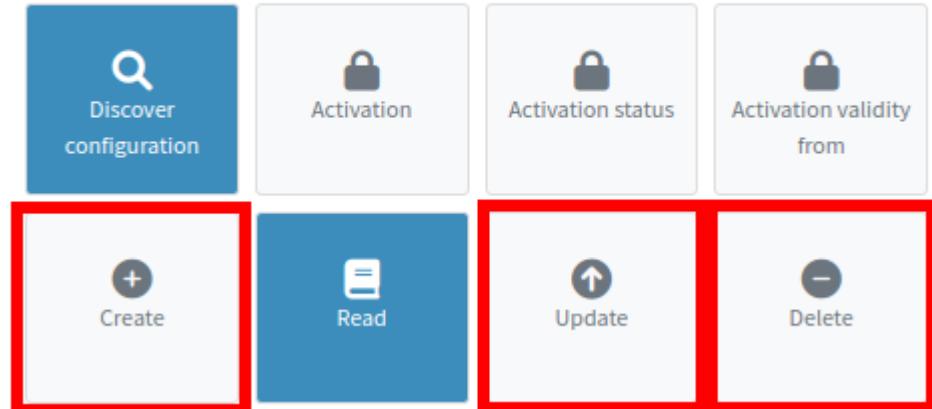
 [Exit wizard](#)



# Resource Capabilities

- Native: features of the connector
- Configured: overridden/disabled for specific resource from GUI
- Useful to make read-only resource
- Global or per-object type

## List of capabilities



# Schema Handling and Object Types

- Configuration of provisioning for **objects** on the resource
- **Kind** (account, entitlement, generic)
- **Intent**: any string, or “default”
- **Default**: true/false
- **Object class**: for connector (AccountObjectClass, inetOrgPerson, ...)
- **Display name**: used in GUI instead of the properties above for brevity purposes

| Display name | Kind    | Intent | Default | Description | Lifecycle state | - |
|--------------|---------|--------|---------|-------------|-----------------|---|
| HR person    | ACCOUNT |        | true    |             | Active          | - |

Rows per page: 20 | 1 to 1 of 1 | << < 1 > >>

# Creating Object Type, Part I.

- Define object type
- Specify “Display name”
- “Kind”: account
- “Intent”: keep empty
- “Default”: True

**Basic information about the object type**

Fill in the basic information about your object type and when you are done proceed further

Basic information

Display name: HR

Description:

Kind: Account

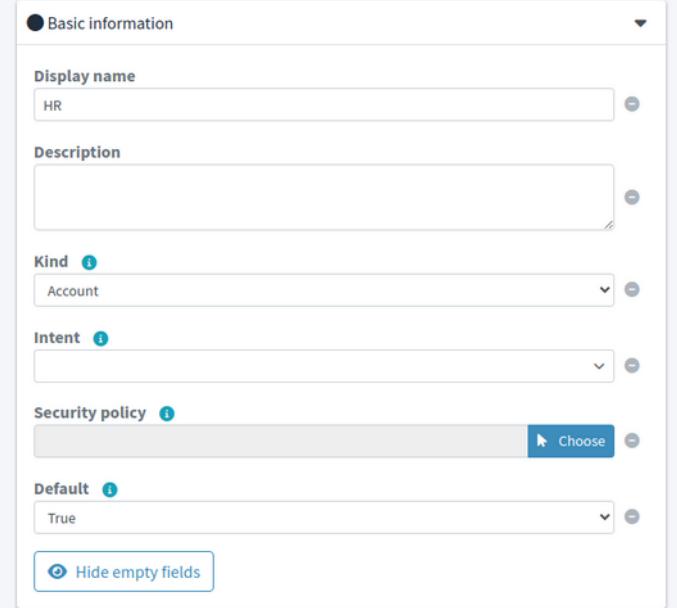
Intent:

Security policy: Choose

Default: True

Hide empty fields

[Exit wizard](#) [Next: Resource data](#)



# Creating Object Type, Part II.

- Specify the resource data
  - CSV: keep “Object class”
  - “Filter”: allows classification of data before using it in midPoint
    - Which accounts should be included
    - MidPoint query language

attributes/**empnum** not startsWith "8"

**Specify the resource data**

Fill in the necessary fields to delineate the boundary of objects that belong to that type and contains supporting instructions regarding classification of objects into types

Resource data

Object class: AccountObjectClass

Auxiliary object class: Add value, Clear all

Search hierarchy scope: Undefined

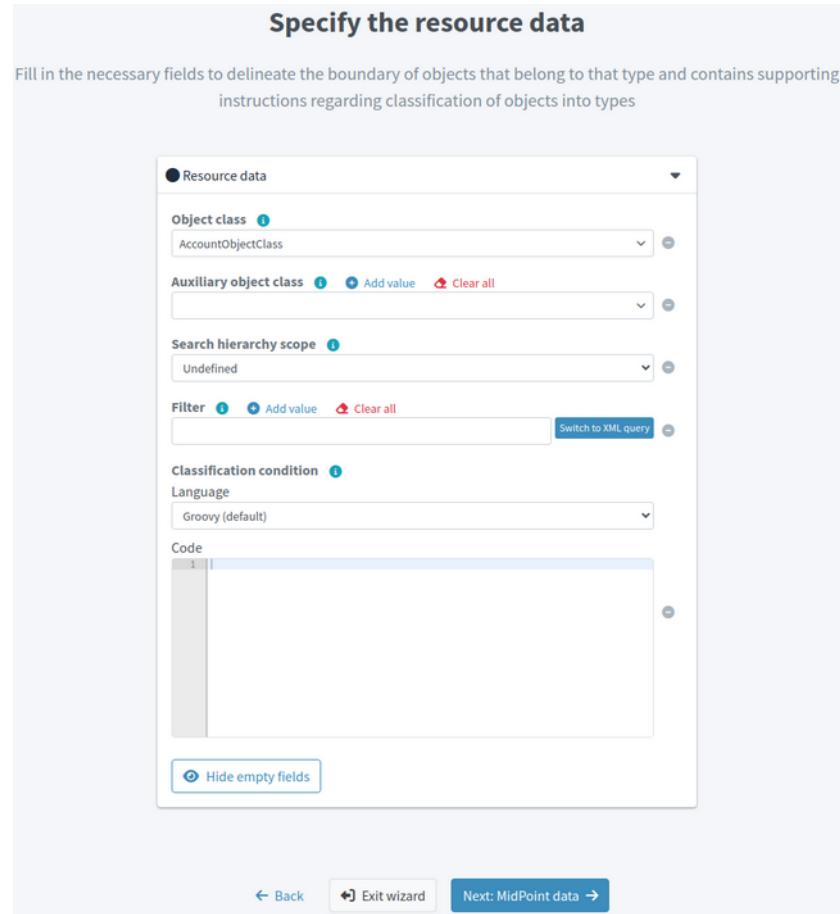
Filter: Add value, Clear all, Switch to XML query

Classification condition: Language, Groovy (default)

Code:  
1

Hide empty fields

Back, Exit wizard, Next: MidPoint data



# Creating Object Type, Part III.

- Specify midPoint data: for which midPoint objects is this object type used
  - “Type”: select User
  - “Archetype”: (optional) select archetype

**Archetype** is “object category”. Archetyped objects are displayed differently in GUI and (optionally) can include provisioning instructions (*birthright*).  
– simplified Archetype definition for First Steps training

**Specify the midPoint data**

Fill in the necessary fields to specify focus objects corresponding to given resource object type

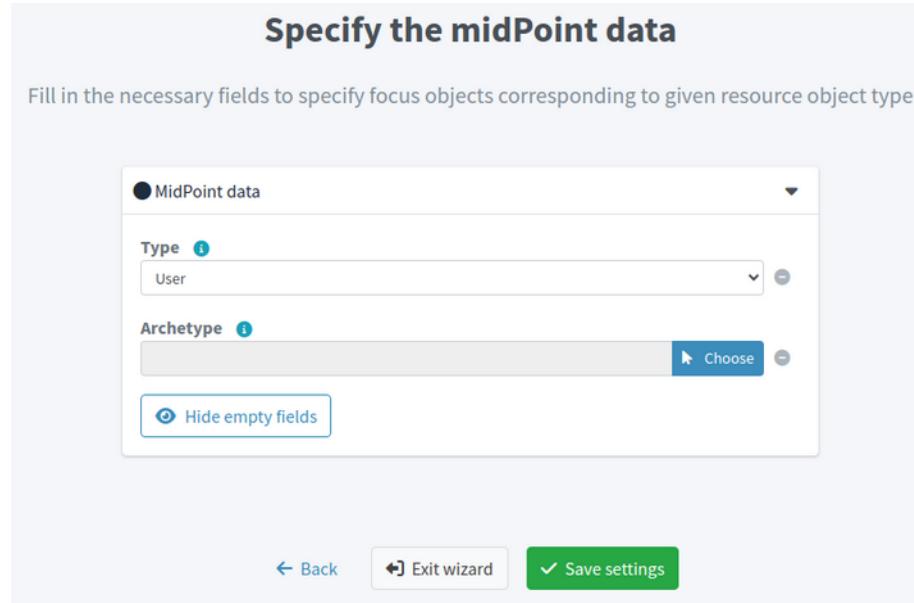
MidPoint data

Type: User

Archetype: Choose

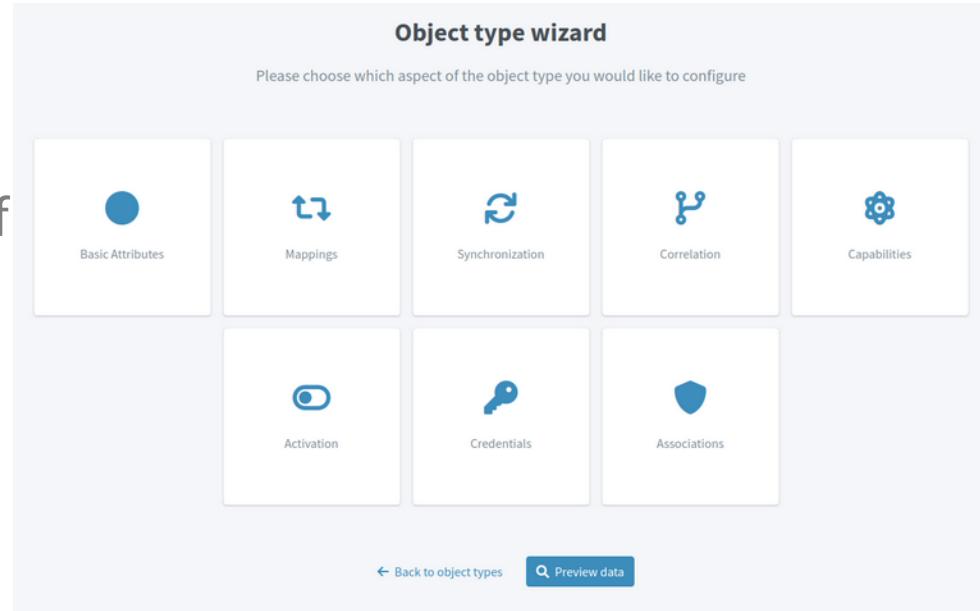
Hide empty fields

Back Exit wizard Save settings



# Creating Object Type, Part IV.

- Object type wizard completed creation of object type
- Further configuration is required



# Resource: Accounts vs Resource Objects

- Accounts: view of already classified accounts (**belong to Object type**)

The screenshot shows the 'Accounts' page in a software application. On the left, a sidebar menu lists several options: Details, Basic, Connector configuration, Defined Tasks, Accounts (which is selected and highlighted with a red box), Entitlements, Generics, Resource objects, Schema handling, Connector statistics, and Schema. At the top right, there are buttons for 'HR Person' (set to 'Active'), 'Configure', 'Tasks', and 'Show statistics'. Below these are search and filter fields for 'Name', 'Situation' (set to 'Undefined'), and a 'Basic' dropdown. The main area displays a table of accounts with columns: Name, Identifiers, Situation, Owner, and Pending operations. Each account row includes a checkbox, a person icon, and the identifier 'empnum: [value]'. To the right of each row are download and export icons. The accounts listed are: 1002 (empnum: 1002), 1003 (empnum: 1003), 1004 (empnum: 1004), 1006 (empnum: 1006), and 1007 (empnum: 1007).

|                          | Name | Identifiers  | Situation | Owner | Pending operations |
|--------------------------|------|--------------|-----------|-------|--------------------|
| <input type="checkbox"/> | 1002 | empnum: 1002 |           |       |                    |
| <input type="checkbox"/> | 1003 | empnum: 1003 |           |       |                    |
| <input type="checkbox"/> | 1004 | empnum: 1004 |           |       |                    |
| <input type="checkbox"/> | 1006 | empnum: 1006 |           |       |                    |
| <input type="checkbox"/> | 1007 | empnum: 1007 |           |       |                    |

# Resource: Accounts vs Resource Objects (2)

- Resource objects: view “as on resource” (disregarding Object type)

Resource objects

|                          | Name | Identifiers  | Situation | Owner | Pending operations                            |
|--------------------------|------|--------------|-----------|-------|---|
| <input type="checkbox"/> | 1008 | empnum: 1008 |           |       | <input type="button"/> <input type="button"/> |
| <input type="checkbox"/> | 8000 | empnum: 8000 |           |       | <input type="button"/> <input type="button"/> |
| <input type="checkbox"/> | 8001 | empnum: 8001 |           |       | <input type="button"/> <input type="button"/> |
| <input type="checkbox"/> | 8002 | empnum: 8002 |           |       | <input type="button"/> <input type="button"/> |
| <input type="checkbox"/> | 8003 | empnum: 8003 |           |       | <input type="button"/> <input type="button"/> |

Rows per page  Displaying 1 to 10, unknown number of matching results. << < 1 2 3 4 5 > >>

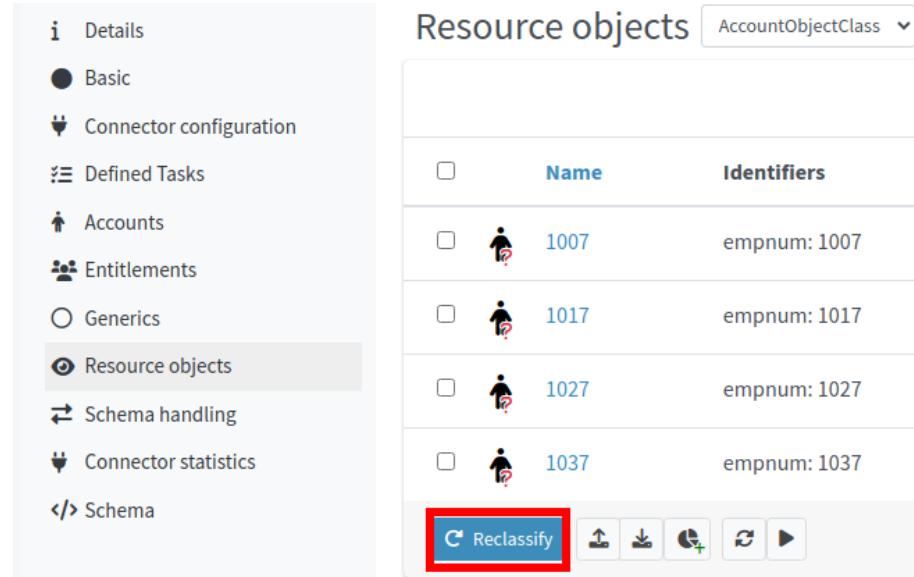
# Resource: Accounts vs Resource Objects (3)

- You can **reclassify** your accounts after you set/change the classification filter any number of times while resource is in *Proposed* lifecycle state
- Reclassification task will be created and deleted after its completion automatically

Resource objects AccountObjectClass ▾

| <input type="checkbox"/> | Name | Identifiers  |
|--------------------------|------|--------------|
| <input type="checkbox"/> | 1007 | empnum: 1007 |
| <input type="checkbox"/> | 1017 | empnum: 1017 |
| <input type="checkbox"/> | 1027 | empnum: 1027 |
| <input type="checkbox"/> | 1037 | empnum: 1037 |

**Reclassify** 



# Object Type: Synchronization

- Synchronization settings specify how midPoint will behave for objects on resource
- **Situation:** resource object state in relation to midPoint
- **Action:** automatic action to execute (optional)
- **Lifecycle state:** controls if/when the synchronization action is applied
- “Name” is optional, but recommended

| List of reactions                   |           |             |                 |   |
|-------------------------------------|-----------|-------------|-----------------|---|
| Name                                | Situation | Action      | Lifecycle state |   |
| unmatched-addFocus                  | Unmatched | Add focus   | Active          |   |
| linked-synchronize                  | Linked    | Synchronize | Active          |   |
| <a href="#">Add simple reaction</a> |           |             |                 | Rows per page: 20   1 to 2 of 2   << < > >>   |

# Synchronization Situations and Reactions

- You need to select appropriate actions for specific situations
  - Unmatched
  - Linked
  - (more situations and reactions later)
- ⚠ There is no default configuration

# Situation: Unmatched

- The resource account owner could not be determined
- Recommended actions:
  - Add focus (for authoritative source systems)
  - Delete resource object (for non-authoritative systems)
  - Inactivate resource object (for non-authoritative systems)
  - (no action)

ⓘ **Focus** means  
User in this  
context

# Situation: Linked

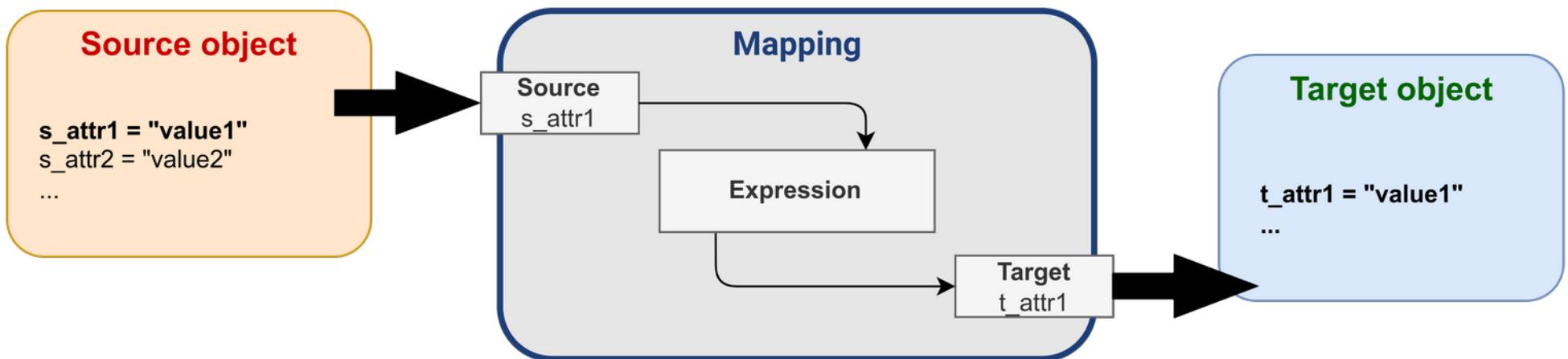
- The resource account is already linked to midPoint user
- Recommended actions:
  - Synchronize (data will synchronize using mappings)

# Object Type: Mappings

- Mapping is a very flexible mechanism used at several places in midPoint
- Transforms one or more properties to a target property
- Evaluates whenever source value(s) change or during reconciliation

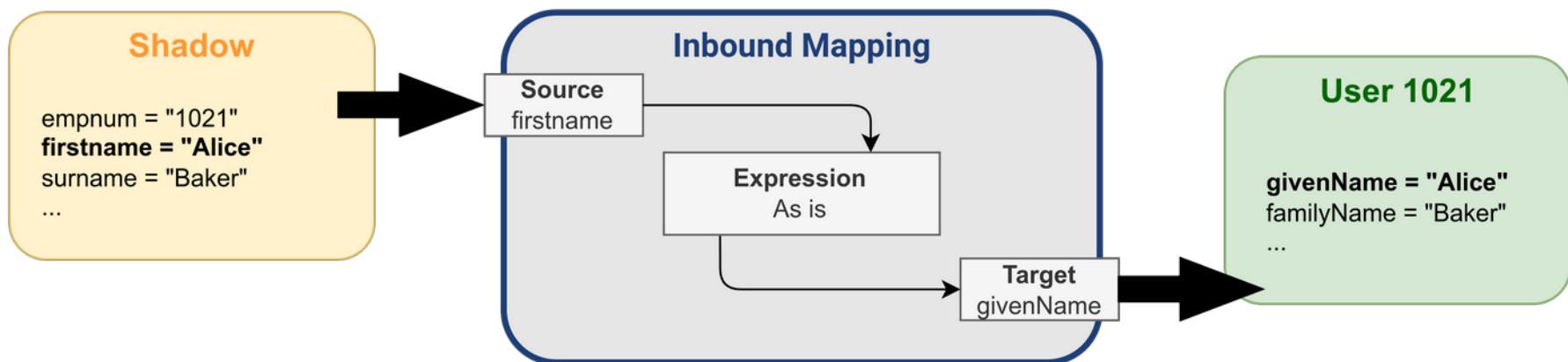
# Object Type: Mappings (2)

- Source: defines the data sources of the mapping (“input variables”)
- Expression: defines how to transform the source data or generate value
- Target: defines where the computed value should go



# Object Type: Inbound Mappings

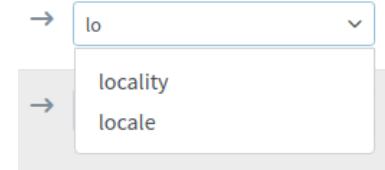
- Data flow direction: from resource to midPoint
- How source **resource attribute** value is used



# Object Type: Inbound Mappings (2)

- Click to attribute field/start typing (auto-completion of attribute/property name)
- Note:** “name” property is mandatory for midPoint users (create a mapping for it)

The screenshot shows the 'Inbound mappings (to MidPoint)' configuration screen. It has two main sections: 'All mappings' on the left and 'Expression' and 'Target' on the right. The 'Expression' section contains several rows of mappings, each with a 'From resource attribute' dropdown (containing values like empnum, firstname, surname, locality, status) and an 'Expression' dropdown. The 'Target' section lists corresponding attributes: name, personalNumber, givenName, familyName, locality, and lifecycleState. Red boxes highlight the 'Expression' dropdowns for 'empnum', 'firstname', 'surname', and 'locality', and the 'Target' dropdown for 'lifecycleState'. A tooltip 'Evaluator value is set' is visible near the 'lifecycleState' target. The top right corner shows a link to 'Outbound mappings (to Resource)'. At the bottom, there are buttons for 'Add inbound' and pagination controls.



# Expressions in Mappings

- Part of the configuration that contains logic
- Expression evaluators supported by resource wizard
  - As is: copies source value
  - Script: executes script (Groovy)

Language

Groovy (default)

Code

```
switch (input) {  
    case 'In':  
        'active'  
        break  
  
    case 'Long-term leave':  
        'suspended'  
        break  
  
    case 'Former employee':  
        'archived'  
        break  
  
    //default:  
    //  //''suspended'  
    //  //break  
}
```

Done

If HR "status" is "In", return "active".  
If HR "status" is "Long-term leave", return "suspended".  
If HR "status" is "Former employee", return "archived".

## Simple value conversion

# Object Type: Resource Wizard Navigation Note

- Two ways to get to object type configuration

The screenshot illustrates two distinct paths to reach the "Object type wizard".

**Path 1 (Left Side):** This path starts from the "Schema handling" section of the left sidebar (marked with red circle ①). It leads to a sub-menu where the "Display name" field is set to "HR Person" (marked with red circle ②).

**Path 2 (Right Side):** This path starts from the "Accounts" page. In the top right corner, there is a "Configure" button (marked with red circle ②). Clicking this button opens a dropdown menu containing several configuration options. The "Schema handling" option is highlighted with a red circle ③.

The "Object type wizard" interface itself is shown on the right, featuring a grid of icons for different configuration aspects: Basic Attributes, Mappings, Synchronization, Correlation, Capabilities, Activation, Credentials, and Associations.

This screenshot shows the "Accounts" page with a list of accounts. At the top right, there is a "Configure" button (marked with red circle ②) which, when clicked, reveals a dropdown menu (marked with red circle ③) containing various configuration options: Basic attributes, Synchronization, Mappings, Correlation, Capabilities, Credentials, Activation, and Associations.

# TODO Makes even sense? Methodology: Connecting Source System, Part I.

- TODO steps needed to connect, basically what will be done in the labs, in generic way.

## Module 2: Labs

LAB 2-1: Create HR Resource

## Module 2: Labs

LAB 2-2: Configure HR Resource

# Module 2: Self-assessment

- TODO

# Module 2: Summary

- Target systems, resources and connectors
- Built-in connectors
- Creating resource from scratch: use resource wizard
- Object types
- Synchronization configuration
- (Inbound) Mappings

# Module 2

End of module

# Module 3

Importing Source Data

# Tasks & Activities

- Activity represents the work to be done
  - E.g. import from resource or reconciliation
- Tasks are the actual vehicles making sure the work is done
  - Running in background
  - One-time or scheduled
- Execution state
  - **Runnable, Running, Suspended, Closed**

# Simulations – Why They Are Needed?

- Configuration is seldom correct for the first time
- Data can be unexpectedly deleted or modified by incorrect configuration
- Data can be unexpectedly modified if target system data is inconsistent
- Incremental adding of new features is **safer** with simulations

# Simulations, Part I

- MidPoint allows to simulate actions using simulations (“**what would be done**”)
- Actions are only reported, not actually executed
- Lifecycle state of midPoint configuration items
- Execution mode (Preview)
- Configuration to use (Development, Production)

# Lifecycle State

- Special property of objects and configuration items
- Represents state of **focal objects** (this boils down to either enabled or disabled)
- Represents state of **configuration items** e.g. resource, object type, attribute mapping, synchronization reaction etc. (this boils down to usage of configuration item for real actions, simulations or not at all)

# Lifecycle State (2)

- **Draft**: disabled, not active
- **Proposed**: for simulations
  - New resources are created in *Proposed* lifecycle state by default
- **Active**: used for normal operation (default)

Lifecycle state  
Proposed

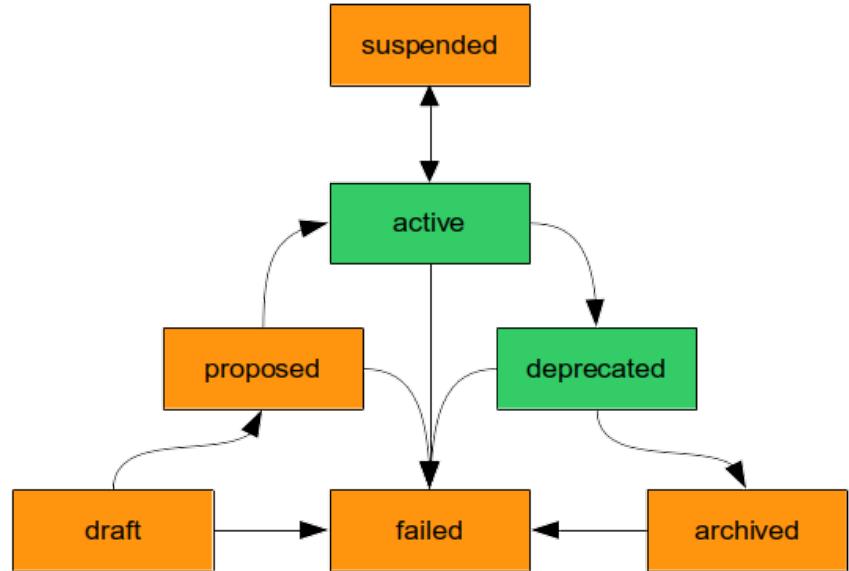
| Action      | Lifecycle state |  |
|-------------|-----------------|--|
| Add focus   | Active          |  |
| Unlink      | Active          |  |
| Synchronize | Active          |  |
| Link        | Active          |  |

Rows per page: 20 | 1 to 4 of 4 | << < 1 > >>

# Lifecycle State (3)

- **Suspended**: temporarily inactive
- **Deprecated**: still used, but being decommissioned
- **Archived**: decommissioned, inactive
- Docs: [Object Lifecycle](#)

ⓘ Draft, Suspended and Archive all mean “disabled” or “inactive”, just the reason is different.



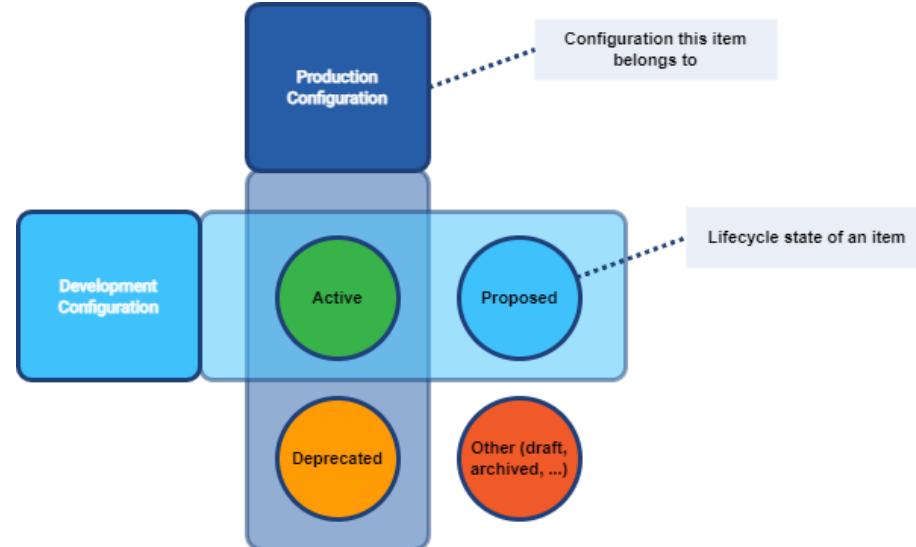
# Execution Mode

- Tasks executed in background can be used for simulations
- Task Execution mode:
  - **Preview**: simulate only
  - **Full** (or undefined): execute real actions (default)



# Configuration to Use

- **Production:** evaluates *Active* and *Deprecated* configuration items (lifecycle state)
- **Development:** evaluates *Active* and *Proposed* configuration items (lifecycle state)



⌚ Simulation executes actions using **Execution mode: Preview** with selected **Configuration to use**.

# Simulations: Adding Configuration Item to Production

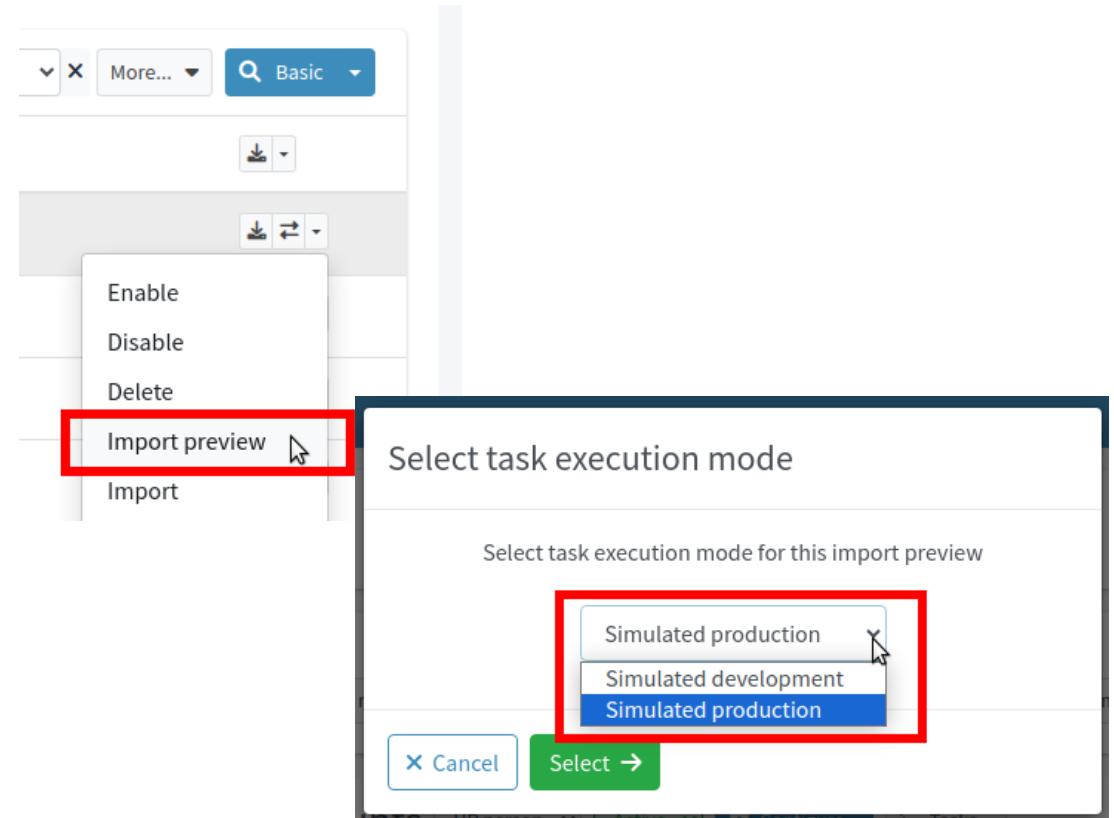
- Configuration item's lifecycle state: *Proposed*
- Run simulation task with Mode: Preview and configuration: Development
- Configuration items which are *Active* and *Proposed* will be simulated, no objects will be modified
- Configuration items with lifecycle state: *Proposed* are not used for real actions during normal operation (no simulation)
- When satisfied, set configuration item's lifecycle state to *Active*

# Simulations: Replacing Configuration Currently In Production

- Old configuration item's lifecycle state: *Deprecated*
- New (replacement) configuration item's lifecycle state: *Proposed*
- Run simulation task with Mode: Preview and configuration: Development
- Configuration items which are *Active* and *Proposed* (not *Deprecated*) will be simulated, no objects will be modified. You will see how your replacement would work.
- Configuration items with lifecycle state: *Deprecated* are still used for real actions during normal operation (no simulation)
- When satisfied, remove old configuration item (or put to *Archived*) and set replacement's lifecycle state to *Active*.

# Usage of Simulations in midPoint

- Single account import preview
- Import, Reconciliation, ...



# Simulation Result: List

← Back      Processed objects

Event mark **i** Undefined    State **i** Undefined    More...    Basic

| <input type="checkbox"/> | Name                                | Type   | State           | Changes                |  |  |
|--------------------------|-------------------------------------|--------|-----------------|------------------------|--|--|
| <input type="checkbox"/> | 1001<br>Focus activated             | User   | <b>Added</b>    | 8 Additions of total 8 |  |  |
| <input type="checkbox"/> | 1001 (Account 1001 (default) on HR) | Shadow | <b>Modified</b> | 2 Edits of total 2     |  |  |

Rows per page: 20    1 to 2 of 2    << < 1 > >>

# Simulation Result: Details

← Back

1001 (Simulation result: 2023-09-14T08:25:51.246Z)

| Processed object details |                 |
|--------------------------|-----------------|
| Type                     | User            |
| State                    | Added           |
| Marks                    | Focus activated |
| Projections              | 1               |

| Related objects |                                     |        |
|-----------------|-------------------------------------|--------|
|                 | 1001                                | User   |
|                 | 1001 (Account 1001 (default) on HR) | Shadow |

Changes

Simple Advanced

**Add User 1001**

| Item            | Value                 |
|-----------------|-----------------------|
| Name            | + 1001                |
| Lifecycle state | + active              |
| Given name      | + Geena               |
| Family name     | + Green               |
| Personal Number | + 1001                |
| Locality        | + Small Red Rock City |
| Projections     | + 1001 [Default]      |

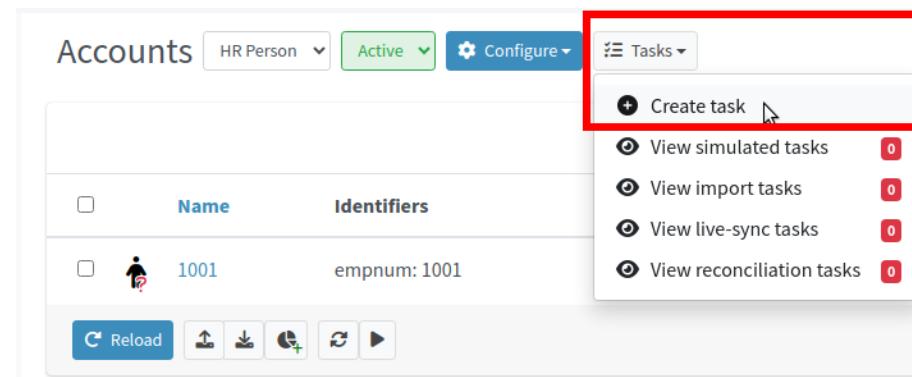
User was enabled.

# Import From Resource

- Import allows importing accounts from resource, following configuration of synchronization, mappings etc.
- Suitable for import from authoritative source of data
- Import will be executed in background using a task
- We will use wizard to create the task

# Simple Import Task Creation Wizard

- Available directly from list of accounts
- Allows to create import or other tasks with or without simulation options

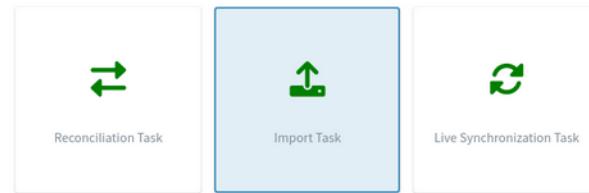


# Simple Import Task Creation Wizard (2)

- Multiple types of resource synchronization tasks supported
- Reconciliation
- Import
- Live Synchronization

## What type of task are you interested in?

Choose what type of task you are going to create and keep in mind that each type serves a specific purpose.



**Simulate task**  
With this on, no actions that could have persistent externally visible effects are executed. Accordingly, nothing is written to the system audit log.

OFF

[Close](#)

[Create task](#)

A modal dialog box with a light gray background. At the top left is the text "Simulate task" in bold. Below it is a descriptive sentence: "With this on, no actions that could have persistent externally visible effects are executed. Accordingly, nothing is written to the system audit log." To the right of the text is a checkbox labeled "OFF". At the bottom left is a blue "Close" button, and at the bottom right is a blue "Create task" button.

# Import Task Configuration, Part I.

- Specify basic configuration
  - “Name”: set your own name or keep empty (midPoint will generate task name using object type's Display name)

**Basic configuration**

Please fill in basic task attributes, such as name of the task, description of its purpose and/or owner.

**Basic**

Name [i](#)

Description [i](#)

Documentation [i](#)

Owner [i](#)  
administrator: UserType [Choose](#)

Category [i](#)

[Hide empty fields](#)

[Back](#) [Next: Resource objects](#)

# Import Task Configuration, Part II.

- Specify resource objects to process
- Already predefined from object type

**Resource objects**

Please specify which resource objects should be processed by the task. Select either a combination of kind/intent or specific objectClass or accept the defaults.

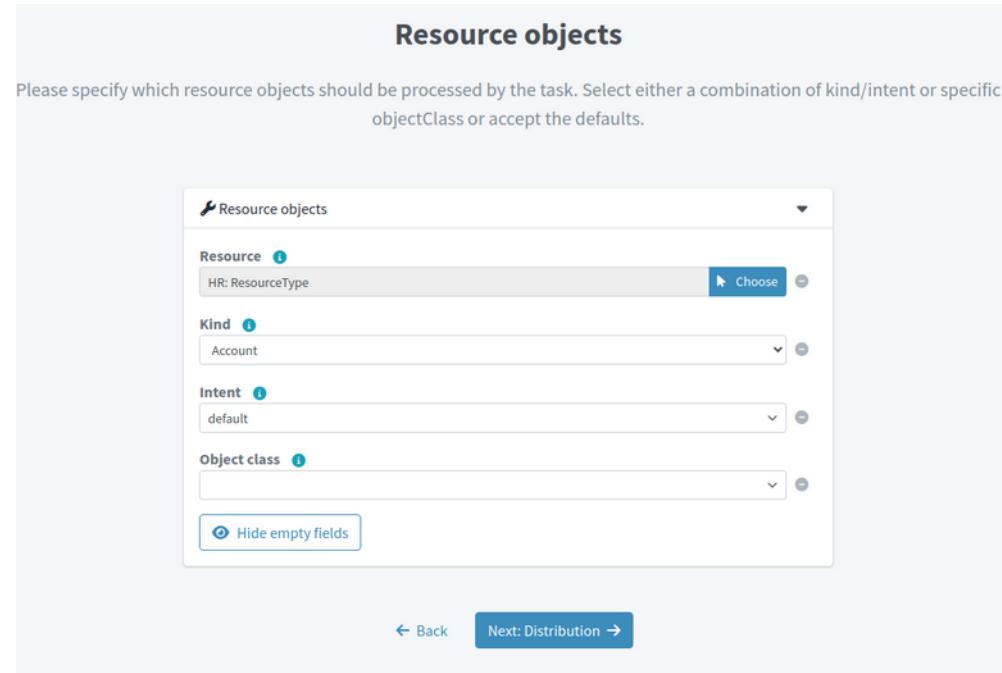
**Resource** i  Choose ...

**Kind** i  ...

**Intent** i  ...

**Object class** i  ...

← Back Next: Distribution →



# Import Task Configuration, Part III.

- Specify distribution parameters
  - “Worker threads” (optional)
- Click “Save & Run”

**Distribution**

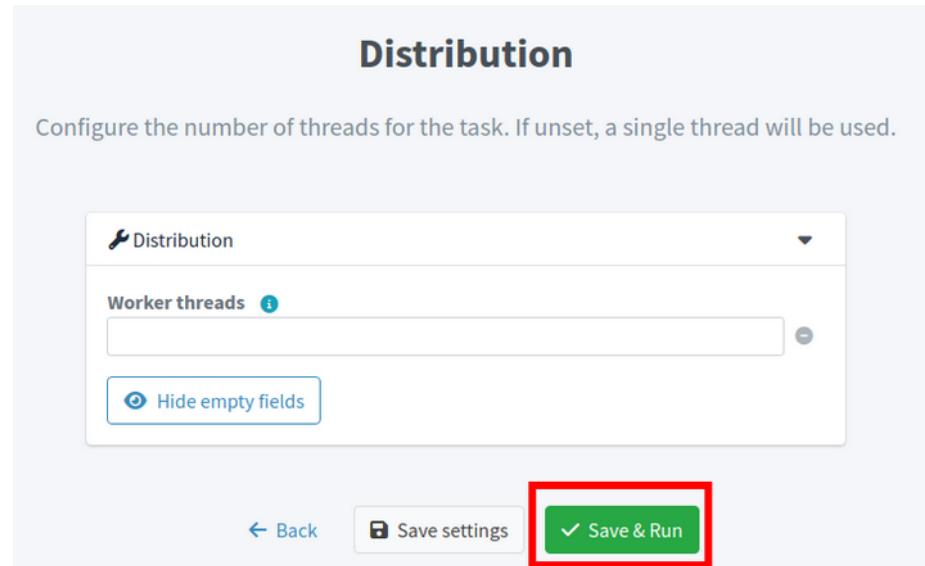
Configure the number of threads for the task. If unset, a single thread will be used.

**Distribution**

Worker threads i

Hide empty fields

[Back](#)



# Viewing Tasks

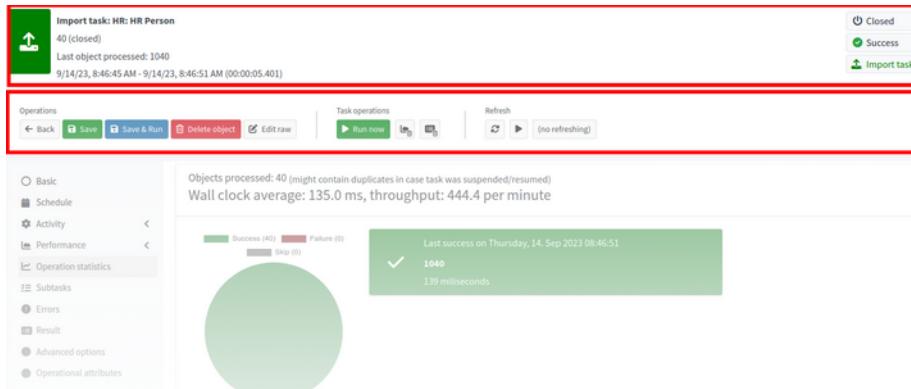
- Multiple ways to display the tasks
  - Tasks button > View import tasks
  - Defined Tasks menu item in resource
  - Server tasks > Import tasks
- Unless you have overridden the task name, it will use object type's Display name

The screenshot shows the 'Accounts' page with the 'HR Person' filter set to 'Active'. In the top right corner, there is a 'Tasks' dropdown menu. The 'View import tasks' option is highlighted with a red box and has a count of 1 next to it. Other options include 'Create task', 'View simulated tasks' (0), 'View live-sync tasks' (0), and 'View reconciliation tasks' (0).

The screenshot shows the 'Defined Tasks' section on the left, which includes 'Details', 'Basic', 'Connector configuration', and a 'Defined Tasks' section. The 'Defined Tasks' section is highlighted with a red box. Below it are categories: 'Accounts', 'Entitlements', 'Generics', 'Resource objects', and 'Schema handling'. To the right, a list of tasks is shown, each with an 'Import task: HR' prefix and a green upload icon. One specific task, 'Import task: HR: HR Person', is highlighted with a red box. At the bottom are buttons for 'Suspend', 'Resume', and 'Run now'.

# Import Task Details

- “Summary panel” shows task status (Running / Closed), result (Success) and also progress
- “Operation statistics” menu item shows more detailed information about processing



# Import Task Detail (2)

- **Synchronization situation transitions** allows you to understand what has happened during the synchronization
- Original state: before the task started
- Synchronization start: when the task started
- Synchronization end: after the task finished

| Synchronization situation transitions |                       |                     |                  |           |        |         |       |
|---------------------------------------|-----------------------|---------------------|------------------|-----------|--------|---------|-------|
| Original state                        | Synchronization start | Synchronization end | Exclusion reason | Succeeded | Failed | Skipped | Total |
| Unmatched                             | Unmatched             | Linked              |                  | 1         | 0      | 0       | 1     |
| No record                             | Unmatched             | Linked              |                  | 39        | 0      | 0       | 39    |

# Import Task Detail (3)

- Example:

- 1 account was already previewed (and linked)
- 39 more accounts were linked during the import task execution
- $1+39=40$  accounts are now linked to their owners in midPoint

## Synchronization situation transitions

| Original state | Synchronization start | Synchronization end | Exclusion reason | Succeeded | Failed | Skipped | Total |
|----------------|-----------------------|---------------------|------------------|-----------|--------|---------|-------|
| Unmatched      | Unmatched             | Linked              |                  | 1         | 0      | 0       | 1     |
| No record      | Unmatched             | Linked              |                  | 39        | 0      | 0       | 39    |

# Users and Accounts

- **Users are objects in midPoint, representing identities**
  - User belong to Focal objects (also Organizations, Roles, Services)
- **Accounts are resource objects**
- Accounts belonging to specific users are **projections** of the users

# Users and Accounts (2)

- Users have *properties* (Name, Given Name, Family Name, ...)
- Users have references to their accounts (linkRef)
- Properties are addressed using their paths
  - **name** is user's username
  - **fullName** is user's Full Name
  - **metadata / createTimestamp** is user's creation timestamp

## User

```
name = "abaker"  
givenName = "Alice"  
familyName = "Baker"  
fullName = "Alice Baker"  
metadata:  
    createTimestamp= "2023-01-..."  
    ...  
linkRef = "cn=Alice Baker, ..."
```

# Users and Accounts (3)

- Accounts have *attributes*, which are not copied to midPoint repository
- midPoint fetches real account data whenever needed (online)
- MidPoint stores **account identifiers** and **metadata** for accounts in its repository: intermediate Shadow objects

# Shadow Object

- Name (corresponds to **resource account identifier**)
- Metadata (create, modify, creator, modifier, synchronization situation,...)
- Resource & Object type reference (resource, object class, kind, intent)
- **Account identifiers**
  - Resource-specific! Real AD: **dn, objectGUID**

## Shadow

```
name = "cn=Alice Baker, ..."  
metadata:  
    createTimeStamp = "2023-08-01T12:00:00"  
    modifyTimeStamp = "2023-08-01T13:00:00"  
resourceRef = "AD"  
objectClass = "inetOrgPerson"  
kind = "account"  
synchronizationSituation = "linked"  
...  
attributes:  
    ri:dn = "cn=Alice Baker, ..."  
    ri:entryUUID = "b3a5c588-e003-103d-8175-  
5d0d35036f2f"
```

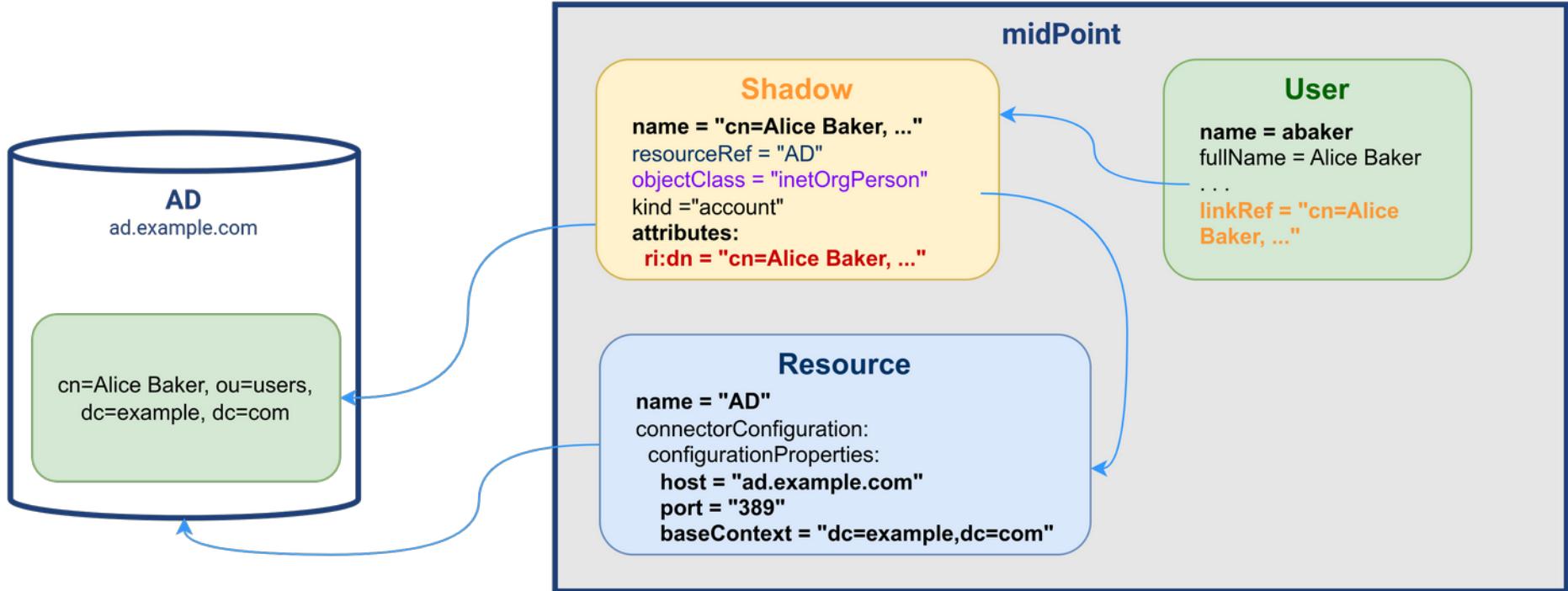
# Shadow Object (2)

- MidPoint does **not** store all account attributes here!
- Pending changes can be *temporarily* stored here (e.g. saving data to resource which is down)

## Shadow

```
name = "cn=Alice Baker, ..."  
metadata:  
    createTimeStamp = "2023-08-01T12:00:00"  
    modifyTimeStamp = "2023-08-01T13:00:00"  
resourceRef = "AD"  
objectClass = "inetOrgPerson"  
kind ="account"  
synchronizationSituation = "linked"  
...  
attributes:  
    ri:dn = "cn=Alice Baker, ..."  
    ri:entryUUID = "b3a5c588-e003-103d-8175-  
5d0d35036f2f"
```

# User / Account Links Implementation



# User / Account Links Implementation (2)

- MidPoint read information about user's linked accounts (**linkRef**) – pointing to Shadow objects
- MidPoint reads information from Shadow objects (**resource, object class, account identifier**)
- MidPoint reads information about **resource** (how to connect there)
- MidPoint **uses connector to connect to the resource** and gets information about specific account using **object class, account identifier**

# Users in midPoint – List

The screenshot shows the midPoint user management interface. The left sidebar has a 'SELF SERVICE' section with Home, Profile, Credentials, and Request access; and an 'ADMINISTRATION' section with Dashboards, Users (which is selected and highlighted with a red border), All users (also highlighted with a red border), and Persons. The main area is titled 'All users' and contains a table of users. The table columns are Name, Personal Number, Full name, Email, and Accounts. A search bar at the top allows filtering by Object collection, Full name, Name, and More... The table rows show users 1001 through 1008, each with a delete icon and a context menu icon.

| Name | Personal Number | Full name | Email | Accounts | Action |
|------|-----------------|-----------|-------|----------|--------|
| 1001 | 1001            |           |       | 1        |        |
| 1002 | 1002            |           |       | 1        |        |
| 1003 | 1003            |           |       | 1        |        |
| 1004 | 1004            |           |       | 1        |        |
| 1005 | 1005            |           |       | 1        |        |
| 1006 | 1006            |           |       | 1        |        |
| 1007 | 1007            |           |       | 1        |        |
| 1008 | 1008            |           |       | 1        |        |

# Users in midPoint – After Import

- Users are created from HR data
- Username and some properties have been imported
  - Username = `empnum` from HR data
  - Lifecycle status is based on `status` from HR data (Active, Suspended, Archived)
  - Full name is not yet populated
  - Archetype is not set
- Users have linked their HR accounts

# Users in midPoint – After Import (2)

Users > All users > Edit user

Enabled    No assignments    No organizations

Operations: Back, Save, Preview changes, Delete object, Edit raw, Options

Properties:

- Basic (highlighted with a red box)
- Projections (1)
- All accesses
- Assignments (0)
- Activation
- Password
- History
- Cases (0)
- Personas (0)
- Delegations (0)
- Delegated to me (0)
- Triggers (0)

| ID | First name | Surname | Art name | Emp type | Job | Employee number | Locality            | Country          | Status | Action |
|----|------------|---------|----------|----------|-----|-----------------|---------------------|------------------|--------|--------|
| 1  | Geena      | Green   |          | FTE      | CEO | 1001            | Small Red Rock City | _loc:Rocky State | In     | Modify |

# User's Accounts (Projections) After Import

- Data about account attributes are not stored in midPoint
- Clicking on account details will fetch data from source/target system

The screenshot shows the midPoint administrative interface. On the left, a sidebar menu is open with the following options: Basic (selected), Projections (highlighted with a red box), All accesses, Assignments (with a count of 0), Activation, and Password.

The main area displays a table of user accounts. The columns are: Resource (with a checkbox), Object type, and Pending operation. A single row is selected and highlighted with a red box. The selected row contains the resource ID '1001', object type 'HR Person', and pending operation status.

| Resource                 |      | Object type | Pending operation   |
|--------------------------|------|-------------|---|
| <input type="checkbox"/> | 1001 | HR Person   | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> |

At the bottom, there is a navigation bar with 'Rows per page' set to 20, '1 to 1 of 1', and a page number '1'. Below this is a detailed table of account attributes:

| ID | First name | Surname | Art name | Emp type | Job | Employee number | Locality            | Country          | Status |        |
|----|------------|---------|----------|----------|-----|-----------------|---------------------|------------------|--------|--------|
| 1  | Geena      | Green   |          | FTE      | CEO | 1001            | Small Red Rock City | _loc:Rocky State | In     | Modify |

# User's Account (Projection) Detail – Attributes

The screenshot shows a user interface for managing account attributes. On the left, a sidebar lists various account management options: Basic, Projections (selected), All accesses, Assignments, Activation, Password, History, Cases, Personas, Delegations, Delegated to me, Triggers, and Applications. The 'Projections' section has a count of 1. In the main area, a header shows '1001' with a person icon and 'Resource: HR HR Person'. Below this, a 'Basic' tab is selected, and an 'Attributes' section is expanded, also highlighted with a red box. The attributes listed are: empnum\*, surname, locality, status, emptytype, country, firstname, and job. Each attribute has a corresponding input field to its right. At the bottom of the attribute list, there is a link 'Show empty fields'.

| ID | First name | Surname | Art name | Emp type | Job | Employee number | Locality            | Country          | Status |        |
|----|------------|---------|----------|----------|-----|-----------------|---------------------|------------------|--------|--------|
| 1  | Geena      | Green   |          | FTE      | CEO | 1001            | Small Red Rock City | _loc:Rocky State | In     | Modify |

# Archetype Information in GUI

- Reprise: archetype is “object category” in midPoint
- Person archetype is built-in in midPoint
- Archetype is indicated in user list
  - Color, icon, tooltip with archetype name (“Person”) over icon
- Archetype is indicated in user's summary panel
  - Color, icon, archetype name (“Person”)



# Archetyped Users in midPoint – List

The screenshot shows the midPoint web interface with the title "Persons". The left sidebar has a "Users" dropdown menu open, with "All users" and "Persons" selected. The main area displays a table of users with columns: Name, Personal Number, Full name, Email, and Accounts. A red box highlights the first row for user 1001, Geena Green. The table contains the following data:

|                          | Name | Personal Number | Full name       | Email | Accounts | Action |
|--------------------------|------|-----------------|-----------------|-------|----------|--------|
| <input type="checkbox"/> | 1001 | 1001            | Geena Green     |       | 1        |        |
| <input type="checkbox"/> | 1002 | 1002            | Ana Lopez       |       | 1        |        |
| <input type="checkbox"/> | 1003 | 1003            | Jimmy Taylor    |       | 1        |        |
| <input type="checkbox"/> | 1004 | 1004            | Peter Hunter    |       | 1        |        |
| <input type="checkbox"/> | 1005 | 1005            | Emanuel Young   |       | 1        |        |
| <input type="checkbox"/> | 1006 | 1006            | Martin Knight   |       | 1        |        |
| <input type="checkbox"/> | 1007 | 1007            | Diane Davis     |       | 1        |        |
| <input type="checkbox"/> | 1008 | 1008            | Elisabeth Mason |       | 1        |        |

# Archetyped User Details

- Summary panel updated
- Archetype icon/color
- Archetype name
- ⓘ Full name computed

The screenshot shows the 'Edit Person' screen for 'Person 1001'. A red box highlights the top navigation bar and summary panel. Another red box highlights the 'Basic' tab in the left sidebar. The main area displays various properties:

| Name            | Value               |
|-----------------|---------------------|
| Name            | 1001                |
| Lifecycle state | Active              |
| Full name       | Geena Green         |
| Given name      | Geena               |
| Family name     | Green               |
| Personal Number | 1001                |
| Locality        | Small Red Rock City |

Below the properties, there is a link to 'Show empty fields'.

# TODO makes even sense? Methodology: Connecting Source System, Part II.

- TODO steps needed to connect, basically what will be done in the labs, in generic way.

## Module 3: Labs

LAB 3-1: Single Source System Entry Import Simulation

# Module 3: Labs

LAB 3-2: Source System Data Import

# Module 3: Self-assessment

- TODO

# Module 3: Summary

- Simulations, lifecycle state, execution mode, configuration to use
- Controlled way of replacing configuration (deprecated and proposed)
- Import from resource
- Users and accounts, shadows
- Archetype introduction

# Module 3

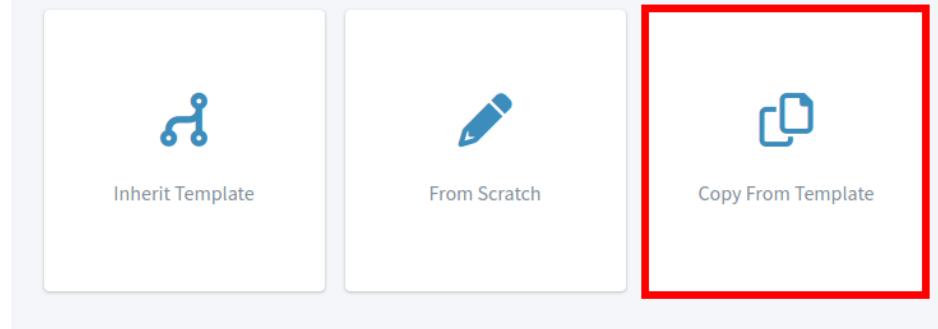
End of module

# Module 4

Connecting Target System

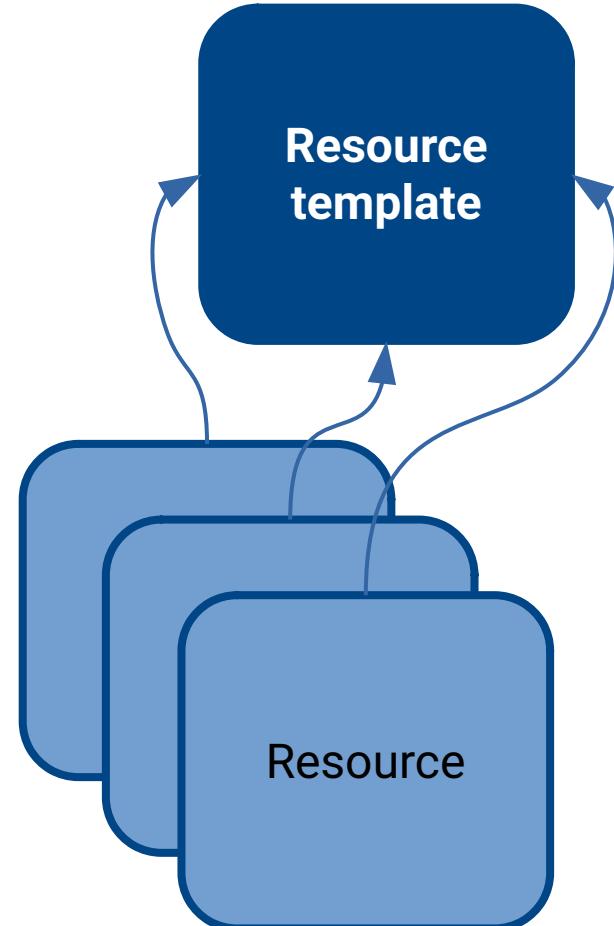
# Resource Templates

- Faster creation of resources (configuration is already present)
- **Inherit template:** use if you also create the template itself
  - Resources will inherit the configuration, “referring” to the configuration of template
- **Copy from template:** use if the template is not managed by you, e.g. midPoint built-in objects
  - Resources will be created as a copy of template



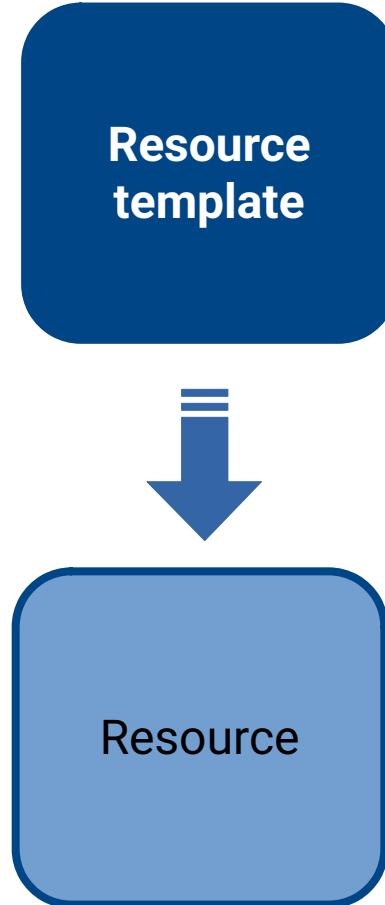
# Resource Template: Inheritance

- Example: you need to manage 100 instances of LDAP servers with the same configuration
- You will create one template and 100 resources using that template
- Resources reference the template
- Any change in template is automatically inherited by all resources



# Resource Template: Copy

- Example: you want to use midPoint built-in resource template as a starting point
- You will create the resource as a copy of template
- Resource does **not** reference the template
- Any change in template is **not** automatically inherited by any resource based on the template



# Using Resource Templates

- Resource wizard will ask for connection properties (e.g. host, port, etc.)
- Parts of configuration may be in *Proposed* or even *Draft* lifecycle state
- Use only configuration that you need, extend as you need (iterations)
- You are encouraged to use simulations

# Creating AD Resource From Resource Template, Part I.

- Specify basic information
  - “Name”
  - Other properties are copied from template
  - “Lifecycle status” is set to *Proposed*

**Basic information about the resource**

Fill in basic information about your resource and for more optional settings click on "Show empty fields" button

**Basic information**

Name i  
AD

Description i  
Resource template for a target Active Directory simulated by OpenLDAP (inetOrgPerson and groupOfNames).

Documentation i  
This is a resource template for Active Directory simulated by OpenLDAP for training purposes:

Lifecycle state i  
Proposed

Show empty fields

← Back      Next: Configuration →

# Creating AD Resource From Resource Template, Part II.

- Specify connection information
- The properties are defined in resource template for you to fill

**Establish a connection**

Fill in the fields down below with correct information which midPoint needs to access your resource in order to continue in the next step

**Configuration**

Host: ad

Port number: 389

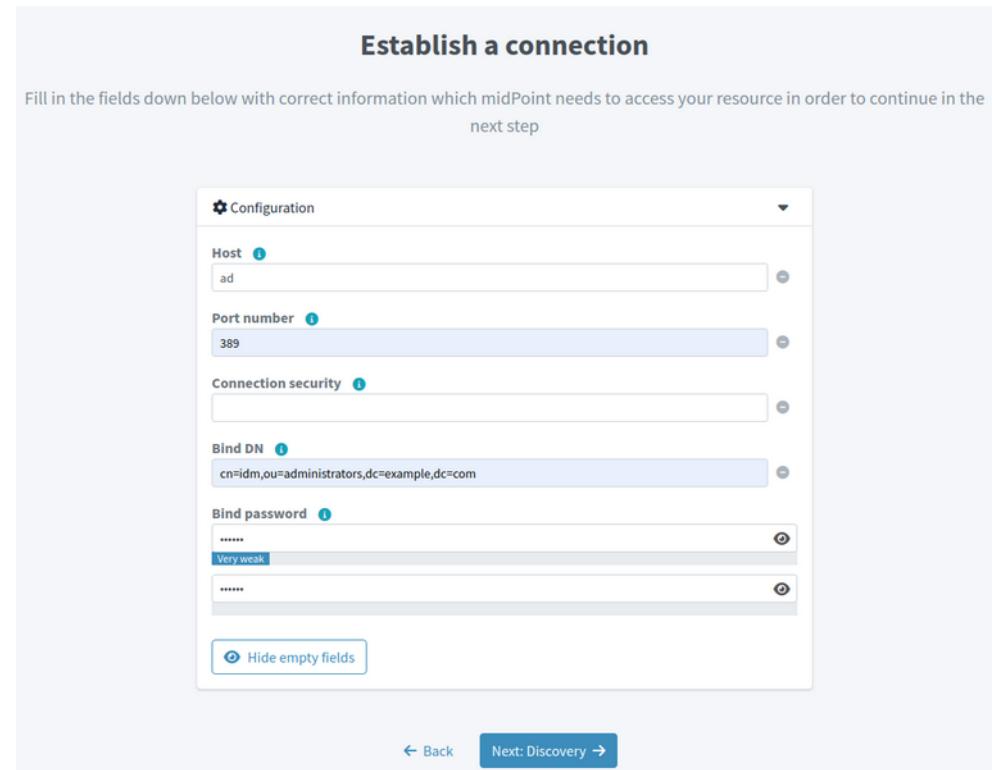
Connection security:

Bind DN: cn=idm,ou=administrators,dc=example,dc=com

Bind password: Very weak

[Hide empty fields](#)

[Back](#) [Next: Discovery](#)



# Creating AD Resource From Resource Template, Part III.

- Specify additional information
  - AD: check/select “Base context”
- We are simulating AD with OpenLDAP

**MidPoint discovery**

Your resource has gone through the discovery process and midPoint found following parameters for you that you can configure

**Configuration**

**Password hash algorithm**: SSHA

**Paging strategy**: auto

**Base context**: dc=example,dc=com

**VLV sort attribute**: uid

**VLV ordering rule**: 2.5.13.3

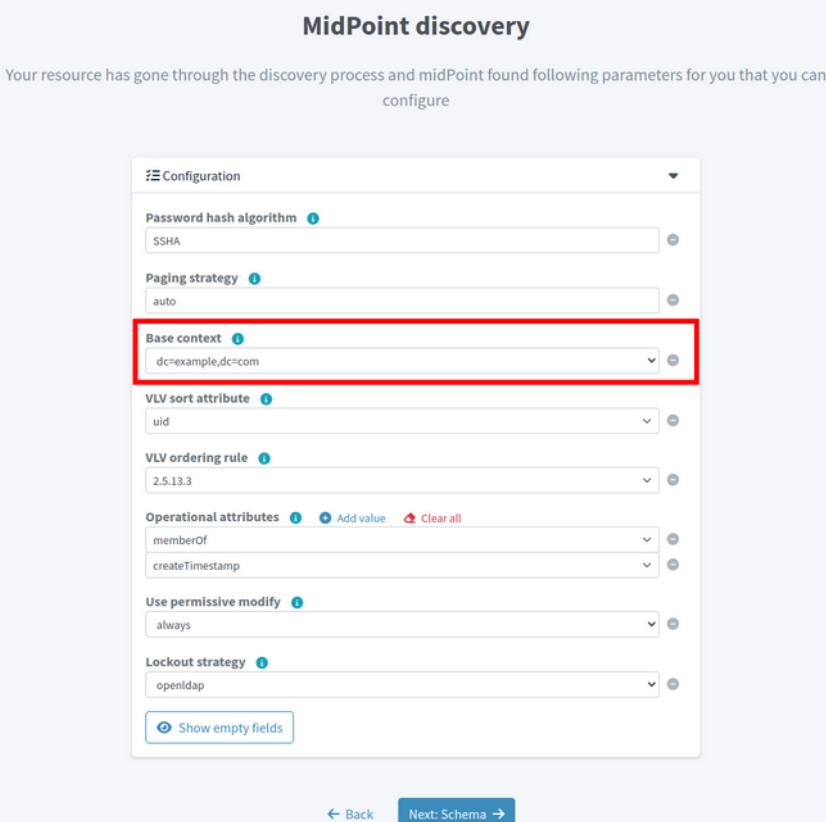
**Operational attributes**: memberOf, createTimeStamp

**Use permissive modify**: always

**Lockout strategy**: openldap

**Show empty fields**

[Back](#) [Next: Schema](#)



# Creating AD Resource From Resource Template, Part IV.

- Check schema of object types
- Some object classes are predefined in resource template

**Schema of object types**

Select all the available object classes from the list below that your future resource should work with

| Schema   |                    |            |
|--|--------------------|------------|
| Selected item  |                    |            |
| <input checked="" type="checkbox"/> organizationalUnit | Native name        | Type       |
| <input checked="" type="checkbox"/> groupOfNames       | groupOfNames       | Structured |
| <input checked="" type="checkbox"/> groupOfUniqueNames | groupOfUniqueNames | Structured |
| <input checked="" type="checkbox"/> inetOrgPerson      | inetOrgPerson      | Structured |
| <input checked="" type="checkbox"/> organizationalUnit | organizationalUnit | Structured |

Rows per page: 20 | 1 to 4 of 4 | << | < | 1 | > | >>

← Back |  Create resource

# Previewing AD Data

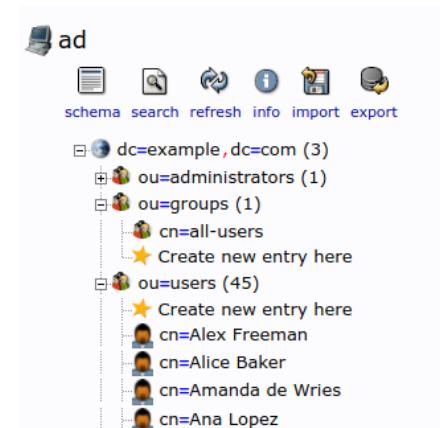
- There are existing accounts and groups

Resource data preview

Here you can see a list of all objects that are on the resource

List of data: inetOrgPerson

|                          | Identifiers   | Situation | Owner | Pending operations |
|--------------------------|---|-----------|-------|--------------------|
| <input type="checkbox"/> | cn=Alice Baker,ou=users,dc=example,dc=com<br>entryUUID: b3a5c588-e003-103d-8175-5d0d35036f2f<br>dn: cn=Alice Baker,ou=users,dc=example,dc=com<br>uid: abaker            |           |       |                    |
| <input type="checkbox"/> | cn=Amanda de Wries,ou=users,dc=example,dc=com<br>entryUUID: b3abfb174-e003-103d-817e-5d0d35036f2f<br>dn: cn=Amanda de Wries,ou=users,dc=example,dc=com<br>uid: adewries |           |       |                    |



# AD Resource Object Types

- The object type configuration has been copied from the resource template

New resource / AD / Object types

### Object type manager

Here is a table with all the objects available in the selected resource, manage existing or create a new one

| Object types                    |             |         |         |             |                 |  |
|---------------------------------|-------------|---------|---------|-------------|-----------------|--|
| Display name                    | Kind        | Intent  | Default | Description | Lifecycle state | -  |
| Normal Account                  | ACCOUNT     |         | true    |             | Proposed        | -  |
| AD Group                        | ENTITLEMENT | adGroup | true    |             | Proposed        | -  |
| <a href="#">Add object type</a> |             |         |         |             |                 | Rows per page 20 1 to 2 of 2 << < 1 > >> |

# Object Type: Correlation

- Mechanism to **find the resource object (account) owner** in midPoint
- Multiple correlators are possible
- Smart correlation: algorithm or approximate search with additional operator interaction
- ⚠ Correlation is not used for already linked accounts
- ⚠ Usually not needed for source of data

**Correlation rules**

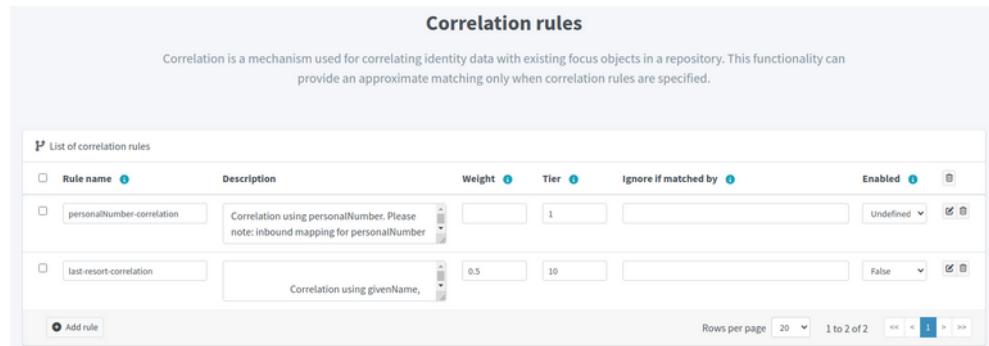
Correlation is a mechanism used for correlating identity data with existing focus objects in a repository. This functionality can provide an approximate matching only when correlation rules are specified.

**List of correlation rules**

| Rule name                  | Description   | Weight | Tier | Ignore if matched by | Enabled                                       |
|----------------------------|---|--------|------|----------------------|---|
| personalNumber-correlation | Correlation using personalNumber. Please note: inbound mapping for personalNumber | 1      |      |                      | Undefined <input checked="" type="checkbox"/> |
| last-resort-correlation    | Correlation using givenName,  | 0.5    | 10   |                      | False <input checked="" type="checkbox"/>     |

**Add rule**

Rows per page: 20 | 1 to 2 of 2 | << < > >>



# Items Correlator

- Correlation using user property, for which a (inbound) mapping is already defined
- Example: **personalNumber**
- Unlike ordinary inbound mappings, **this mapping is evaluated only during correlation**

**Configuration of correlation items**

Add and specify correlation item(s) to make the correlation rule work correctly

List of correlation items

| Item           | Search method | Match threshold | Inclusive |
|----------------|---------------|-----------------|-----------|
| personalNumber | Exact match   |                 |           |

Reset path

Add correlator

Rows per page: 20 | 1 to 1 of 1 | << < > >>

**Inbound mappings (to MidPoint)**

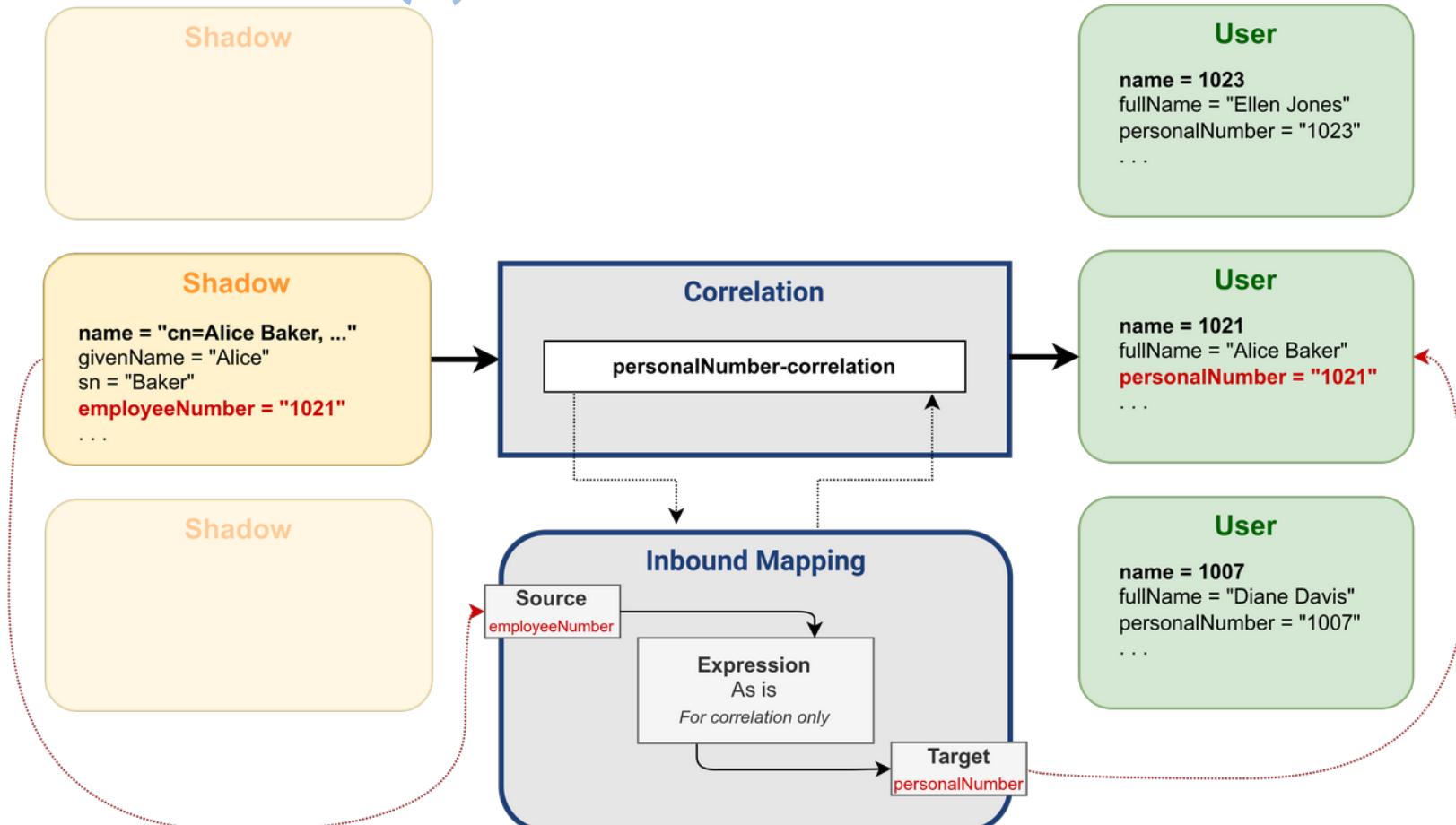
| Name   | From resource attribute | Expression | Target         | Lifecycle state |
|--|-------------------------|------------|----------------|-----------------|
| mapping-Inbound-familyName-for                             | sn                      | As is      | familyName     | Active          |
| mapping-Inbound-givenName-for                              | givenName               | As is      | givenName      | Active          |
| mapping-Inbound-locality-for-corr                          | l                       | As is      | locality       | Active          |
| <b>mapping-Inbound-employeeNum</b><br>Used for correlation | employeeNumber          | As is      | personalNumber | Active          |

Add inbound

Rows per page: 20 | 1 to 4 of 4 | << < > >>

**mapping-inb  
Used for correlation**

# Items Correlator (2)



# Items Correlator: Example

- There is an inbound mapping from AD's `employeeNumber` → user's `personalNumber` marked to be evaluated only during correlation
- MidPoint interprets this as searching for midPoint user with the same `personalNumber` as AD's `employeeNumber`
  - We expect this will match either 0 or 1 user
- If potential owner has been found, the account situation is: **Unlinked**
- If no owner has been found, the account situation is: **Unmatched** (orphaned account)
- Synchronization reaction will be applied for the situations (if configured)

# Approximate Matching With Human Interaction

- Correlation using user properties, for which (inbound) mappings are already defined
- Example: **givenName AND familyName AND locality**
- Unlike ordinary inbound mappings, **these mappings are evaluated only during correlation**

**Configuration of correlation items**

Add and specify correlation item(s) to make the correlation rule work correctly

| Item       | Search method | Match threshold | Inclusive |
|------------|---------------|-----------------|-----------|
| givenName  | Exact match   |                 |           |
| familyName | Exact match   |                 |           |
| locality   | Exact match   |                 |           |

**Add correlator**

**Inbound mappings (to MidPoint)**

| Name                                    | From resource attribute * | Expression | Target         | Lifecycle state |
|---|---------------------------|------------|----------------|-----------------|
| mapping-inbound-familyName-for-sn       | sn                        | As Is      | familyName     | Active          |
| mapping-inbound-givenName-for-givenName | givenName                 | As Is      | givenName      | Active          |
| mapping-inbound-locality-for-l          | l                         | As Is      | locality       | Active          |
| mapping-inbound-employeeNumt            | employeeNumber            | As Is      | personalNumber | Active          |

**Used for correlation**

**mapping-inbound**

# Items Correlator (Approximate Matching): Example

- There are inbound mappings marked to be evaluated only during correlation:
  - from AD's **givenName** → user's **givenName**
  - from AD's **sn** → user's **familyName**
  - from AD's **locality** → user's **locality**
- MidPoint interprets this as searching for midPoint user with the same values for all three properties

## Items Correlator: Approximate Matching: Example (2)

- We are not sure this will match proper users, therefore we set weight between 0.0 – 1.0 (0.5)
- If potential owner has been found, the account situation is: **Disputed**
- Synchronization configuration will be applied for the situation (if configured)

# Multiple Correlators Evaluation

- Correlation rule has a “Tier” identifier, starting with lowest number
- Correlation rule has a “Weight” (default: 1.0) – confidence if matched
  - Confidence value 1.0 means the account owner has been found
  - Lower confidence (between 0.0 – 1.0) means midPoint is not sure
- If account owner has not been found, midPoint continues with processing higher tiers
- If resulting “Weight” is between 0.0 – 1.0, the candidate may be selected by human operator (synchronization configuration is needed too)

# Multiple Correlators – Screenshot

**Correlation rules**

Correlation is a mechanism used for correlating identity data with existing focus objects in a repository. This functionality can provide an approximate matching only when correlation rules are specified.

| Rule name                  | Description   | Weight | Tier | Ignore if matched by | Enabled   |
|----------------------------|---|--------|------|----------------------|-----------|
| personalNumber-correlation | Correlation using personalNumber. Please note: inbound mapping for personalNumber |        | 1    |                      | Undefined |
| last-resort-correlation    | Correlation using givenName,  | 0.5    | 10   |                      | False     |

**Add rule**      Rows per page: 20      1 to 2 of 2      << < 1 > >>

# Object Type: Synchronization (Extended)

- Correlation and synchronization settings are related
- Synchronization settings are extended for more situations

| List of reactions        |                       |                              |                         |                                  |  |
|--------------------------|-----------------------|------------------------------|-------------------------|----------------------------------|--|
| <input type="checkbox"/> | Name <small>i</small> | Situation <small>* i</small> | Action <small>i</small> | Lifecycle state <small>i</small> | <small>trash</small>                     |
| <input type="checkbox"/> | <input type="text"/>  | Unlinked                     | Link                    | Proposed                         | <small>edit</small> <small>trash</small> |
| <input type="checkbox"/> | <input type="text"/>  | Linked                       | Synchronize             | Proposed                         | <small>edit</small> <small>trash</small> |
| <input type="checkbox"/> | <input type="text"/>  | Deleted                      | Synchronize             | Proposed                         | <small>edit</small> <small>trash</small> |
| <input type="checkbox"/> | <input type="text"/>  | Unmatched                    | Delete resource object  | Proposed                         | <small>edit</small> <small>trash</small> |

plus Add simple reaction      Rows per page: 20 < 1 > >>

# Synchronization Situations and Reactions (Extended)

- You need to select appropriate actions for specific situations
  - Linked
  - Unlinked
  - Unmatched
  - Deleted
  - Disputed
- ⚡ The default configuration is copied from resource template

# Situation: Linked

- The resource account is already linked to midPoint user
- Recommended actions:
  - Synchronize (data will synchronize using mappings)

# Situation: Unlinked

- The resource account *could be* linked to midPoint user, because midPoint determined who should be using correlation, but the link does not exist yet
- Recommended actions:
  - Link

# Situation: Unmatched

- The resource account owner could not be determined
- Recommended actions:
  - Add focus (for authoritative source systems)
  - Delete resource object (for non-authoritative systems)
  - Inactivate resource object (for non-authoritative systems)
  - (no action)

ⓘ **Focus** means  
User in this  
context

# Situation: Deleted

- The resource account has been deleted
- Recommended actions:
  - Synchronize (for non-authoritative systems)
  - Inactivate focus (for authoritative systems)
  - Delete focus (for authoritative systems)
  - (no action)

# Situation: Disputed

- The resource account cannot be correlated (e.g. correlation returns multiple candidate owners or correlation confidence is too low)
- Recommended actions:
  - Create correlation case (human interaction, administrator decides)
  - (no action)

# Why Correlation Was Not Needed for HR

- We did not need correlation for source system and only used situations Linked and Unmatched as we simply created and updated users in midPoint
  - “If owner for the account does not yet exist in midPoint, create a new user”
- Correlation is not evaluated for already linked accounts
  - “If the owner for the account already exist, update the user”
- Data in HR is never deleted
- As a result, we did not need actions for Unlinked, Deleted and Disputed

# How Resource Template Can Help

- Configuring everything from scratch would take time (and effort)
- Resource templates can be prepared by Evolveum, partners, administrators, ...
- Resource template can define object types and their correlation, synchronization, mappings etc.
- You can re-use the resource template or take it as a starting point
- We will uncover the rest of the configuration in later modules

## Module 4: Labs

LAB 4-1: Creating Active Directory Resource From Template

## Module 4: Labs

LAB 4-2: Reviewing Active Directory Resource Synchronization Configuration

# Module 4: Self-assessment

- TODO

# Module 4: Summary

- Creating new resource from template: use resource wizard
- Resource template: copy from vs inherit
- (Smart) Correlation – Items, approximate with human interaction
- Synchronization configuration (extended)

# Module 4

End of module

# Module 5

Target System Integration

# Correlation With AD

- We are about to correlate existing AD accounts with midPoint owners
- AD resource is in *Proposed* lifecycle state, suitable for simulation
- **We want to make sure we don't modify / delete anything in AD during the process**

# Reconciliation With Resource

- Reconciliation compares the state of accounts in resource vs. midPoint, following configuration of correlation, synchronization, mappings etc.
- Suitable for both source and target resources
- Reconciliation will be executed in background using a task – first using the simulation
- We will use wizard to create the task

# Reconciliation With Resource (2)

- Reconciliation run in **multiple phases**
- Finish pending operations (data cached in midPoint Shadow objects)
- List resource objects (e.g. accounts), synchronize each
- Detect missing resource objects (e.g. deleted accounts)

# Simple Reconciliation Task Creation Wizard

- Multiple types of resource synchronization tasks supported
- Reconciliation
- Import
- Live Synchronization

## What type of task are you interested in?

Choose what type of task you are going to create and keep in mind that each type serves a specific purpose.



Simulate task

With this on, no actions that could have persistent externally visible effects are executed. Accordingly, nothing is written to the system audit log.

ON

[Close](#)

[Create task](#)

A screenshot of a modal dialog box. It contains a checkbox labeled 'Simulate task' with the description 'With this on, no actions that could have persistent externally visible effects are executed. Accordingly, nothing is written to the system audit log.' A switch button next to the checkbox is set to 'ON'. At the bottom left is a 'Close' button, and at the bottom right is a 'Create task' button.

# Reconciliation Task Configuration, Part I.

- Specify basic configuration

- “Name”: we will set our own name because we will have multiple reconciliation tasks for the same resource – to use with and without simulation

**Basic configuration**

Please fill in basic task attributes, such as name of the task, description of its purpose and/or owner.

**Basic**

Name i  
Reconciliation with AD - development simulation Edit

Description i

Documentation i

Owner i  
administrator: UserType Choose

Category i

Hide empty fields

[Back](#) [Next: Resource objects](#)

# Reconciliation Task Configuration, Part II.

- Specify resource objects to process
- Already predefined from object type

**Resource objects**

Please specify which resource objects should be processed by the task. Select either a combination of kind/intent or specific objectClass or accept the defaults.

**Resource** i  
AD: ResourceType [Choose](#)

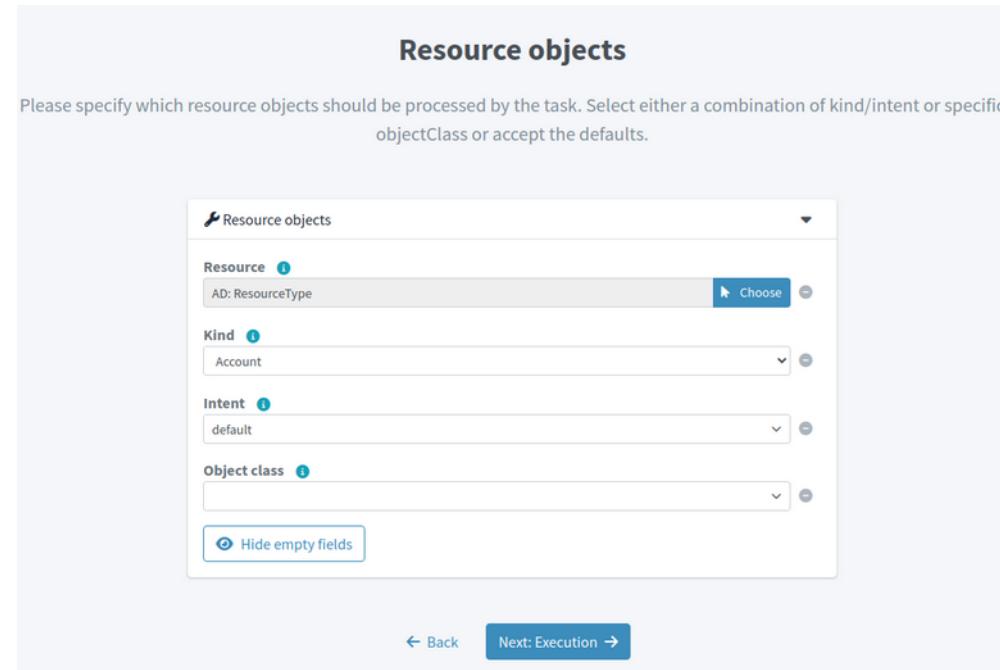
**Kind** i  
Account

**Intent** i  
default

**Object class** i

Hide empty fields

[← Back](#) [Next: Execution →](#)

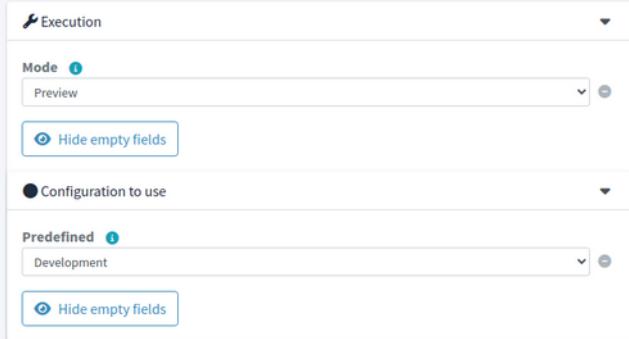


# Reconciliation Task Configuration, Part III.

- Specify execution options for simulated reconciliation
  - “Mode”: **Preview** (only simulate)
  - “Configuration to use / Predefined”: **Development**
- Development = configuration items in *Active / Proposed* lifecycle state

**Execution options**

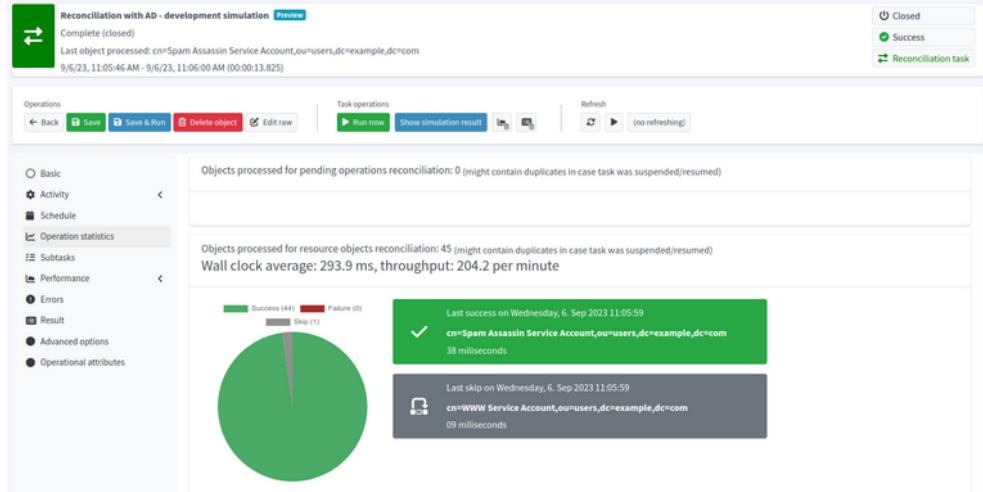
Please specify task execution options. Determine if the task should execute changes (Full) or just simulate the operations for later analysis. For simulation use Preview, or Shadow management preview - only shadow simulation (e.g. correlation rules,...). Please specify also Configuration to use: Production to use only the active configuration of the resource or Development to use the active and proposed configuration of the resource during the simulations.



The screenshot shows a configuration dialog titled "Execution". It has two main sections: "Mode" and "Predefined". Under "Mode", the dropdown is set to "Preview". Under "Predefined", the dropdown is set to "Development". Both sections have a "Hide empty fields" button. Below these sections are dropdown menus for "Configuration to use" and "Predefined". At the bottom right of the dialog are "Back" and "Next: Schedule" buttons.

# Reconciliation Task Details

- “Summary panel” shows task status (Running / Closed), result (Success) and also progress
- “Operation statistics” menu item shows more detailed information about processing



# Understanding Correlation Results

- Synchronization situations transitions is now more interesting...
  - 39 accounts are (would be) linked (because we're just simulating) during the reconciliation task execution
  - 5 accounts are Unmatched (orphaned)
  - 1 account is reported as Protected

| Synchronization situation transitions |                       |                     |                  |           |        |         |       |
|---------------------------------------|-----------------------|---------------------|------------------|-----------|--------|---------|-------|
| Original state                        | Synchronization start | Synchronization end | Exclusion reason | Succeeded | Failed | Skipped | Total |
| No record                             | Unlinked              | Linked              |                  | 39        | 0      | 0       | 39    |
| No record                             | Unmatched             | Unmatched           |                  | 5         | 0      | 0       | 5     |
| No record                             | No record             | No record           | Protected        | 0         | 0      | 1       | 1     |

# Unmatched & Protected Accounts

- It's typical for target systems that there will be some Unmatched (orphaned) accounts
- Some accounts should be kept "as is" by configuring as "Protected"
  - In this case, certain account(s) configuration of "Protected" is in the resource template itself

# Reconciliation Simulation Results

- After simulations, various information is reported
- AD accounts have not been touched yet, we were only simulating!**
- Click “More info” to list the accounts
- We need to understand what midPoint would do – and why

The screenshot shows a "Simulation result" page with the following details:

| Simulation task details |   | Event marks                 |   |  |   |
|-------------------------|---|-----------------------------|---|--|---|
| Start timestamp         | September 6, 2023 at 11:05:46 AM UTC            | Focus activated             | 0 | Focus deactivated                      | 0 |
| End timestamp           | September 6, 2023 at 11:06:00 AM UTC            | More info                   |   | More info                              |   |
| Finished in             | 13 seconds                                      | Focus archetype changed     | 0 | Focus parent organization reference... | 0 |
| Task                    | Reconciliation with AD - development simulation | More info                   |   | Focus role membership changed          | 0 |
| Status                  | Finished  | Projection deactivated      | 5 | Projection renamed                     | 0 |
| Configuration           | Development                                     | More info                   |   | Projection identifier changed          | 0 |
| Added objects           | 0   | More info                   |   | Projection entitlement changed         | 0 |
| Deleted objects         | 0   | More info                   |   | More info                              |   |
| Modified objects        | 78  | More info                   |   | More info                              |   |
| Unmodified objects      | 39  | Projection password changed | 0 | Resource object affected               | 5 |
| All processed objects   | 122   | More info                   |   | More info                              |   |

# Reconciliation Simulation Results (2)

- Modified users indicate a single modification
- Link to account is being added to owning user objects for **successfully correlated accounts**

| Processed objects |                      |      |          |                        |
|-------------------|----------------------|------|----------|------------------------|
|                   | Name                 | Type | State    | Changes                |
| □                 | 1001 (Geena Green)   | User | Modified | 1 Additions of total 1 |
| □                 | 1003 (Jimmy Taylor)  | User | Modified | 1 Additions of total 1 |
| □                 | 1004 (Peter Hunter)  | User | Modified | 1 Additions of total 1 |
| □                 | 1005 (Emanuel Young) | User | Modified | 1 Additions of total 1 |

| Changes                                       |                |   |
|---|----------------|---|
| Item  | Old value      | New value   |
| Modify User Geena Green (1001)<br>Projections | 1001 [Default] | cn=Geena Green,ou=users,dc=example,dc=com [Default] |

# Reconciliation Simulation Results (3)

- Affected accounts to be deleted indicate a potential problem
- Notice the protected account is not affected (won't be deleted / modified) – only 5, not 6 accounts displayed

|                          | Name  | Type   | State   | Changes    |
|--------------------------|---|--------|---------|------------|
| <input type="checkbox"/> | cn=Ana Lopez,ou=users,dc=example,dc=com (Account cn=Ana Lopez,ou=users,dc=example,dc=com (default) on AD)<br>Projection deactivated, Resource object affected   | Shadow | Deleted | No changes |
| <input type="checkbox"/> | cn=Mail Service Account,ou=users,dc=example,dc=com (Account cn=Mail Service Account,ou=users,dc=example,dc=com (default) on AD)<br>Projection deactivated, Resource object affected                   | Shadow | Deleted | No changes |
| <input type="checkbox"/> | cn=Secret Admin,ou=users,dc=example,dc=com (Account cn=Secret Admin,ou=users,dc=example,dc=com (default) on AD)<br>Projection deactivated, Resource object affected                                   | Shadow | Deleted | No changes |
| <input type="checkbox"/> | cn=Spam Assassin Service Account,ou=users,dc=example,dc=com (Account cn=Spam Assassin Service Account,ou=users,dc=example,dc=com (default) on AD)<br>Projection deactivated, Resource object affected | Shadow | Deleted | No changes |
| <input type="checkbox"/> | cn=Test123,ou=users,dc=example,dc=com (Account cn=Test123,ou=users,dc=example,dc=com (default) on AD)<br>Projection deactivated, Resource object affected   | Shadow | Deleted | No changes |

# Why Accounts Are Indicated To Be Deleted?

- Correlation and synchronization configuration are related
  - Unmatched → Delete resource object** will cause the delete operation (we don't want orphaned accounts! It is just doing too much now)

| List of reactions                              |   |                         |                                  |  |                      |
|--|---|-------------------------|----------------------------------|--|----------------------|
| <input type="checkbox"/> Name <small>i</small> | Situation <small>*</small> <small>i</small> | Action <small>i</small> | Lifecycle state <small>i</small> |  | <small>trash</small> |
| <input type="checkbox"/>                       | Unlinked                                    | Link                    | Proposed                         | <small>edit</small> <small>trash</small> |                      |
| <input type="checkbox"/>                       | Linked                                      | Synchronize             | Proposed                         | <small>edit</small> <small>trash</small> |                      |
| <input type="checkbox"/>                       | Deleted                                     | Unlink                  | Proposed                         | <small>edit</small> <small>trash</small> |                      |
| <input type="checkbox"/>                       | Unmatched                                   | Delete resource object  | Proposed                         | <small>edit</small> <small>trash</small> |                      |

# Displaying Unmatched Accounts

- Accounts menu item displays all AD accounts midPoint is aware of
- The list can be filtered e.g. by “Situation”

| Accounts                 |   |   |           |   |
|--------------------------|---|---|-----------|---|
|                          | Name  | Situation   | Owner     | Pending operations  |
| <input type="checkbox"/> |  cn=Ana Lopez,ou=users,dc=example,dc=com                     | entryUUID: b398e57a-e003-103d-8162-5d0d35036f2f<br>uid: alopez<br>dn: cn=ana lopez,ou=users,dc=example,dc=com                       | UNMATCHED | <input data-bbox="1985 438 2081 454" type="button" value="Import"/> <input data-bbox="1985 454 2081 483" type="button" value="Sync"/> |
| <input type="checkbox"/> |  cn=Secret Admin,ou=users,dc=example,dc=com                  | entryUUID: b3b9fb20-e003-103d-8189-5d0d35036f2f<br>uid: hacker<br>dn: cn=secret admin,ou=users,dc=example,dc=com                    | UNMATCHED | <input data-bbox="1985 533 2081 550" type="button" value="Import"/> <input data-bbox="1985 550 2081 578" type="button" value="Sync"/> |
| <input type="checkbox"/> |  cn=Mail Service Account,ou=users,dc=example,dc=com          | entryUUID: b3cb09b0-e003-103d-818b-5d0d35036f2f<br>uid: mail-svc<br>dn: cn=mail service account,ou=users,dc=example,dc=com          | UNMATCHED | <input data-bbox="1985 628 2081 645" type="button" value="Import"/> <input data-bbox="1985 645 2081 674" type="button" value="Sync"/> |
| <input type="checkbox"/> |  cn=Test123,ou=users,dc=example,dc=com                       | entryUUID: b3bbf632-e003-103d-818a-5d0d35036f2f<br>uid: test123<br>dn: cn=test123,ou=users,dc=example,dc=com                        | UNMATCHED | <input data-bbox="1985 724 2081 740" type="button" value="Import"/> <input data-bbox="1985 740 2081 769" type="button" value="Sync"/> |
| <input type="checkbox"/> |  cn=Spam Assassin Service Account,ou=users,dc=example,dc=com | entryUUID: b3cbad02-e003-103d-818c-5d0d35036f2f<br>uid: spam-svc<br>dn: cn=spam assassin service account,ou=users,dc=example,dc=com | UNMATCHED | <input data-bbox="1985 800 2081 817" type="button" value="Import"/> <input data-bbox="1985 817 2081 845" type="button" value="Sync"/> |

# Reviewing Unmatched Accounts

- Unmatched accounts are considered orphaned because of many reasons
  - Really orphaned accounts (e.g. former employees, backdoors)
  - Technical/service/administrator accounts
  - Bad correlation criteria (e.g. if all accounts would be Unmatched)
  - Uncorrelable accounts (e.g. invalid account data, invalid correlator etc.)
- We can define **exceptions** for Unmatched accounts (e.g. *mark* them as Protected or not correlable at this time)

# Displaying Protected Accounts

- Protected accounts are displayed too

Accounts Normal Account ▾ Proposed ▾ Configure ▾ Tasks ▾  Show statistics

|                          | Name <span>i</span>  | Situation <span>i</span>  | Owner | Pending operations                        |
|--------------------------|--|---|-------|---|
| <input type="checkbox"/> |  |   |       | <span>Download</span> <span>Import</span> |
| <input type="checkbox"/> |  cn=WWW Service<br>Account.ou=users,dc=example,dc=com | entryUUID: b3ccdd26-e003-103d-818d-5d0d35036f2f<br>uid: www-svc<br>dn: cn=www service<br>account,ou=users,dc=example,dc=com |       | <span>Download</span> <span>Import</span> |

# Marking Accounts

- You can make exception from default policy using marks (Protected or other)
- Available from simulation results or account list
- Information is stored in midPoint's account representation
  - Shadow object

The screenshot shows the midPoint interface with the following components:

- Processed objects** panel: Shows two accounts being processed:
  - cn=Ana Lopez,ou=users,dc=example,dc=com** (Account cn=Ana Lopez,ou=users,dc=example,dc=com (default) on AD)  
Projection deactivated, Resource object affected
  - cn=Mail Service Account,ou=users,dc=example,dc=com** (Account cn=Mail Service Account,ou=users,dc=example,dc=com (default) on AD)  
Projection deactivated, Resource object affected
- Select Marks** dialog: A modal window listing event marks:
  - Correlate later (checked)
  - Decommission later
  - Do not touch
  - Invalid data
  - Protected
- Buttons**: Add, Cancel, Shadow, Deleted, No changes.

# Marking Accounts (2)

- **Protected:** ignored during both synchronization and provisioning
  - Usually used for administrative/technical accounts
- **Do not touch:** the same as Protected, but reason may be different
  - Usually used for accounts where we should not make any changes unless we have more information
- **Invalid data:** the same as Protected, but reason may be different
  - Usually used for source system accounts with obvious errors
  - We can stop synchronizing the accounts and **override their data**

# Marking Accounts (3)

- **Decommission later:** not updated automatically by synchronization, but may be deleted manually later
  - Usually used for accounts that will be eventually deleted
- **Correlate later:** ignored during correlation and synchronization
  - Useful if the account cannot be correlated but we want to continue with the others
- Docs: [Object Marks](#)

# Displaying Marked Accounts

Accounts Normal Account Proposed Configure Tasks Show statistics

|                          | Identifiers   | Situation | Owner | Pending operations |
|--------------------------|---|-----------|-------|--------------------|
| <input type="checkbox"/> | entryUUID: b3cb09b0-e003-103d-818b-5d0d35036f2f<br>uid: mail-svc<br>dn: cn=mail service<br>account,ou=users,dc=example,dc=com | UNMATCHED |       |                    |
| <input type="checkbox"/> | cn=Mail Service<br>Account,ou=users,dc=example,dc=com<br>   | UNMATCHED |       |                    |
| <input type="checkbox"/> | cn=Ana Lopez,ou=users,dc=example,dc=com<br>   | UNMATCHED |       |                    |
| <input type="checkbox"/> | cn=Secret Admin,ou=users,dc=example,dc=com  | UNMATCHED |       |                    |
| <input type="checkbox"/> | cn=Test123,ou=users,dc=example,dc=com<br>   | UNMATCHED |       |                    |
| <input type="checkbox"/> | cn=Spam Assassin Service<br>Account,ou=users,dc=example,dc=com<br>  | UNMATCHED |       |                    |

# Running Simulated Reconciliation Again

- After marking accounts, the simulation results should be improved

[← Back](#) Simulation result: Reconciliation with AD - development simulation

| Simulation task details |   |
|-------------------------|---|
| Start timestamp         | September 6, 2023 at 1:48:56 PM UTC             |
| End timestamp           | September 6, 2023 at 1:49:00 PM UTC             |
| Finished in             | 4 seconds                                       |
| Task                    | Reconciliation with AD - development simulation |
| Status                  | Finished  |
| Configuration           | Development                                     |
| Added objects           | 0   |
| Deleted objects         | 1   |

▼ Event marks

|                         |   |
|-------------------------|---|
| Focus activated         | 0 |
| More info →             |   |
| Focus archetype changed | 0 |
| More info →             |   |
| Projection deactivated  | 1 |
| More info →             |   |

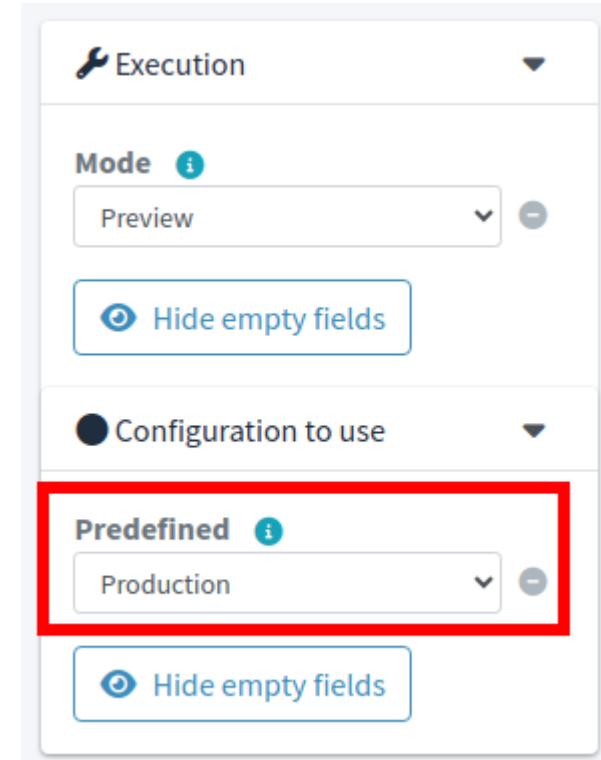
# Reconciliation Simulation – Ignoring Unmatched

- To move forward, we can ignore the Unmatched accounts – for now
- Set all other synchronization actions as Active
- Switch resource to Active
- Switch object type to Active

| Situation * <small>i</small> | Action <small>i</small> | Lifecycle state <small>i</small> |  |
|------------------------------|-------------------------|----------------------------------|--|
| Unlinked                     | Link                    | Active                           | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| Linked                       | Synchronize             | Active                           | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| Deleted                      | Unlink                  | Active                           | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| Unmatched                    | Delete resource object  | Proposed                         | <input checked="" type="checkbox"/> <input type="checkbox"/> |

# Reconciliation Simulation – Production Configuration

- Create new reconciliation task with simulation
  - Name: “Reconciliation with AD – production simulation”
- Production = configuration items in *Active / Deprecated* lifecycle state
- Configuration items with *Proposed* lifecycle state are ignored



# Reconciliation Simulation – Production Configuration (2)

- Simulation results show no deleted objects!

|                    |    |
|--------------------|----|
| Added objects      | 0  |
| Deleted objects    | 0  |
| Modified objects   | 78 |
| Unmodified objects | 39 |

# Reconciliation Simulation – Production Configuration (3)

- Modified users still indicate a single modification
- Link to account is being added to owning user objects for **successfully correlated accounts**

|   | Name                 | Type | State    | Changes                |  |  |
|---|----------------------|------|----------|------------------------|--|--|
| □ | 1001 (Geena Green)   | User | Modified | 1 Additions of total 1 |  |  |
| □ | 1003 (Jimmy Taylor)  | User | Modified | 1 Additions of total 1 |  |  |
| □ | 1004 (Peter Hunter)  | User | Modified | 1 Additions of total 1 |  |  |
| □ | 1005 (Emanuel Young) | User | Modified | 1 Additions of total 1 |  |  |

| Changes     |                  |   |
|-------------|------------------|---|
| Item        | Old value        | New value   |
| Projections | ● 1001 [Default] | cn=Geena Green,ou=users,dc=example,dc=com [Default] |

# Understanding Correlation Results – Again

- Synchronization situations transitions are a bit different again
  - 39 accounts are (would be) linked (because we're just simulating) during the reconciliation task execution
  - 1 account is Unmatched (orphaned) – this is OK, we will eventually delete it, just not yet
  - 4+1 accounts are reported as Protected – our marked accounts are included!

Synchronization situation transitions

| Original state | Synchronization start | Synchronization end | Exclusion reason | Succeeded | Failed | Skipped | Total |
|----------------|-----------------------|---------------------|------------------|-----------|--------|---------|-------|
| Unlinked       | Unlinked              | Linked              |                  | 39        | 0      | 0       | 39    |
| Unmatched      | No record             | No record           | Protected        | 0         | 0      | 4       | 4     |
| Unmatched      | Unmatched             | Unmatched           |                  | 1         | 0      | 0       | 1     |
| No record      | No record             | No record           | Protected        | 0         | 0      | 1       | 1     |

# Real Correlation With AD

- We have simulated the reconciliation which will correlate our AD accounts with midPoint
- We have marked some accounts as exceptions to ignore them
- We know what would happen – just make it happen now!
- Create non-simulated reconciliation task with AD
- Vast majority of users have links to their AD accounts (correlation was done)

# Real Correlation With AD (2)

- One of the users has not been successfully correlated
  - Account was Unmatched and marked as “Do not correlate” – for now

| Persons                  |      | Users > Persons |              | US                 | Power  |   |  |   |  |   |  |   |  |
|--------------------------|------|-----------------|--------------|--------------------|--|---|--|---|--|---|--|---|--|
| Full name <i>i</i>       |      | Name <i>i</i>   |              | Users with account |  |   |  |   |  |   |  |   |  |
|                          |      |                 |              | ResourceType       |  |   |  |   |  |   |  |   |  |
| <input type="checkbox"/> | Name | Personal Number | Full name    | Email              | Accounts   |   |  |   |  |   |  |   |  |
| <input type="checkbox"/> | 1001 | 1001            | Geena Green  |                    | <table border="1"><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>2</td><td></td></tr></table> | 2 |  | 1 |  | 2 |  | 2 |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 1                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| <input type="checkbox"/> | 1002 | 1002            | Ana Lopez    |                    | <table border="1"><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>2</td><td></td></tr></table> | 2 |  | 1 |  | 2 |  | 2 |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 1                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| <input type="checkbox"/> | 1003 | 1003            | Jimmy Taylor |                    | <table border="1"><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>2</td><td></td></tr></table> | 2 |  | 1 |  | 2 |  | 2 |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 1                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| <input type="checkbox"/> | 1004 | 1004            | Peter Hunter |                    | <table border="1"><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>2</td><td></td></tr></table> | 2 |  | 1 |  | 2 |  | 2 |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 1                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |
| 2                        |      |                 |              |                    |  |   |  |   |  |   |  |   |  |

# Module 5: Labs

LAB 5-1: Simulated Correlation With Active Directory

# Module 5: Labs

LAB 5-2: Marking Accounts

# Module 5: Labs

LAB 5-3: Ignoring Orphaned Accounts

# Module 5: Labs

LAB 5-4: Real Correlation With Active Directory

# Module 5: Self-assessment

- TODO

# Module 5: Summary

- Reconciliation
- Protected accounts and other account marks

# Module 5

End of module

# Module 6

Importing Usernames From Target System

# Importing Usernames From AD

- MidPoint usernames are created from HR's `empnum`
- To import usernames from AD, we need to
  - Update HR inbound mapping for `name`: we will make it *weak*
  - Create AD inbound mapping for `name`: we will make it *strong*

# Mapping Strength

- Decides when/if the value produced by the mapping will be used
- **Weak**: only if the target property is empty and there is no other change
- **Normal** (default): if there is a change of mapping sources
- **Strong**: always applied
- ⓘ Mappings created by resource wizard are by default strong

# Importing Usernames From AD (2)

- We will use HR's `empnum` value as default username
  - We will override username with AD's `uid` for users with AD accounts
  - We will improve this to generate usernames in midPoint later
- ⓘ We are simulating AD with OpenLDAP.  
You could use `sAMAccountName` for AD.

# Changing Mapping Strength

- Mapping editor – click “Edit”
- Set mapping strength
- Save mappings

| Name           | From resource attribute * | Expression | Target | Lifecycle state |   |
|----------------|---------------------------|------------|--------|-----------------|---|
| empnum-to-name | empnum                    | As is      | name   | Active          |  |

Strength   
Strong  
Undefined  
Strong  
Normal  
**Weak**  
+

# Creating New Inbound Mapping For AD Resource

- Mapping editor – click “Add inbound” (will create a strong mapping)
- Set mapping lifecycle *Proposed* – we will simulate the action first!
- Save mappings

The screenshot shows the 'Inbound mappings (to MidPoint)' tab of a mapping editor. The interface includes a dropdown for 'All mappings', a header with columns for 'Name', 'From resource attribute', 'Expression', 'Target', and 'Lifecycle state', and a toolbar with icons for creating, deleting, and editing mappings.

| Name                          | From resource attribute * | Expression | Target     | Lifecycle state |
|-------------------------------|---------------------------|------------|------------|-----------------|
| mapping-inbound-familyName-fo | sn                        | As is      | familyName | Active          |
| mapping-inbound-username-to-n | uid                       | As is      | name       | Proposed        |

# Run Import Preview For Username Import

- We will use import preview for one account with **Simulated development** option as the mapping is in *Proposed* lifecycle state

The screenshot shows the SAP Fiori interface for managing user accounts. On the left, there is a list of users with IDs 0001 and 0003. A context menu is open over user 0001, with options: Enable, Disable, Delete, Import preview (which is highlighted with a red box), and Import. To the right, a modal dialog titled "Select task execution mode" is displayed, showing a dropdown menu with "Simulated development" selected, also highlighted with a red box. At the bottom, there is a summary row for user "geena (Geena Green)" showing status "User", modification level "Modified", and "1 Edits of total 1". The name "geena (Geena Green)" is also highlighted with a red box.

# Run Simulated Reconciliation For Username Import

- Run “Reconciliation with AD – development simulation” task again

Simulation result: Reconciliation with AD - development simulation, 2023-09-07T08:35:04.032Z

[View processed objects](#)

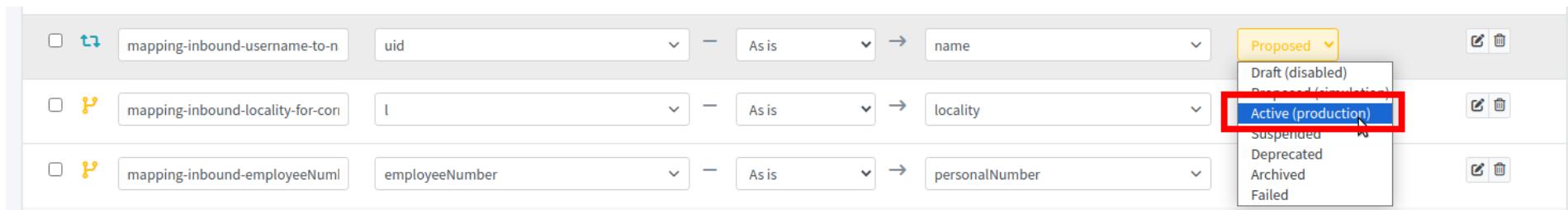
| Simulation task details |                                     | Event marks     |    |                           |   |
|-------------------------|-------------------------------------|-----------------|----|---------------------------|---|
| Start timestamp         | September 7, 2023 at 8:35:04 AM UTC | Focus activated | 0  | Focus deactivated         | 0 |
| End timestamp           | September 7, 2023 at 8:35:07 AM UTC | Focus renamed   | 39 | Focus assignments changed | 0 |

[More info](#)

| Name                  | Type | State    | Changes            |
|-----------------------|------|----------|--------------------|
| geena (Geena Green)   | User | Modified | 1 Edits of total 1 |
| jaylor (Jimmy Taylor) | User | Modified | 1 Edits of total 1 |
| hunter (Peter Hunter) | User | Modified | 1 Edits of total 1 |

# Run Reconciliation To Import Usernames

- Switch the AD inbound mapping for username to Active lifecycle state
- Run reconciliation task (without simulation)



# Username Import – Result

- All users with linked AD accounts have been renamed using their `uid` attribute

|                          | Name     | Personal Number | Full name         | Email | Accounts |   |
|--------------------------|----------|-----------------|-------------------|-------|----------|---|
| <input type="checkbox"/> | 1002     | 1002            | Ana Lopez         |       | 1        |       |
| <input type="checkbox"/> | abaker   | 1021            | Alice Baker       |       | 2        |       |
| <input type="checkbox"/> | adewries | 1030            | Amanda de Wries   |       | 2        |       |
| <input type="checkbox"/> | afreeman | 1010            | Alexander Freeman |       | 2        |       |
| <input type="checkbox"/> | ajackson | 1029            | Ashley Jackson    |       | 2        |   |

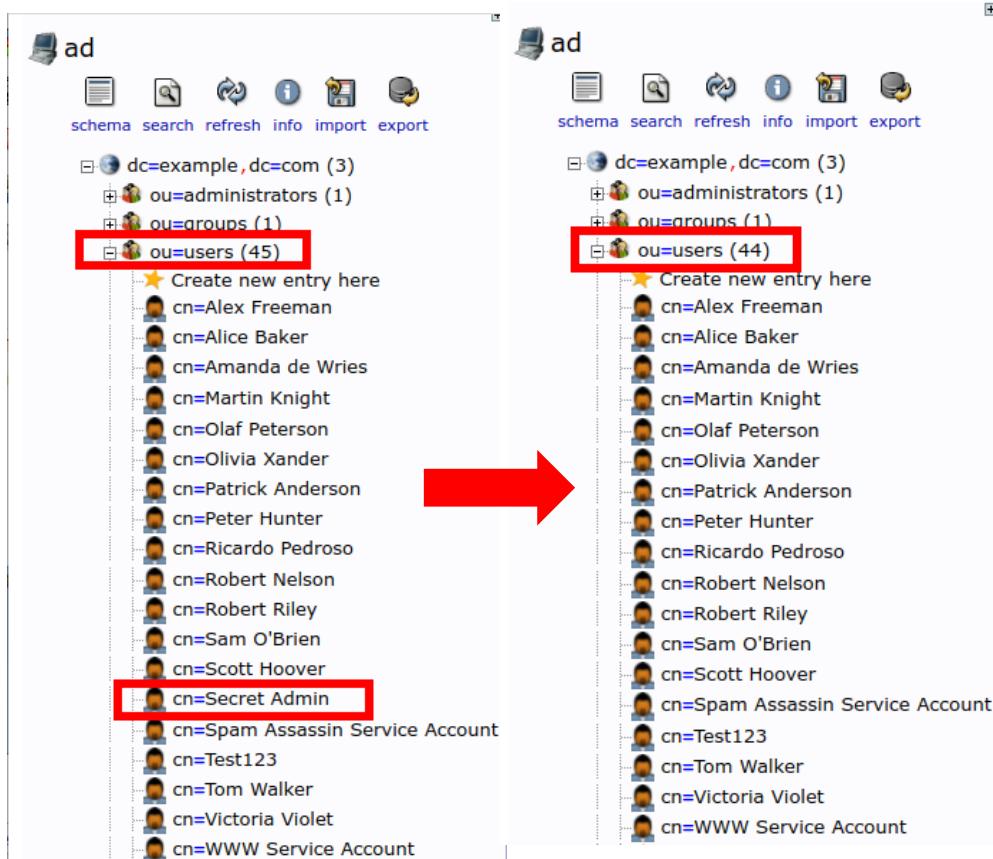
# Deleting Orphaned AD Accounts

- Now or later, you can get rid of the unmatched AD accounts which have not been marked
- Set the action **Unmatched → Delete resource object** to Active
- Run reconciliation
- MidPoint is authoritative for deleting orphaned accounts from now on**

| Situation <small>?</small> | Action <small>?</small> | Lifecycle state <small>?</small> |  |
|----------------------------|-------------------------|----------------------------------|--|
| Unlinked                   | Link                    | Active                           | <input type="checkbox"/> <input type="button" value="Delete"/> |
| Linked                     | Synchronize             | Active                           | <input type="checkbox"/> <input type="button" value="Delete"/> |
| Deleted                    | Unlink                  | Active                           | <input type="checkbox"/> <input type="button" value="Delete"/> |
| Unmatched                  | Delete resource object  | Active                           | <input type="checkbox"/> <input type="button" value="Delete"/> |

# Deleting Orphaned AD Accounts (2)

- Orphaned accounts have been deleted



# Correlate Previously Uncorrelated AD Accounts

- We have used all fully-automatic ways to correlate users, but there may be still some uncorrelated
- If data is inconsistent/missing, we may want to use smart correlation with operator intervention

Properties ↴ ⚙

|                   |               |
|-------------------|---------------|
| Name ⓘ            | 1002          |
| Lifecycle state ⓘ | Active        |
| Full name ⓘ       | Ana Lopez     |
| Given name ⓘ      | Ana           |
| Family name ⓘ     | Lopez         |
| Personal Number ⓘ | 1002          |
| Locality ⓘ        | Hot Lava City |

DN: cn=Ana Lopez,ou=users,dc=example,dc=com

| Attribute Description | Value                             |
|-----------------------|-----------------------------------|
| <i>objectClass</i>    | <i>inetOrgPerson (structural)</i> |
| <b>cn</b>             | Ana Lopez                         |
| <b>sn</b>             | Lopez                             |
| <b>displayName</b>    | Ana Lopez                         |
| <b>employeeNumber</b> | 2                                 |
| <b>givenName</b>      | Ana                               |

MIKULU-4.0-LIS-A.01, Since 2013

Show empty fields

# Correlate Previously Uncorrelated AD Accounts (2)

- Enable second correlator (predefined in the resource template)
- Remove “Do not correlate” mark from accounts you wish to correlate now

| Rule name                  | Description   | Weight | Tier | Ignore if matched by | Enabled   | Actions |
|----------------------------|---|--------|------|----------------------|-----------|---------|
| personalNumber-correlation | Correlation using personalNumber. Please note: inbound mapping for personalNumber | 1      |      |                      | Undefined |         |
| last-resort-correlation    | Correlation using givenName,  | 0.5    | 10   |                      |           |         |

# Correlate Previously Uncorrelated AD Accounts (3)

- Add new synchronization action: Disputed → Create correlation case
  - Remember: our second correlation rule compares `givenName`, `familyName` and `locality` and return weight is just  $0.5 < 1.0$ . Account will be Disputed
- Run reconciliation task

The screenshot shows the 'Accounts' page of the Evolveum reconciliation tool. At the top, there are filters for 'Normal Account' (set to 'Active'), 'Configure', 'Tasks', and a checkbox for 'Show statistics'. Below the filters is a search bar with fields for 'Name' and 'Situation' (set to 'Disputed'). A dropdown menu shows 'Basic' selected. The main table lists accounts with columns for 'Identifiers', 'Situation', 'Owner', and 'Pending operations'. One account is highlighted: 'cn=Ana Lopez,ou=users,dc=example,dc=com'. Its identifiers are listed as entryUUID: b398e57a-e003-103d-8162-5d0d35036f2f, uid: alopez, dn: cn=ana lopez,ou=users,dc=example,dc=com. The 'Situation' column for this account is highlighted with a red box and contains the word 'DISPUTED'. The bottom of the screen shows navigation buttons like 'Reload', 'Upload', 'Download', 'Import', 'Export', and 'Search', along with pagination controls for 'Rows per page' (set to 20), '1 to 1 of 1', and 'Navigation arrows'.

| Identifiers   | Situation | Owner | Pending operations |
|---|-----------|-------|--------------------|
| entryUUID: b398e57a-e003-103d-8162-5d0d35036f2f<br>uid: alopez<br>dn: cn=ana lopez,ou=users,dc=example,dc=com | DISPUTED  |       |                    |

# Correlate Previously Uncorrelated AD Accounts (4)

- Correlation case is created for Disputed accounts
- Work item is assigned to “administrator” (predefined in resource template)

| Situation * <small>i</small> | Action <small>i</small> | Lifecycle state <small>i</small> |                                     |
|------------------------------|-------------------------|----------------------------------|-------------------------------------|
| Unlinked                     | Link                    | Active                           | <input checked="" type="checkbox"/> |
| Linked                       | Synchronize             | Active                           | <input checked="" type="checkbox"/> |
| Deleted                      | Unlink                  | Active                           | <input checked="" type="checkbox"/> |
| Unmatched                    | Delete resource object  | Active                           | <input checked="" type="checkbox"/> |
| Disputed                     | Create correlation case | Active                           | <input checked="" type="checkbox"/> |

administrator

Cases 1

All work items

My work items

| Name   | Stage | State | Object                                  | Target | Actor(s)                               | Created             | Process started     | Deadline | Esc. level |
|--|-------|-------|---|--------|--|---------------------|---------------------|----------|------------|
| Correlation of account 'cn=Ana Lopez,ou=users,dc=example,dc=com' on AD | open  | AD    | cn=Ana Lopez,ou=users,dc=example,dc=com |        | midPoint Administrator (administrator) | 9/7/23, 11:09:39 AM | 9/7/23, 11:09:39 AM |          |            |

# Correlate Previously Uncorrelated AD Accounts (5)

- Correlation candidates are presented as matched by the correlation rules
- Matching and non-matching items are displayed
- Administrator confirms the candidate by clicking a button
- Account is linked to midPoint user

The screenshot shows a midPoint interface for correlating an account from AD. The top header indicates the correlation is for account 'cn=Ana Lopez,ou=users,dc=example,dc=com' on AD, requested by midPoint Administrator (administrator) on September 7, 2023, at 11:09:39 AM. The main area displays correlation details and a list of candidates:

| Action           | Object being correlated | Correlation candidate 1    |
|------------------|-------------------------|----------------------------|
| Create new       |                         | <button>Correlate</button> |
| Name             |                         | Ana Lopez (1002)           |
| Match confidence |                         | 50% ⓘ                      |
| Given name       | Ana                     | Ana                        |
| Locality         | Hot Lava City           | Hot Lava City              |
| Family name      | Lopez                   | Lopez                      |
| Personal Number  | 2                       | 1002                       |

At the bottom, there are 'Back' and 'Forward' navigation buttons.

# Correlate Previously Uncorrelated AD Accounts (6)

- Correlation is finished
- Account is linked to its owner
- AD username was imported too (remember: we were actually running reconciliation)

| <input type="checkbox"/> | Name   | Personal Number | Full name   | Email | Accounts | <input type="button" value=""/> |
|--------------------------|--------|-----------------|-------------|-------|----------|---------------------------------|
| <input type="checkbox"/> | abaker | 1021            | Alice Baker |       | 2        | <input type="button" value=""/> |
| <input type="checkbox"/> | alopez | 1002            | Ana Lopez   |       | 2        | <input type="button" value=""/> |

# Module 6: Labs

LAB 6-1: Preparing Configuration For Username  
Import

# Module 6: Labs

LAB 6-2: Username Import Simulation

# Module 6: Labs

LAB 6-3: Username Import From Active Directory

# Module 6: Labs

LAB 6-4: Deleting Orphaned Active Directory Accounts

# Module 6: Labs

LAB 6-5: Finalize Correlation

# Module 6: Self-assessment

- TODO

# Module 6: Summary

- Mapping strength
- Importing existing usernames from AD
- Deleting orphaned AD accounts
- Finalizing correlation of previously uncorrelated AD accounts

# Module 6

End of module

# Module 7

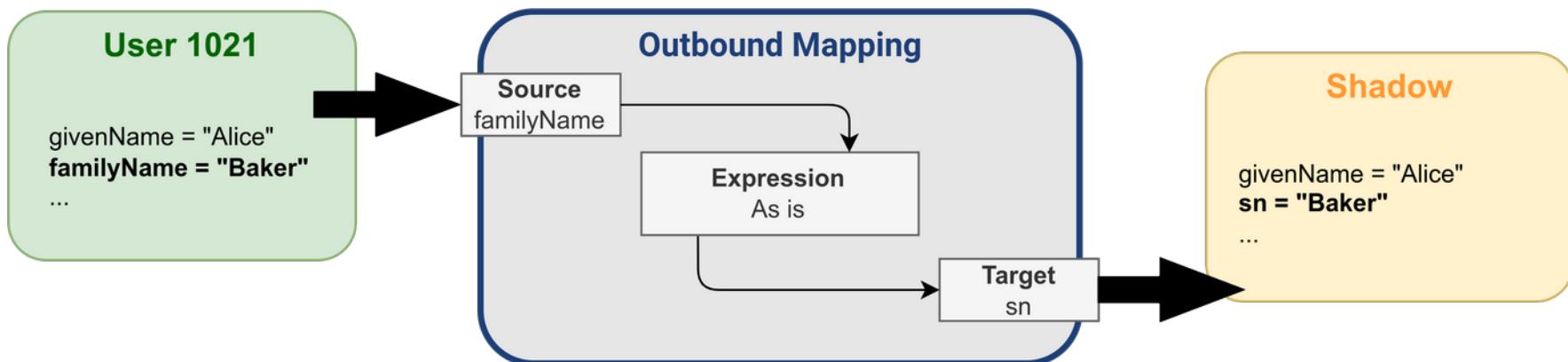
Enabling Provisioning to Target System

# Enable Provisioning To AD

- We have correlated existing AD accounts to their midPoint owners (from HR)
- It's time to configure provisioning to AD from midPoint
- Outbound mappings
- Outbound activation mappings
- Outbound password mappings
- Simulation before going to production

# Object Type: Outbound Mappings

- Data flow direction: midPoint to resource
- How source **resource attribute** value is set



# Object Type: Outbound Mappings (2)

- Click to attribute field/start typing (auto-completion of attribute/property name)
- All outbound mappings are in *Draft* lifecycle state (predefined in resource template)
- Use only mappings you need

The screenshot shows the configuration of five outbound mappings:

- mapping-dn:** Source: name, fullName; Expression: Script (Evaluator value is set); To resource attribute: dn.
- mapping-cn-weak:** Source: name, fullName; Expression: Script (Evaluator value is set); To resource attribute: cn.
- mapping-displayName:** Source: fullName; Expression: As is; To resource attribute: displayName.
- mapping-sn:** Source: familyName; Expression: As is; To resource attribute: sn.
- mapping-givenName:** Source: givenName; Expression: As is; To resource attribute: givenName.

Each mapping has a Lifecycle state of Draft. The "Outbound mappings (to Resource)" section is highlighted with a red box.

# Activation Concept

- Status of user, account, other objects or assignments (later)
- Based on several properties (Administrative status, Valid From, Valid To, Lifecycle status)
- *Effective status* is computed from these properties and passed to outbound activation mapping(s) for Administrative status
- Docs: [Activation](#)

User's effective status displayed in Summary panel



# Object Type: Activation Mappings (Outbound)

- Activation is the mechanism to enable/disable user or resource account
- The “Administrative status” mapping enables/disabled accounts based on midPoint user status
  - Predefined in resource template

**Activation configuration**

The term activation is used in midPoint to denote a set of properties that describe whether the user is active, therefore please create and configure desired scenarios for activation

The screenshot shows the 'Activation configuration' screen in midPoint. It displays three outbound activation mappings:

- set-account-status-based-on-midpoint-user**: This mapping is highlighted with a red box. It is described as "Mapping is strong and use As is evaluator". It has 'Settings' and 'Remove' buttons.
- Disable instead of delete**: Instead of deleting the user, it changes its state to "Disabled". It has 'Settings' and 'Remove' buttons.
- Delayed delete**: Definitively deletes the user within the specified timeframe. It has 'Settings' and 'Remove' buttons.

At the bottom left is a blue 'Add outbound' button. At the bottom right are navigation icons: '<<', '<', '1', '>', and '>>'.

# Object Type: Activation Mappings (Outbound) (2)

- Activation allows much more activation rules

- Disable instead of delete (predefined in resource template)
- Delayed delete (predefined in resource template)
- Pre-provision
- Valid From / Valid To setting
- ...

## Activation rules

Select an activation rule type you would like to create/add for your automatization.

|  |
|--|
|  <b>Administrative status</b><br>Defines the "administrative state" of the object (user)                            |
|  <b>Disable instead of delete</b><br>Instead of deleting the user changes its state to "Disabled"                   |
|  <b>Delayed delete</b><br>Definitively deletes the user within the specified timeframe                              |
|  <b>Pre provision</b><br>Pre-creates an account and sets up bindings at a specified time before official activation |
|  <b>Valid from</b><br>Definition of the circumstances for start of the object's validity                            |
|  <b>Valid to</b><br>Definition of the circumstances for end of the object's validity                                |
|  <b>Lockout status</b><br>Defines whether is account locked   |
|  <b>Existence</b><br>Determines whether the resource object should exist or not                                   |

# Object Type: Activation Mappings (Outbound) (3)

- **Disable instead of delete**

- By default, if assignments are used (later), the account may be removed if unassigned or in *Archived* lifecycle state
- This configuration will keep account disabled instead

- **Delayed delete**

- Combined with Disable instead of delete, you may configure the automatic deletion e.g. 6 months after user leaves the organization

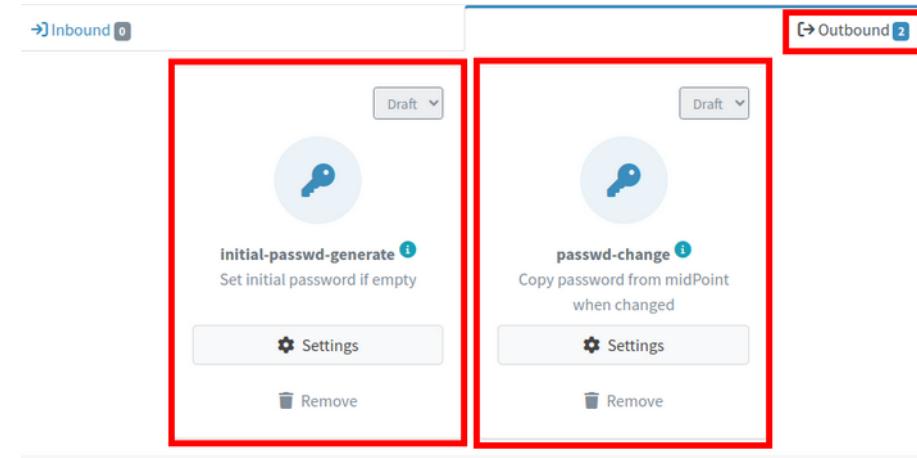
The screenshot shows the midPoint Activation configuration interface. At the top, it says "Activation configuration" and provides a brief description: "The term activation is used in midPoint to denote a set of properties that describe whether the user is active, therefore please create and configure desired scenarios for activation". Below this, there are two main sections: "Inbound" and "Outbound". The "Outbound" section is highlighted with a red box. It contains three mapping cards:

- set-account-status-based-on-midpoint-user**: A mapping card for "user" type. It shows a brief description: "Instead of deleting the user changes its state to 'Disabled'". It has "Settings" and "Remove" buttons.
- Disable instead of delete**: A mapping card for "user" type. It shows a brief description: "Instead of deleting the user changes its state to 'Disabled'". It has "Settings" and "Remove" buttons.
- Delayed delete**: A mapping card for "user" type. It shows a brief description: "Definitively deletes the user within the specified timeframe". It has "Settings" and "Remove" buttons.

At the bottom of the "Outbound" section, there is a blue "Add outbound" button and a navigation bar with page numbers (1, 2, 3, etc.).

# Object Type: Credentials Mappings (Outbound)

- MidPoint can populate account's password either by using midPoint user password or generated one
- We don't want to overwrite existing AD passwords
- Note: we do not have passwords in midPoint!



# Credentials Additional Notes

- Password could be imported from source system
- Password could be generated when importing data from source system
- Credentials inbound mappings (“As is” or “Generate”) would be used

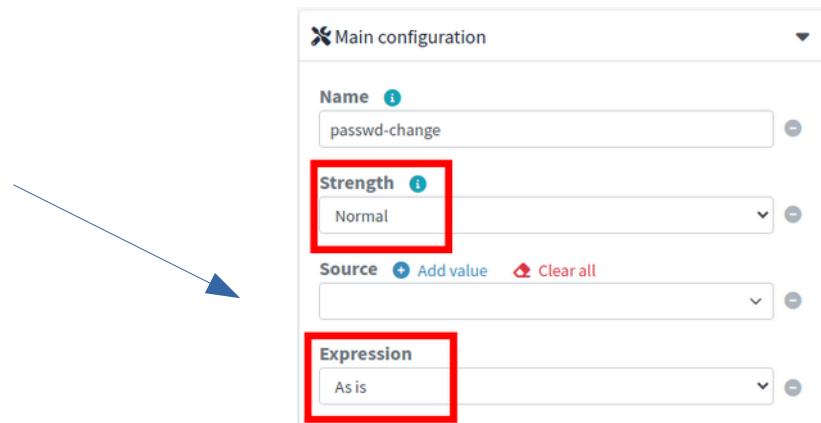
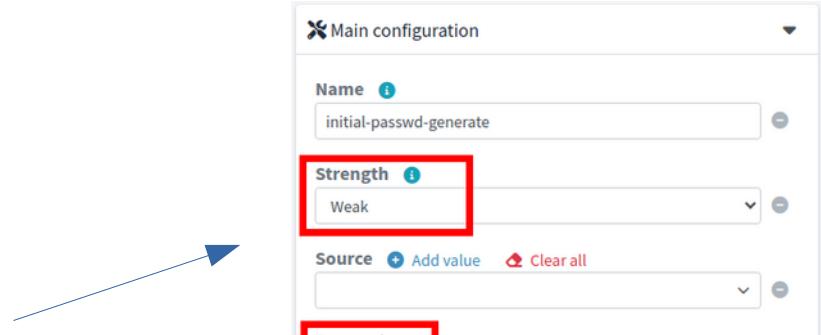
# MidPoint Password Options

- We will not generate any midPoint passwords
- We can't import AD account passwords
- We can generate random passwords for **new** AD accounts
- We assume new users will reset their initial AD passwords using helpdesk
- We could configure midPoint to authenticate against AD
- We can allow password change in midPoint to provision it to AD

Out  
of  
scope

# How To Avoid AD Password Overwrite

- Do not use mapping strength: strong!
- Initial password will be generated using **strength=weak**
  - Default password policy from midPoint will be used
- Password can be changed from midPoint using **strength=normal**
  - Source of this “As is” mapping is by default user's password



# Running Simulated Reconciliation Again

- Running simulated reconciliation with development configuration will show us how good are our **outbound** mappings... and target system data
  - Out outbound mappings are in *Proposed* lifecycle state
- Expect problems here

| Simulation result: Reconciliation with AD - development simulation, 2023-09-07T12:52:43.032Z |   |                             |   |
|--|---|-----------------------------|---|
| Simulation task details  |   | Event marks                 |   |
| Start timestamp  | September 7, 2023 at 12:52:43 PM UTC            | Focus activated             | Focus deactivated                       |
| End timestamp  | September 7, 2023 at 12:52:49 PM UTC            | 0                           | 0                                       |
| Finished in  | 6 seconds                                       | More info ↗                 | More info ↗                             |
| Task   | Reconciliation with AD - development simulation | Focus archetype changed     | Focus parent organization reference ... |
| Status   | Finished  | 0                           | 0                                       |
| Configuration  | Development                                     | More info ↗                 | More info ↗                             |
| Added objects  | 0   | Projection deactivated      | Projection renamed                      |
| Deleted objects  | 0   | 2                           | 5                                       |
| Modified objects   | 20  | More info ↗                 | More info ↗                             |
| Unmodified objects   | 100   | Projection password changed | Projection identifier changed           |
| All processed objects  | 120   | 0                           | 5                                       |
|  |   | More info ↗                 | More info ↗                             |
|  |   | Resource object affected    | More info ↗                             |
|  |   | 10                          |   |
|  |   | More info ↗                 |   |

# Running Simulated Reconciliation Again (2)

- Typical issues found in this stage:
  - Outbound mappings are incorrect
  - Outbound mappings are correct and trying to fix incorrect existing account data
- **Review the simulation results and**
  - Run real reconciliation to allow midPoint to change the accounts
  - Fix outbound mappings
  - Mark AD accounts

# Typical Issues With Outbound Mappings

- Typos in AD data; outbound mapping is going to fix data
- MidPoint has more authoritative data from HR

| Modify attributes  |  |   |
|--------------------|--|---|
| Item               | Old value                                    | New value   |
| Distinguished Name | - cn=Alex Freeman,ou=users,dc=example,dc=com | + cn=Alexander Freeman,ou=users,dc=example,dc=com |
| Given Name         | - Alex                                       | + Alexander                                       |
| Display Name       | - Alex Freeman                               | + Alexander Freeman                               |

# Typical Issues With Outbound Mappings (2)

- Incorrectly enabled AD accounts; outbound mapping is going to fix data
- MidPoint has more authoritative data from HR

The screenshot shows the MidPoint Changes interface. At the top, there are tabs for 'Changes' (selected), 'Simple' (with a magnifying glass icon), and 'Advanced' (with a gear icon). Below the tabs, a message indicates an account was modified on AD. A table displays the account's properties:

| Item     | Value        |
|----------|--------------|
| Resource | AD [Default] |
| Kind     | Account      |
| Intent   | default      |

A note at the bottom states: "Shadow was disabled." There are navigation arrows at the bottom right.

# Turn On AD Provisioning From MidPoint

- Switch the AD outbound mappings, activation mappings and credentials mappings from *Proposed* to *Active* lifecycle state
- Run reconciliation task (without simulation)
- MidPoint is authoritative for AD account data from now on
- As a side effect, midPoint has fixed the incorrect **employeeNumber** for Ana Lopez's AD account

| DN: cn=Ana Lopez,ou=users,dc=example,dc=com |                                   |
|---|-----------------------------------|
| Attribute Description                       | Value                             |
| <i>objectClass</i>                          | <i>inetOrgPerson (structural)</i> |
| <i>cn</i>                                   | Ana Lopez                         |
| <i>sn</i>                                   | Lopez                             |
| <i>displayName</i>                          | Ana Lopez                         |
| <b>employeeNumber</b>                       | 1002                              |
| <i>givenName</i>                            | Ana                               |

# Turn On AD Provisioning From MidPoint (2)

- midPoint is now ready for automatic provisioning to AD based on HR data
- There is no automatic synchronization yet

# Module 7: Labs

LAB 7-1: Preparing Configuration For Username  
Import

# Module 7: Labs

LAB 7-2: Active Directory Provisioning Simulation

# Module 7: Labs

LAB 7-3: Active Directory Provisioning

# Module 7: Self-assessment

- TODO

# Module 7: Summary

- Activation concept
- Provisioning: outbound mappings, outbound activation mappings, outbound credentials mappings
- Using simulations to prevent unexpected AD changes during integration

# Module 7

End of module

# Module 8

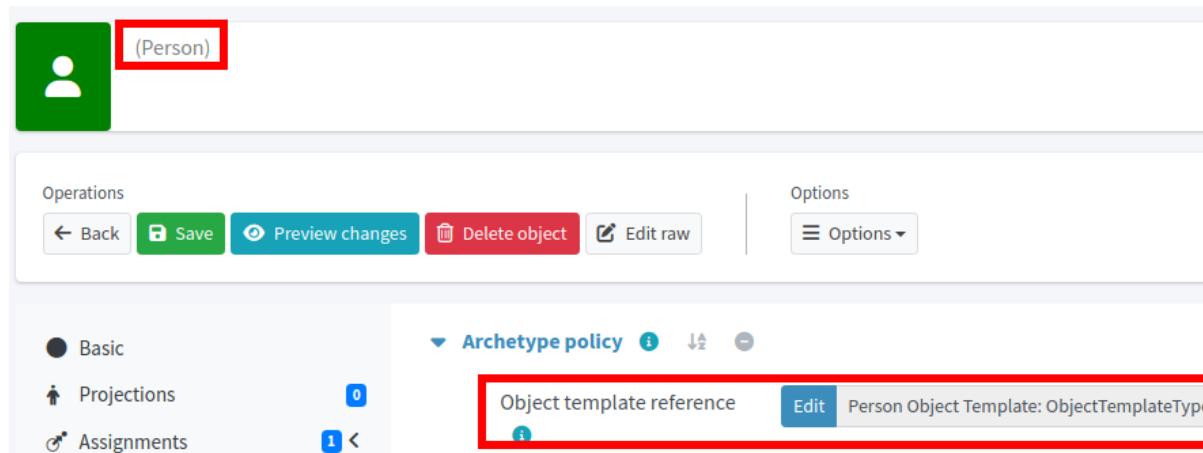
Automating Integration

# Generating Usernames In MidPoint

- MidPoint usernames can be generated using any naming convention
- To use the generator, we need to
  - Enable mapping in object template (predefined for Person archetype)
  - Remove or archive HR inbound mapping for `name`
  - Remove or archive AD inbound mapping for `name`

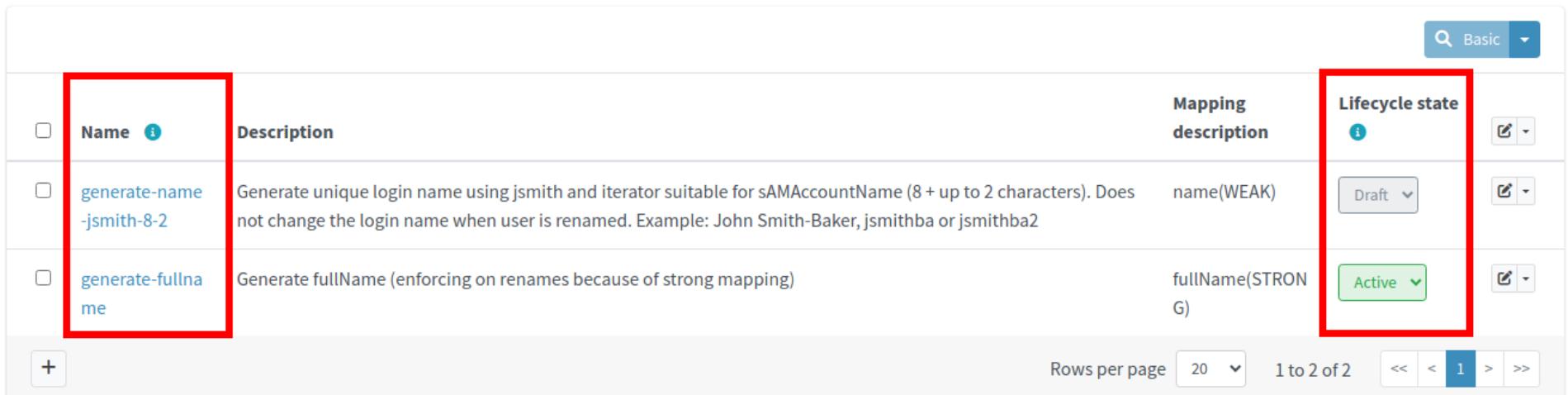
# Object Template

- Allows running mappings whenever focal object (e.g. User) is **created or updated**
- Either global or per archetype
- Person archetype: Person object template (both are built-in objects)



# Person Object Template

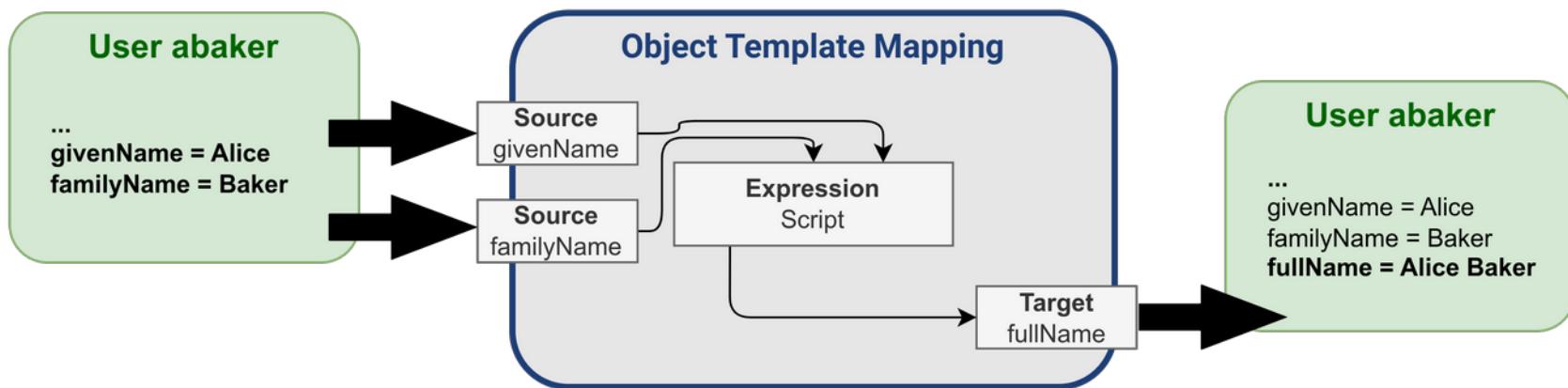
- Contains two default mappings
- **generate-fullname (Active)** to generate Person's Full name
- **generate-name-jsmith-8-2 (Draft)** to generate Person's unique username



| Name                         | Description  | Mapping description | Lifecycle state |
|------------------------------|--|---------------------|-----------------|
| generate-name<br>-jsmith-8-2 | Generate unique login name using jsmith and iterator suitable for sAMAccountName (8 + up to 2 characters). Does not change the login name when user is renamed. Example: John Smith-Baker, jsmithba or jsmithba2 | name(WEAK)          | Draft           |
| generate-fullname            | Generate fullName (enforcing on renames because of strong mapping)   | fullName(STRONG)    | Active          |

# Object Template Mappings

- Similar to inbound/outbound, but working only with focus object (User) and its properties
- Source, expression, target
- Strength (use strong to always enforce)



# Username Generator

- The unique username generator is suitable for AD's **sAMAccountName** (< 20 characters)
  - `jsmith, jsmith2, jsmith3, ... jsmith99`
  - Juan-Carlos de la Garcia → `jdelagar` (normalized, trimmed, shortened to 8)
- Iteration token: 2-99 (predefined in Person Object Template)

```
tmpGivenName = basic.trim(basic.norm(basic.stringify(givenName)))
tmpFamilyName = basic.trim(basic.norm(basic.stringify(familyName)))
tmpGivenNameInitial = tmpGivenName?.take(1)
return (tmpGivenNameInitial + tmpFamilyName?.replaceAll(" ", "
"))?.take(8) + iterationToken
```

# Username Generator: Notes

- The username is generated only when user is created (weak mapping)
- This is not a limitation of midPoint but a decision of configuration as username change can be quite complex process in company/organization
- Username can be overridden manually (administrator)

# Username Generator: Simulation (Screenshot)

Processed objects

|                          | Name   | Type   | State    | Changes   |   |
|--------------------------|--|--------|----------|---|---|
| <input type="checkbox"/> | 9000 (Account 9000 (default) on HR)                          | Shadow | Modified | <div style="width: 80%; background-color: #00AEEF;"></div><br>2 Edits of total 2        |   |
| <input type="checkbox"/> | Icallaha (Louise Callahan)<br><small>Focus activated</small> | User   | Added    | <div style="width: 100%; background-color: #2ECC71;"></div><br>10 Additions of total 10 |   |

Rows per page: 20 | 1 to 2 of 2 | << < 1 > >>

# Creating Scheduled Reconciliation Task

- Create reconciliation task as usual
- No simulation, use defaults
- Scheduling (interval in seconds or cron-like definition)
  - For real-life scenarios, you would probably use interval in hours
  - For this training, the interval will be 60 seconds

# Creating Scheduled Reconciliation Task (2)

- Usernames are unique, iterated
- No AD accounts have been provisioned for new users

| <input type="checkbox"/> | Name      | Personal Number | Full name          | Email | Accounts |  |  |  |
|--------------------------|-----------|-----------------|--------------------|-------|----------|--|--|--|
| <input type="checkbox"/> | abaker    | 1021            | Alice Baker        |       | 2        |  |  |  |
| <input type="checkbox"/> | abaker2   | 9001            | Andreas Baker      |       | 1        |  |  |  |
| <input type="checkbox"/> | cwhitehe  | 1039            | Charles Whitehead  |       | 2        |  |  |  |
| <input type="checkbox"/> | cwhitehe2 | 9002            | Clara Whiteherring |       | 1        |  |  |  |
| <input type="checkbox"/> | cwhitehe3 | 9003            | Clara Whiteherring |       | 1        |  |  |  |
| <input type="checkbox"/> | lcallaha  | 9000            | Louise Callahan    |       | 1        |  |  |  |

# Assignments and Inducements

- **Assignment is a decision**, a policy, that object (e.g. User) *should have* something (e.g. a Role)
- **Inducement is an indirect assignment**, a policy, that object assigned to e.g. User may cause something for the assignee (e.g. create a resource account)
- Users have assignments (Roles, Organizations, Services, Accounts, Archetypes)
- Roles, Organizations, Services and Archetypes have inducements that apply for the users who have them assigned and can cause e.g. resource account creation

# Assignments and Inducements (2)

- During First steps, we have no roles, organizations, services (yet), but we already have Person archetype **assigned** to our users!
- Archetypes are used to distinguish various object types and subtypes
  - E.g. employees, contractors, students etc. each with distinctive behavior
- **Archetype is a role-like object; you can consider it *birthright***
- Configuration to provision AD account for Persons is to be added to Person archetype
- Resource account construction inducement

# Assignments, Inducements and Lifecycle State

- Assignments and their inducements are evaluated for users in *Active* and *Suspended*, but not *Archived* lifecycle states
- *Suspended* users keep their AD accounts (disabled)
- *Archived* users would lose their AD accounts upon entering *Archived* lifecycle state by default
- **Disable instead of delete + delayed delete for their accounts (AD resource – from resource template)**
- Docs: [Object Lifecycle](#)

# How Delayed Delete Works

- If configured in resource, midPoint will not delete the account immediately
- Trigger to delete account is set for disableTimestamp + configured time and stored in Shadow object
- Trigger Scanner task runs each 5 minutes and processes all objects with triggers in the past
- Trigger Scanner deletes the account

## Shadow

```
name = "cn=Alice Baker, ..."  
...  
trigger:  
  timestamp = "2023-12-31T12:00:00"  
  handlerUri = "... ./trigger/recompute/handler-3  
  originDescription = "DelayedDeleteEvaluator"  
resourceRef = "AD"  
objectClass = "inetOrgPerson"  
kind = "account"  
synchronizationSituation = "linked"  
...  
attributes:  
  ri:dn = "cn=Alice Baker, ..."  
  ri:entryUUID = "b3a5c588-e003-103d-8175-  
  5d0d35036f2f"
```

# Adding AD Account Inducement for Persons, Part I.

- Edit Person archetype
- Inducements > Resource > New

The screenshot shows the 'Person' archetype editor interface. At the top right, there are checkboxes for 'Enabled' and 'Person'. Below the header, there are 'Operations' buttons: Back, Save, Preview changes, Delete object, Edit raw, and Options. On the left, a sidebar lists various sections: Basic (selected), Projections, Assignments, Activation, Password, History, Applicable Policies, Inducements (selected), All, Role, Organization, Service, and Resource (highlighted with a red box). The main area displays a table with columns: Name and Activation. A search bar at the top right says 'Basic' and shows a result count of 'No matching result found.' A red box highlights the 'Edit' icon in the table's first row.

# Adding AD Account Inducement for Persons, Part II.

- Select target system (AD)

Select application resource

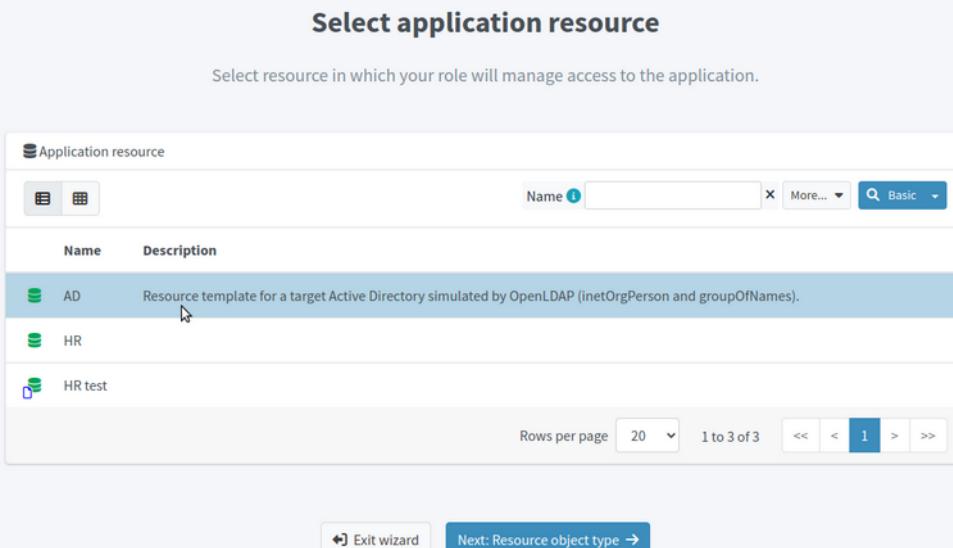
Select resource in which your role will manage access to the application.

Application resource

| Name    | Description   |
|---------|---|
| AD      | Resource template for a target Active Directory simulated by OpenLDAP (inetOrgPerson and groupOfNames). |
| HR      |   |
| HR test |   |

Rows per page: 20 | 1 to 3 of 3 | << < 1 > >>

[Exit wizard](#) [Next: Resource object type →](#)



# Adding AD Account Inducement for Persons, Part III.

- Keep default (Account)

**Select resource object type**

Select type of object your role will create or modify on the resource.



Default for kind: Account



Default for kind: Entitlement

[← Back](#) [Exit wizard](#) [Next: Entitlements →](#)

# Adding AD Account Inducement for Persons, Part IV.

- Provisioning of group membership is possible, but we're not going to use this feature now
- Keep defaults

**Grant entitlements / Group membership**

Select object(s) on the resource to be associated with the account when your role is assigned (e.g.: group, your user will be member of / entitlement, that will be assigned to the user's account...).

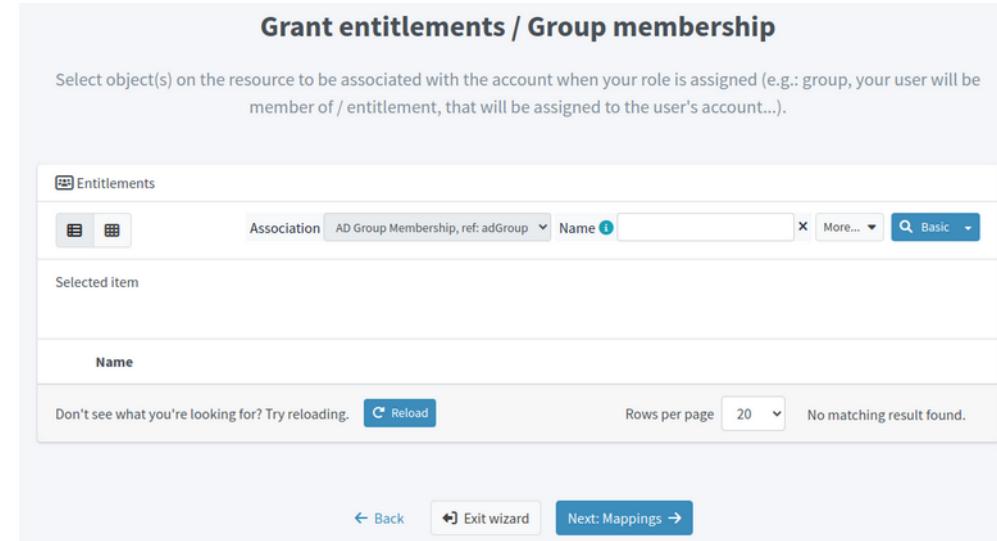
**Entitlements**

Association: AD Group Membership, ref: adGroup | Name:  More...

Selected item

Name

Don't see what you're looking for? Try reloading.  Rows per page: 20 | No matching result found.



# Adding AD Account Inducement for Persons, Part V.

- Additional mappings for Persons on AD are possible, but we're not going to use this feature now
- Keep defaults

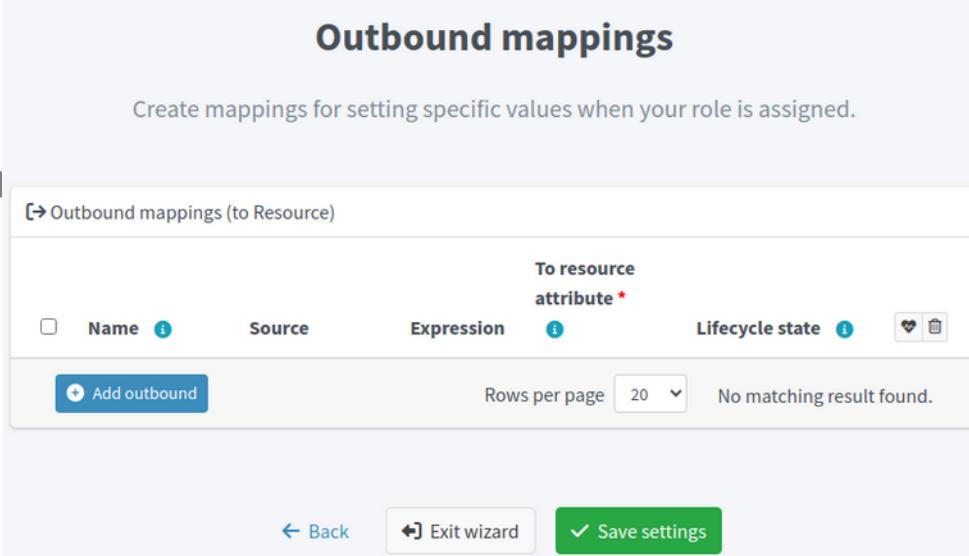
**Outbound mappings**

Create mappings for setting specific values when your role is assigned.

| Outbound mappings (to Resource)             |                       |        |                             |  |
|---|-----------------------|--------|-----------------------------|--|
|   | Name <small>i</small> | Source | Expression <small>i</small> | To resource attribute <small>*</small> |
| <input type="button" value="Add outbound"/> |                       |        |                             |  |

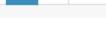
Rows per page  No matching result found.

[Back](#) [Exit wizard](#) [Save settings](#)



# Adding AD Account Inducement for Persons, Part VI.

- Wait for the next reconciliation run (60 seconds)
- Users now have AD accounts

|                          | Name      | Personal Number | Full name          | Email | Accounts |   |
|--------------------------|-----------|-----------------|--------------------|-------|----------|---|
| <input type="checkbox"/> | abaker    | 1021            | Alice Baker        |       | 2        |          |
| <input type="checkbox"/> | abaker2   | 9001            | Andreas Baker      |       | 2        |          |
| <input type="checkbox"/> | cwhitehe  | 1039            | Charles Whitehead  |       | 2        |          |
| <input type="checkbox"/> | cwhitehe2 | 9002            | Clara Whiteherring |       | 2        |          |
| <input type="checkbox"/> | cwhitehe3 | 9003            | Clara Whiteherring |       | 2        |       |
| <input type="checkbox"/> | lcallaha  | 9000            | Louise Callahan    |       | 2        |    |

# Inducement With Association For AD Group

- Until using roles, we can use Person archetype to provision also group membership(s) for AD accounts
- All users with Person archetype will be members of the group(s) from now on

## Grant entitlements / Group membership

Select object(s) on the resource to be associated with the account when your role is assigned (e.g.: group, your user will be member of / entitlement, that will be assigned to the user's account...).

The screenshot shows a software interface for managing group memberships. At the top, there are tabs: 'Entitlements' (selected), 'Association', 'AD Group Membership, ref: adGroup', 'Name' (with a search input and 'More...' button), and 'Basic'. Below the tabs, it says 'Selected item'. Under 'Name', there is a list with one item: 'cn=all-users,ou=groups,dc=example,dc=com'. At the bottom, there is a message 'Don't see what you're looking for? Try reloading.' followed by a 'Reload' button. The 'Reload' button is highlighted with a red box. To the right, there are buttons for 'Rows per page' (set to 20), '1 to 1 of 1', and navigation icons (<<, <, 1, >, >>).

# Displaying Associations

- Associations are displayed in projection details in Associations container
- Just like the account attributes, the data is fetched from resource

The screenshot shows a user profile for 'Louise Callahan' with the following details:

- cn=Louise Callahan,ou=users,dc=example,dc=com** (with a copy icon)
- Resource: AD
- Normal Account

The interface includes tabs for **Basic** and **Attributes**, and a collapsed section for **Associations**. A red box highlights the **Associations** section, which contains the following data:

| AD Group Membership * | cn=all-users,ou=groups,dc=example,dc=com: ShadowType                 |
|-----------------------|--|
| <a href="#">Edit</a>  | <a href="#">cn=all-users,ou=groups,dc=example,dc=com: ShadowType</a> |

# Assignments and Inducements: Bottom Line

- Assignment and inducements form a policy, that object (e.g. User) *should have* something (e.g. a Role)
- MidPoint will make sure the resource account exists
- Combined with strong mappings, midPoint will make sure the resource account exists and attributes and/or associations with groups are populated as defined in outbound mappings



# Tasks in midPoint

- Import from resource
- Reconciliation
- Trigger Scanner task
  - Recomputes objects with triggers (interval: 5 min.)

# Tasks in midPoint (2)

- Validity Scanner task
  - Recomputes objects if activation/effectiveStatus changes (interval: 15 min.)
- Shadow Refresh Task
  - Pushes pending changes and executes shadow maintenance operations (e.g. removes dead shadows after their retention period expires (default: 7 days))

# What Are Dead Shadows?

- MidPoint stores **account identifiers** and **metadata** for accounts in its repository: intermediate Shadow objects
- Dead shadow contains information about deleted resource object
- There is still reference between User and the dead shadow object (not displayed by default)
  - User by midPoint to re-create new account (and new shadow) object

## Shadow

```
name = "cn=Alice Baker, ..."  
...  
dead = "true"  
deathTimestamp = "2023-12-31T12:00:00"  
exists = "false"  
resourceRef = "AD"  
objectClass = "inetOrgPerson"  
kind ="account"  
synchronizationSituation = "deleted"  
...  
attributes:  
    ri:dn = "cn=Alice Baker, ..."  
    ri:entryUUID = "b3a5c588-e003-103d-8175-  
5d0d35036f2f"
```

# Module 8: Labs

LAB 8-1: Generate Usernames in midPoint

## Module 8: Labs

LAB 8-2: Automate Active Directory Account  
Creation For All Persons

## Module 8: Labs

LAB 8-3: Automate Active Directory Group  
Membership For All Persons

# Module 8: Labs

LAB 8-4: Enforcing AD Account Data

# Module 8: Labs

LAB 8-5: Handling HR Data Updates

# Module 8: Labs

LAB 8-6: Handling Long-term Leave

# Module 8: Labs

LAB 8-7: Handling Leavers

## Module 8: Labs

LAB 8-8: Adding A New Outbound Mapping  
TODO  
BONUS?

## Module 8: Labs

LAB 8-9: Adding New Attribute Provisioning From  
HR to AD TODO BONUS?

## Module 8: Labs

LAB 8-10: Exchanging Inbound Mapping  
TODO  
BONUS?

# Module 8: Self-assessment

- TODO

# Module 8: Summary

- Object templates
- Username generator concept for Persons
- Assignments and inducements
- AD group association
- Delayed delete concept
- MidPoint tasks

# Module 8

End of module

# Module 9

Overriding Incorrect Data

# What To Do With Incorrect Source Data?

- MidPoint is now configured to take data from authoritative source system, process them and provision to AD ...
- ... that means, also mistakes from authoritative system will be provisioned ...
- ... but HR is authoritative for data, right?
- What should we do if HR data is incorrect and cannot / will not be corrected *fast enough*?
- Incorrect data in general, incorrect user state

# Incorrect Source Data Override

- If source system contains incorrect data, we can override using midPoint until it is fixed
- Administrative status
- “Invalid data” mark
- Data can be overridden within midPoint



# Status Override Using Administrative Status

- User's **effectiveStatus** is computed from **lifecycleStatus**, **validFrom**, **validTo** and **administrativeStatus**
- User's **effectiveStatus** is used as a source for outbound activation administrative status mapping
- You can override the **effectiveStatus** using **administrativeStatus**

```
activation:  
  validFrom: 2020-01-01 12:00:05  
  validTo: 2099-12-31 12:00:00  
  administrativeStatus: null  
effectiveStatus: enabled
```

```
activation:  
  validFrom: 2020-01-01 12:00:05  
  validTo: 2099-12-31 12:00:00  
  administrativeStatus: disabled  
effectiveStatus: disabled
```

# Status Override Using Administrative Status (2)

- HR data contain incorrect information that account is active
- Reality is that the user must be deactivated immediately (suspected of fraudulent activity)
- Set Administrative status: Disabled
- **Effective status of the user is Disabled**
- ⓘ in 4.8 we can override the status to disable the user, not to enable

```
activation:  
  lifecycleState: active  
  administrativeStatus: disabled  
  effectiveStatus: disabled
```

# Status Override: Forcing User Activation

- HR data contain incorrect information that the account is inactive
- Reality is that the user must be active
- Mark the source account as “Invalid data”
- MidPoint stops synchronizing data from HR for this account
- Change user's Lifecycle state in midPoint

The screenshot shows the MidPoint Accounts interface. At the top, there are navigation buttons: 'Accounts' (selected), 'HR Person' dropdown, a green 'Active' button with a dropdown arrow, a 'Configure' button with a gear icon, and a 'Tasks' button with a grid icon. Below this is a search bar with 'Name' and '9006'. The main table lists accounts. One account row is highlighted with a red box around the 'Invalid data' status indicator. The columns are: Name (with a shield icon), Identifiers (containing 'empnum: 9006'), Situation (containing 'LINKED'), and Owner (containing 'jdoe'). At the bottom of the table are buttons for 'Reload', 'Import', 'Export', 'Create', 'Edit', and 'Delete'.

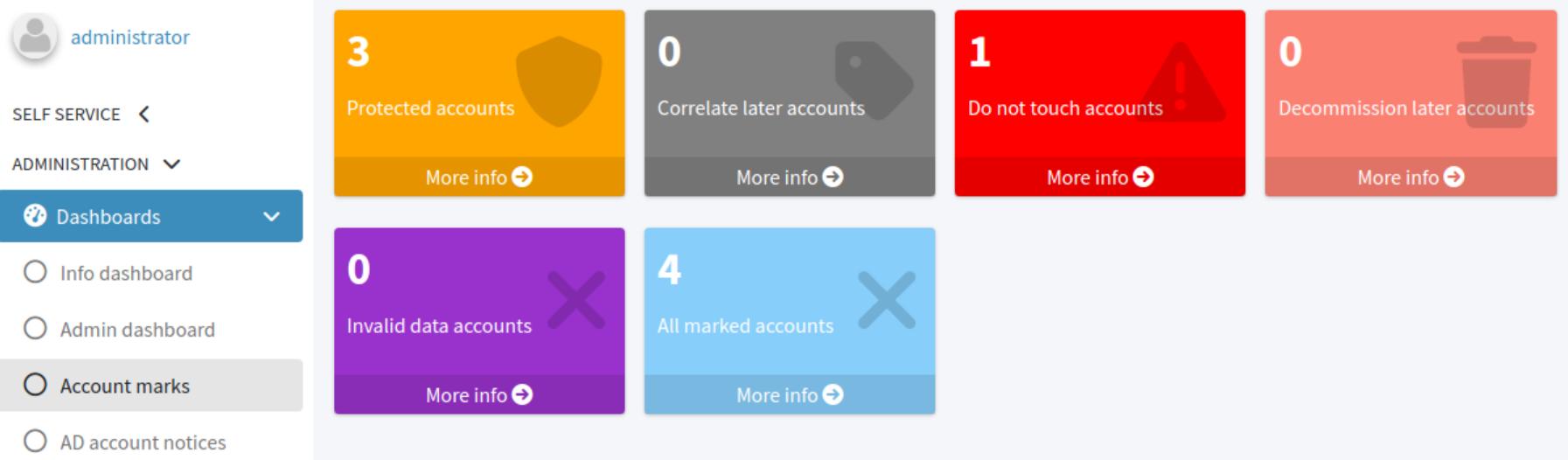
|  | Name                 | Identifiers  | Situation | Owner |
|--|----------------------|--------------|-----------|-------|
|  | 9006<br>Invalid data | empnum: 9006 | LINKED    | jdoe  |

# Overriding Other Data

- HR data contain typos in attribute values
- Mark the source account as “Invalid data”
- MidPoint stops synchronizing data from HR for this account
- Change user's properties in midPoint
  - Locality, Given Name, Family Name ... as needed

# Reporting Account Marks Using Dashboard

- Some exceptions (marks) **should be only temporary**
- Account marks dashboard can be used to report them
- Total count displayed, click to display the list



# Reporting Account Marks Using Dashboard (2)

Invalid data accounts Invalid data accounts X Name ⓘ  X More... Basic

| <input type="checkbox"/> | Account ID   | Situation | Resource | Kind    | Intent  | <input type="button"/>                           |
|--------------------------|--|-----------|----------|---------|---------|--|
| <input type="checkbox"/> | 9006<br>  Invalid data | Linked    | HR       | Account | default | <input type="button"/><br><input type="button"/> |

Protected accounts Protected accounts X Name ⓘ  X More... Basic

| <input type="checkbox"/> | Account ID  | Situation | Resource | Kind    | Intent  | <input type="button"/>                           |
|--------------------------|---|-----------|----------|---------|---------|--|
| <input type="checkbox"/> |  cn=WWW Service Account,ou=users,dc=example,dc=com<br> Protected              |           | AD       | Account | default | <input type="button"/><br><input type="button"/> |
| <input type="checkbox"/> |  cn=Mail Service Account,ou=users,dc=example,dc=com<br> Protected             | Unmatched | AD       | Account | default | <input type="button"/><br><input type="button"/> |
| <input type="checkbox"/> |  cn=Spam Assassin Service<br>Account,ou=users,dc=example,dc=com<br> Protected | Unmatched | AD       | Account | default | <input type="button"/><br><input type="button"/> |

Rows per page 20 1 to 3 of 3 << < 1 > >>

# Reporting Account Marks Using Report

## Widget

Report generated on: Friday, 29. Sep 2023 13:39:27

| Label                       | Number | Status |
|-----------------------------|--------|--------|
| Protected accounts          | 3      |        |
| Correlate later accounts    | 0      |        |
| Do not touch accounts       | 1      |        |
| Decommission later accounts | 0      |        |
| Invalid data accounts       | 0      |        |

## Protected accounts

Report generated on: Friday, 29. Sep 2023 13:39:27

| Account ID   | Situation | Resource | Kind    | Intent  |
|--|-----------|----------|---------|---------|
| cn=Mail Service<br>Account,ou=users,dc=example,dc=com          | Unmatched | AD       | Account | default |
| cn=Spam Assassin Service<br>Account,ou=users,dc=example,dc=com | Unmatched | AD       | Account | default |
| cn=WWW Service<br>Account,ou=users,dc=example,dc=com           |           | AD       | Account | default |

# Username Override – Change of Names

- To force username change when user's Given name / Family name is changed, delete name and save user
  - New unique username will be generated
- Username is mapped to AD's username, it will be changed as well

# Username Override – User's Decision

- To force username change for other reasons, simply set `name` in midPoint to a desired value
  - MidPoint will check if the username is unique
  - Useful if the generated username is an “inappropriate word”
- Username is mapped to AD's username, it will be changed as well

# Module 9: Labs

LAB 9-1: Overriding Malicious User Status

# Module 9: Labs

LAB 9-2: Overriding Incorrect HR Data

# Module 9: Labs

LAB 9-3: Overriding Username

# Module 9: Self-assessment

- TODO

# Module 9: Summary

- Concepts for overriding incorrect HR data
- Administrative status, Invalid data account mark
- Reporting of account marks (dashboard, report)
- Username override

# Module 9

End of module

# Conclusion

Quod erat demonstrandum  
(What was to be demonstrated)

# Goals vs Results: Verification

- We have deployed midPoint and connected first source and target systems in iterations using First Steps Methodology
- We have used only GUI, no XML
- We have cleaned up orphaned accounts in AD and set the policy to detect orphaned accounts in the future



## Goals vs Results (2)

- We have automated the provisioning from HR to AD: accounts and single group membership with full user life cycle support (joiner/leaver)
- We have re-used usernames from AD for midPoint
- We have used simulations to prevent unexpected changes and deletions in AD



# Goals vs Results (3)

- We have corrected AD data to correspond to HR data
- We have prepared midPoint for overrides if HR data is incorrect or if username needs to be changed



# Utilizing First Steps Methodology: Results

| # | Step                       | Description, goals  |
|---|----------------------------|---|
| 1 | Connect Source System (HR) | We have connected the HR application using CSV export and CSV connector.  |
| 2 | Import Source Data         | We have imported data from source system and created users in midPoint  |
| 3 | Connect Target System      | We have connected AD using a predefined resource template.  |
| 4 | Target System Integration  | We have correlated existing accounts to midPoint users. We have created exceptions using marks (protected etc.) |
| 5 | Import Usernames           | We have imported usernames from AD to midPoint. We have also deleted orphaned accounts.                         |

# Utilizing First Steps Methodology: Results (2)

| # | Step                                 | Description, goals  |
|---|--------------------------------------|---|
| 6 | Enable Provisioning to Target System | We have prepared AD resource for provisioning from midPoint, simulated what would happen before it happened.<br>                 |
| 7 | Automate Integration                 | We have automated the AD account provisioning based on HR data in regular intervals. MidPoint is used to generate usernames.<br> |
| 8 | Override Incorrect Data              | We have tested situations when HR data is incorrect and midPoint must override the data to maintain business.<br>                |

# Next Steps

- Which steps should be “next” – well, it's up to you
- Continue with connecting other target systems (one by one)
  - Simulated correlation, simulated outbound mappings, automation using Person archetype inducement
  - No username import



## Next Steps (2)

- Continue with connecting other source system for different population
- Import AD groups as roles and start managing them (and their membership) from midPoint



# | Q & A



# Feedback Appreciated

- This training has been prepared based on real deployments, real issues and with love
- We encourage you to provide your feedback especially if we omitted something important that could be considered as “First steps”



# Thank you for your attention

Do you have any **questions**? Feel free to contact us at [info@evolveum.com](mailto:info@evolveum.com)

**Follow us** on social media or **join us** at GitHub or Gitter!



/Evolveum



@Evolveum



/Evolveum



/Evolveum



/Evolveum



/Evolveum

**Evolveum**

© 2023 Evolveum s.r.o. All rights reserved.