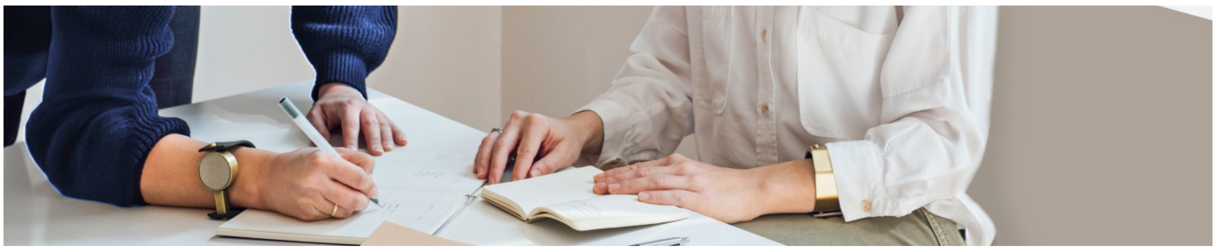
RFC

Requirement Summary for Group Management System



Rolf Brugger

Document type:	RFC
Version:	V1.0
Created on:	01.02.2026
Last updated:	01.02.2026
Classification:	Public

8

9

Versions

Version	Author	Main updates	Date

Contents

Introduction

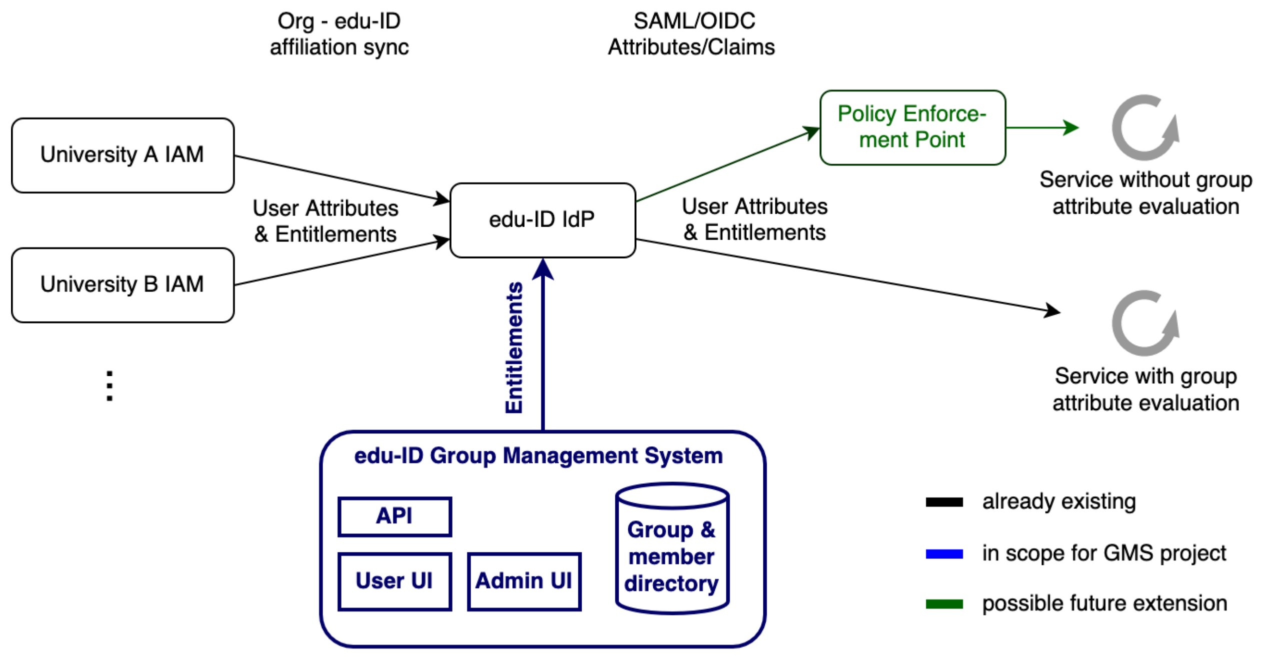
The attribute-based access model is used to control access to a service by persons with an edu-ID account. After a person has authenticated themselves for a service via edu-ID, the IdP provides a set of attributes that the service can evaluate for access control. With the current attribute model, global or general access rules such as ‘students at a university of applied sciences’, ‘employees at University A’, etc. can generally be implemented well. More finely defined access rules, especially those involving individuals from different universities, are difficult to implement with the current model because often they don't share a common attribute.

This document is not a specification. It summarizes a number of concrete use cases for group management at universities. Requirements and a high-level architecture are derived from these use cases.

The purpose of the Switch edu-ID Group Management System (GMS) is to enable flexible management of access authorisations for individuals. An individual is granted access to a service or a specific authorisation within the service by being added to a corresponding group. Group membership is represented as an attribute in the edu-ID identity by the IdP. A service evaluates the attribute and grants the corresponding authorisation.

The GMS proposed for edu-ID includes the following functions:

- Flexible, dynamic formation of groups
- Flexible assignment of individuals to groups, regardless of their organisational affiliation
- Manual management of group memberships via web GUI
- Automatic/programmatic management of group memberships via API



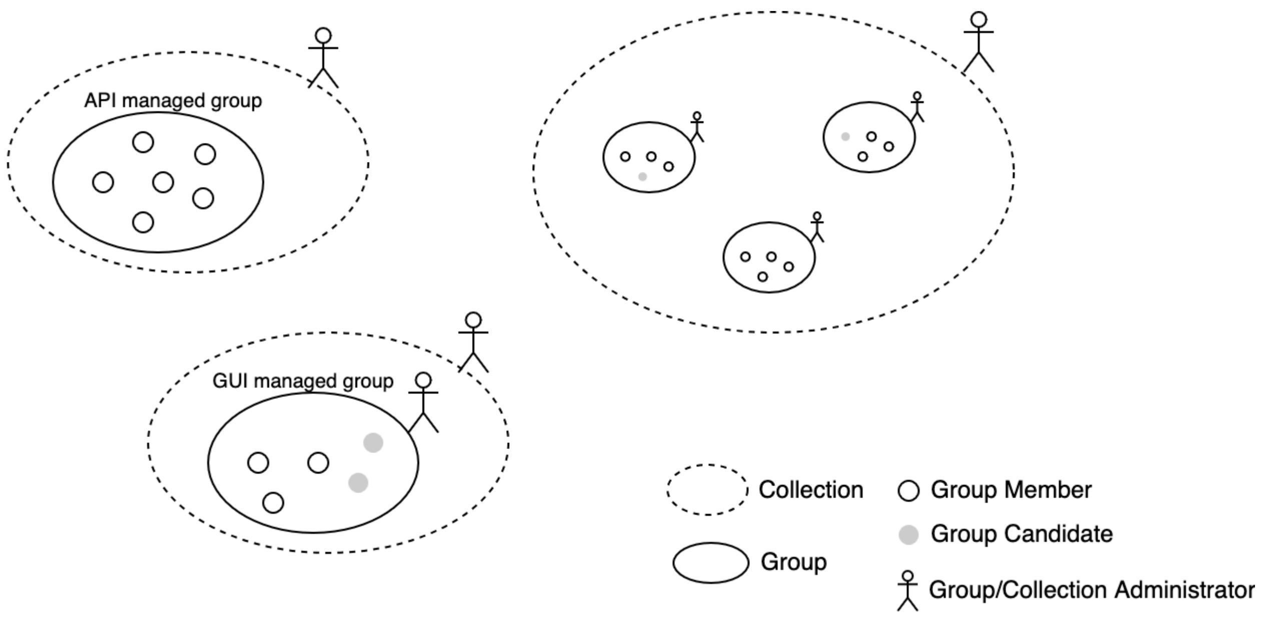
Org - edu-ID affiliation sync
SAML/OIDC Attributes/Claims
University A IAM
University B IAM
⋮
User Attributes & Entitlements
edu-ID IdP
User Attributes & Entitlements
Policy Enforce-ment Point
Service without group attribute evaluation
Service with group attribute evaluation
Entitlements
edu-ID Group Management System
API
User UI
Admin UI
Group & member directory
already existing
in scope for GMS project
possible future extension

In the proposed model, a service must be able to evaluate the group attributes of an authenticated person. For services that cannot evaluate attributes for access management, a policy enforcement point (PEP) could be provided as a future extension. This allows access rules to be defined outside the service. The PEP is not part of the current GMS project.

A new GMS for edu-ID should fully replace the following legacy services: AAI virtual home organisations (VHO), edu-ID shared attributes API and AAI group management tool.

GMS Data Model

The GMS allows for the flexible creation, updating and removal of groups. For efficient management, groups can be combined into collections and administered collectively. People can be added to groups in various ways, either manually or via API. If a person's edu-ID identity is known (by means of a unique identifier like unique-ID or email address), they can be added to a group directly. If the invited person does not yet have an edu-ID account or their edu-ID account identifier is unknown, they are added as a group candidate. Once the account has been created and the group invitation has been linked to the personal edu-ID account, they become a regular group member.



72 Components

Component	Description	Properties	Associations
Group	A group is an object that contains an arbitrary number of (group) members. A group is always part of a collection. A group can be managed via GUI and via API.	• Group Identifier • Group name	• collection reference • list of group administrators • list of group members
Candidate	A group candidate is a user that is not yet linked to an edu-ID identity. The linking process with an edu-id identity has not yet been completed.	• invitation email address • linking code • First name, last name • membership expiration date	
Member	A group member is a user that is associated to a group, and refers to an existing edu-ID identity. Such identities have group membership permissions. As a result of membership, the identity receives a group attribute (Entitlement, isMemberOf) of the corresponding group and collection.	• edu-ID identifier • First name, last name • membership expiration date	• list of groups with membership • membership expiration dates
Collection	A collection contains an arbitrary number of groups. The purpose of a collection is the uniform administration of logically or administratively related groups.	• Collection Identifier • Collection name • API credentials • Associations	• list of groups the collection contains • list of collection administrators
Group Administrator	A group member is a user that is associated to a group, and refers to an existing edu-ID identity. Such identities have group administrator permissions.		

Collection Adminis-trator	A group member is a user that is associ-ated to a collection, and refers to an exist-ing edu-ID identity. Such identities have collection administrator permissions.		
Superad-min	These are edu-ID identites with superad-min permissions		
Web GUI	A graphical web user interface for all group management system users		
API	An API to manage groups and group members.		

73 Requirements

74 Collections

#	Requirement	Description
REC-001	collection creation	A collection can be created by the Superadmin. It is then dele-gated to a collection administrator
REC-002	collection update	A collection can be further managed by the collection adminis-trator

75 Group lifecycle

#	Requirement	Description
REC-101	group creation by collec-tion admin	Creation of a group by the collection admin in the Web GUI
REC-102	group modification by admin	Modification of a group in the Web GUI by the collection admin or the group admin
REC-103	group deletion by admin	
REC-104	group management via API	Creation/update/deletion of a group via SCIM API.

76 Membership management via GUI

#	Requirement	Description

REC-201	Invitation by email	A group candidate is invited to be a group member. The user is identified by email address, first name and last name. Optionally, a membership expiration date is set. The GMS checks if an edu-ID account with that email address exists 1. edu-ID account exists: 1. the user is directly added as member to the group 2. the user is notified by email, that they were added as a group member 2. edu-ID account does not exist 1. the user is added as candidate to the group 2. the user is notified by email, that an edu-ID account containing the email address is required to become a member. The user should either create an account or add the email address to an existing account. 3. as soon as the account was created by the user, they are added to the group as member. A confirmation of the membership is sent by email.
REC-202	Bulk invitation by email	A list of candidates (CSV text file) can be uploaded to add a large number of candidates at once.
REC-203	Reminder for candidates	In a group, the membership status is shown. Candidates can be reminded by email to do the necessary to be added a group members.
REC-204	Candidate/member removal	A candidate or member can be removed from the group • by the group administrator • by the member herself • automatically when the account expiration is reached
REC-205	Bulk removal by email	A list of candidates/members can be removed at once by uploading a list of email addresses.
REC-206	Membership status overview	A group manager can visualize the members and candidates of a group in a web application. They can trigger actions like removal, expiration date extension, reminder sending. The web application is easy to be used by non-technical staff.

77 Membership management via API

#	Requirement	Description

REC-301	Membership management via API	Group Members can be fully managed via API: list/add/remove member
REC-302	Identificator	Members are identified by their swissEduID or eduPersonUniqueID identifier of the private identity. Any of these identifiers can be used alternately.
REC-303	SCIM 2.0 standard compliant	Basic membership functions can be performed via standard SCIM 2.0 API.

Membership features

#	Requirement	Description
REC-401	Group Entitlement	For each group membership, an entitlement value is set in the private identity of a user in the attribute eduPersonEntitlement. The entitlement value contains the group identifier and the collection identifier. Example: https://eduid.ch/gms/<collection-ID>/<group-ID>

Use cases

Use cases for the group management system (GMS)

UniBE further education

Description:

- Further education students need access to the UniBE course platform Ilias. These students are not onboarded as regular UniBE members and they have no UniBE affiliation.
- Registration of further education students: the registration office registers students. Before the course starts, the course manager registers the participating students in the course.

Stakeholders:

- registration office
- further education administrator (for a domain of courses or for all courses)
- course manager (for one course)
- member (further education student)

Requirements:

- Groups are created by the further education administrator.
- Groups are mapped to Ilias resources using the entitlement attribute.
- The further education administrator delegates the membership management to course managers.
- Course managers manage the members of a group
    - add members by uploading member lists

    - add individual members
    - remove members by uploading a list
    - remove individual members
- Identification
    - members are identified by their email address
    - the course manager is informed about email addresses that cannot be matched to an edu-ID account
    - it should be possible to match an email address if
- List group members
    - first name, last name, email address, matching edu-ID found (y/n), uniqueID, creation date, expiration date, custom field
- Automatic expiration
    - When adding group members an expiration date can optionally be set by the course manager
    - The course manager gets a notification before memberships expire
    - The course manager can optionally extend the expiration date
- Mailing functions
    - send emails to members

DeepL SSO

Description:

- Universities acquire usage licenses for DeepL translate and DeepL write from Switch Procurement. The licenses are expensive and usually assigned to selected university members.
- Option 1 - maintain current registration process
    - License administrator at university sends list of university members (names and email addresses) who are allowed to use DeepL to PROC
    - PROC maintains an Excel sheet with all entitled users
    - PROC can easily 1-way-sync the Excel sheet to the GMS. This is done with an upload script that syncs entitlements via API to the GMS by adding or removing users from group.
- Option 2 - delegated group management
    - PROC creates an empty group per organisation
    - The membership management is delegated to the license administrator at the organisation.
- Option 3 - access management by organisation IAM
    - Organisations manage their member using their own tools (IAM processes, internal service "shops", ...)
- PROC is billing the service to participating organisations according to the usage of DeepL by the users.

Stakeholders:

- Switch PROC Staff members
- License administrators (at university)
- Users (university members who access DeepL service via edu-ID SSO)

Requirements:

- Groups are manually created created by PROC.
- (Option 1) The memberships are synced via an API. Users are identified by their email address

- (Option 2) The membership management is delegated to the license administrator at the organisation. Users are identified by their email address. The license administrators...
    - add users by uploading member lists
    - add individual users
    - remove users by uploading a list
    - remove individual users
- (Option 3) The memberships are specified in the affiliation
    - For authorised users, group membership is stored in the attribute Entitlement or isMemberOf of the affiliation.
- Users with group membership have an according entitlement in their private identity.
- The DeepL service is configured so that only people with the entitlement get access.
- Reporting: PROC gets access statistics for a definable period
    - unique users from an organisation who accessed the service during a specific period

Access for private library customers via SLSkey

Description:

- Library patrons without university affiliation must go through special authorisation processes. All these processes are implemented in SLSP Alma. The outcome of the processes is if a library patron is allowed to use the licenses of a library or not.
- Users who are allowed to use the license package of a library will receive a corresponding label (group membership) in their personal edu-ID account . This allows them to directly access a content provider. (Content providers need to evaluate the entitlement for access management.)

Stakeholders:

- ExLibris Alma on which SLSKey permission processes are implemented
- Patrons (library customers without university affiliation)

Requirements:

- Users can be added and removed to groups via API
- SLSKey potentially manages a large number of groups (at least one per library). SLSKey needs a simple procedure to add new groups and get the necessary API credentials.
- Each group has an associated entitlement value.
- Users in a group get the according entitlement value in their personal identity
- Users are identified by their private uniqueID

Access to FHNW library for non affiliated school teachers

Description:

- FHNW manages the access of school teachers of the cantons AG, BL, BS and SO to the FHNW library.
- The user lifecycle of the school teachers is managed by FHNW ERP and IAM.

Stakeholders, Components:

- FHNW IAM (to manage access rights or school teachers)
- FHNW Library (as part of SLSP)
- school teachers

Requirements:

- FHNW synchronizes school teacher memberships to groups in the GMS via API
- There is one group per canton (AG, BL, BS and SO)
- For each group a dedicated entitlement is used.
- The API should be standard SCIM, using the group resource. The SCIM API for groups should be coherent with the existing SCIM API for Affiliations, and SCIM API for Users. This allows to use one single SCIM connector in the FHNW IAM for edu-ID to manage Users (limited), Affiliations and Groups.
- To ensure a smooth migration path, the users are identified in the API by their swissEduID identifier.

Case University of Geneva

Requirements:

- Scope of a given group based one mail (private)
- Web app for management of groups
- Could be created as sub-groups and access added to applications to given groups or sub-groups.
- Closed off list of members added by manager
  - Import system for a large amount of items
  - Same import system for removal of access.
- Membership Validity
  - Date to date
  - Predefined amount of time (month,6months,1,2yearsetc.)
  - Indefinite time
- User account suspension capability for managers
- Managers and such roles in admin system
- Password management for users
- Email notification (automated is necessary) with texttemplate edition with variables
  - for a affliation addition
  - account creation
  - password change (not needed anymore)
  - account suspension
- Review of: User rights, last access etc.
- Statistics: last usage of a group...

UNIGE is not responsible for the group. The manager oversees the people he chooses to add to the group and the group itself.

Case ZHAW external affiliates

Description:

- Access to ZHAW services must be made available to individuals who have an edu-ID account, but no ZHAW account. This includes user groups like
    - prospective students who are admitted to study but do not yet have a ZHAW account
    - Aptitude assessments i.e. for language studies on Moodle
    - students who have left the university
    - summer course students
    - visiting lecturers
    - visiting students
- The user lifecycle and permissions are managed by ZHAW ERP and the future IdM.

Stakeholders, Components:

- ZHAW ERP and IAM/IdM (to manage guests and their access rights)

Requirements:

- ZHAW synchronizes affiliate memberships to groups in the GMS via API
- There is at least one separate group per department for beginners, alumni and guests, leading to a few dozen groups to be managed.
- For each group a dedicated entitlement value is used.
- To ensure a smooth integration path, the users are identified in the API by their eduPersonUniqueID identifier