

# Eduteam Account Registry Flow

To clarify the [requirements](#) for the Account Registry (AR) Component, I'll try to describe the flow that happens when a user logs in to eduteams/SCZ.

This is split in separate sections for the Login flow, the Registration flow, the Account Management flow, and the Account Linking flow. Only the first two of those are elaborated below.

A proposed data model is included at the end of the document.

## Login flow

1. User is invited to a CO in SBS; she gets an email and clicks the link to <https://sbs.scz.example.edu/>
2. SBS redirects the user to SATOSA to log in; SATOSA redirects to the discovery screen, user picks IdP, logs in and is redirected back to SATOSA
3. SATOSA microservice generates one or more hashes based on idp-identifiers it has determined to be useful to identify/lookup whether a user exists (this is existing eduTEAMS code). These "internal user identifiers" (iuid) are further used to identify a user internally in the platform; multiple hashes are used to track if an idp adds or changes identifiers (e.g., adds support for subject-ids) so to still be able to look up the user.
4. SATOSA invokes a microservice that send a backend query to the AR to determine if the user is already registered in the platform; the microservice sends an API call to the "identity check" endpoint of the AR:

`https://ar.scz.example.edu/check-identity`

of the form (number of hashes is 1 or more):

```
{
  iuid: [
    "4355a46b19d348dc2f57c046f8ef63d4538ebb936000f3c9ee954a27460dd865",
    "53c234e5e8472b6ac51c1ae1cab3fe06fad053beb8ebfd8977b010655b added3c3",
    "1121cfccd5913f0a63fec40a6ffd44ea64f9dc135c66634ba001d10bcf4302a2",
    "7de1555df0c2700329e815b93b32c571c3ea54dc967b89e81ab73b9972b72d1d"
  ]
}
```

The AR responds with:

- "Unknown user" (HTTP 404) if none of the attributes match anyone in the database;
- "Error: conflicting matches" (HTTP 409) if multiple hashes match different users;

- “Matched User” (HTTP 200) if one or more attributes match the same user. It then return a list of which of the identifiers matched and a list of the registered user’s attributes in the form:

```
{
  "result": "match",
  "matches": {
    "4355a46b19d348dc2f57c046f8ef63d4538ebb936000f3c9ee954a27460dd865": true,
    "53c234e5e8472b6ac51c1ae1cab3fe06fad053beb8ebfd8977b010655bfdd3c3": false,
    "1121cfccd5913f0a63fec40a6ffd44ea64f9dc135c66634ba001d10bcf4302a2": false,
    "7de1555df0c2700329e815b93b32c571c3ea54dc967b89e81ab73b9972b72d1d": true
  },
  "user": {
    "cuid": "9706aa89-6012-4ee1-99fa-87689f1a47b4",
    "iuid": [
      "4355a46b19d348dc2f57c046f8ef63d4538ebb936000f3c9ee954a27460dd865",
      "7de1555df0c2700329e815b93b32c571c3ea54dc967b89e81ab73b9972b72d1d",
      "f0b5c2c2211c8d67ed15e75e656c7862d086e9245420892a7de62cd9ec582a06"
    ],
    "displayName": [ "Jane Doe", "贾内朵埃" ],
    "mail": [
      "jane.doe@uniharderwijk.nl",
      "jane@myvo.example.org"
    ],
    "eduPersonPrincipalName": [ "jane@uniharderwijk.nl" ],
    "eduPersonScopedAffiliation": [
      "employee@physics.uniharderwijk.nl",
      "member@uniharderwijk.nl",
      "employee@uniharderwijk.nl"
    ],
    "postalAddress": "Gebäude 465\nRaum 325\nBrandenburgische Straße 85\nBerlin",
    "County": "Germany",
    "telephoneNumber": "+49305836429",
    "preferredLanguage": "zh-Hant"
  }
}
```

Where the cuid (community user identifier) is the main platform identifier. The attributes shown here are an example; the exact set of things to register will probably change.

5. If necessary, the microservice updates the user’s identifiers to the AR via REST call. The microservice sends a PATCH request to the user endpoint

`https://ar.scz.example.edu/user/9706aa89-6012-4ee1-99fa-87689f1a47b4`

of the form

```
{
  "iuid": [
    "4355a46b19d348dc2f57c046f8ef63d4538ebb936000f3c9ee954a27460dd865",
    "7de1555df0c2700329e815b93b32c571c3ea54dc967b89e81ab73b9972b72d1d",
```

```

        "f0b5c2c2211c8d67ed15e75e656c7862d086e9245420892a7de62cd9ec582a06"
    ]
}

```

To replace the iuid record of this user with the new values.

6. If the response from the AR was a 'not found', the microservice parks the user's SAML state (**Determine if this is necessary and how to handle this (for example sending an extra "redirect url" parameter to the AR, so that the user can be redirected to their original destination. No need to implement this in the first version)**) and redirects the user to the registration interface of the AR.  
[continue in Registration Flow]
7. Handling new/updates user attributes.  
If the user's IdP attributes have changed (other than the iuid) these will need to be updated at the AR. **How should this work, exactly? The AR would need to keep a record of the upstream IdP attributes, sends them along with the match user record in step 4 in order for the microservice to compare them to the incoming attributes. If anything changed, the microservice would then redirect the user to a "Account update" screen. No need to implement this in the first version.**  
If the user is fully registered and known, the microservice does some attribute mangling (discards IdP attributes, and gathers attributes from the AR and possibly the MMSes) and redirect the user back to the SP.

## Registration flow

1. The user is redirected to the registration interface of the AR. The AR initiates a normal SAML flow and the user should be logged in without interaction (SSO).
2. The user is identified by a set of attributes, the so-called IUID (internal user identifier) set. This IUID is generated by SATOSA and sent to the AR as a single SAML attribute (voPersonExternalId). It is a list of hashes, each of which should be recorded as a iuid-identifier by the AR.  
**Note: SATOSA will need to include some attributes about the user's IdP; specifically the entityid and the display name.**
3. Duplicate check:  
When the user logs in, the AR checks if all of the values of the IUD set match to the same user. If all of them match (ie user is registered already), redirect her to the Account Management interface (see below)
  - a. If none of them match (ie unknown user), continue with registration as outlined below
  - b. If some of them match and the others are unknown, show an error page. This should never occur, as this case was handled by satosa already
  - c. If multiple of them match different users, throw an error (should also already have been caught by SATOSA).

4. Show the user a welcome screen, explaining that she was supposed to go to Service X, but that she needs to register first.

The screenshot shows a web browser window with a single tab labeled "Page 1". The address bar displays the URL "https://ar.scz.example.edu/register/". The page content features the "eduTeams Collaboration Management" title and logo. A message informs the user that the service is part of the eduTeams ecosystem and requires registration. A three-step registration process is outlined: "Agree to ToS", "Verify personal data", and "Validate email", connected by dashed arrows. A list of these steps is provided below the flowchart. An orange button labeled "Continue to Term of Service" is positioned at the bottom left.

Page 1

https://ar.scz.example.edu/register/

eduTeams Collaboration Management

The service that you were trying to reach is part of the eduTeams ecosystem of services.  
It seems you have not used the eduTeams service before so we kindly ask you to register here before continuing to the service

Agree to ToS → Verify personal data → Validate email

Registration consists of three steps:

1. Agreeing to the eduTeam term of service
2. Verifying your personal data
3. Validating your email address

Continue to Term of Service


5. When the user clicks, she is sent to the AUP screen:

Page 1

← → ↺

https://ar.scz.example.edu/register/aup

eduTeams Collaboration Management



In order to use the eduTeams service, you need to agree to a number of rules and regulations. These are explained in the document below.

For more information or explanations, you can contact [boss@eduteams.example.org](mailto:boss@eduteams.example.org)

Please read the document, and if you agree, please check the checkmark and press Next

**Acceptable Usage Policy**  
version 20190812

Ut in turpis leo. Sed enim eros, blandit id mi quis, cursus porta ante. Nunc efficitur elit et semper placerat. Aliquam erat volutpat. Mauris sed vestibulum odio. Proin eu urna nec augue interdum posuere. Vivamus pulvinar nibh elit, id efficitur quam sagittis et. Aliquam mi neque, imperdiet sit amet vehicula cursus, dignissim in dolor. Pellentesque finibus imperdiet sapien, eget blandit nibh pellentesque sed. Aliquam a accumsan velit. Etiam vel eleifend erat. Cras sagittis at arcu ac aliquet.

Sed aliquet velit ipsum, nec blandit lectus fermentum ut. Integer fermentum est eget dolor sollicitudin elementum. Sed sit amet mollis nulla. Donec quis tempus sapien. Phasellus at hendrerit sem. Nam commodo justo ac turpis blandit, sed pulvinar arcu rutrum. Nunc fringilla varius sem, quis pulvinar libero accumsan tristique. Vestibulum nibh libero, scelerisque varius mattis ut, semper vitae lacus. Praesent placerat laoreet ante non maximus. Vestibulum nibh magna, porta nec tellus eget, consectetur molestie lacus.

[Download Acceptable Usage Policy \(pdf\)](#)

☒ I Agree to the Acceptable Usage Policy

Continue to  
personal data

After accepting the AUP, the date/time and version number of the AUP is saved in the user's account

Note: the current AUP text and version number should be configurable somehow. Eventually we would like to add functionality to force users to agree to a new version of the AUP, or to remind them to review it once a year. And maybe also manage AUP versions in the AR.

6. After the user agrees, she is sent to the actual registration interface

Page 1

https://ar.scz.example.edu/register/aup

## eduTeams Collaboration Management

Please enter your details. Some have already been provided by your university.

**Provided by University of Harderwijk**

Name: Jane Doe  
贾内朵埃

Email: jane.doe@uniharderwijk.nl

Affiliation: employee@physics.uniharderwijk.nl  
member@uniharderwijk.n  
employee@uniharderwijk.nl

**Local details**

Name: + Jane Doe x  
贾内朵埃 x

Email: + jane.doe@uniharderwijk.nl x

Phone: + +0000000000

Address: Address

Country: Germany

Pref. Language: Traditional Chinese

Continue


Note that the screen first shows the attributes it got from the IdP (which the user can't edit) and after that the "local" attributes, which the user can change, add, remove, etc. The user should at least supply one name and one email address.

7. If a user has entered new email addresses (which do not match an email address from the IdP attributes), it needs to be validated.

Page 1

https://ar.scz.example.edu/register/validate

eduTeams Collaboration Management



In order to continue, the following email addresses need to be validated:

<input checked="" type="checkbox"/>	<input type="text" value="jane.doe@unihardewijk.nl"/>	verified by University of Harderwijk	
<input type="checkbox"/>	<input type="text" value="jane1653@gmail.com"/>	enter code	<input type="text" value="XXXXXXXXXXXXXXXXXX"/> <input type="button" value="Verify"/>
<input type="checkbox"/>	<input type="text" value="jane@wannadoo.nl"/>	enter code	<input type="text" value="XXXXXXXXXXXXXXXXXX"/> <input type="button" value="Verify"/>
<input checked="" type="checkbox"/>	<input type="text" value="verified@email.example.org"/>	verified by email code	

We have sent a verification code to each of the addresses.  
Please enter the code form the email in the appropriate box.

Finish

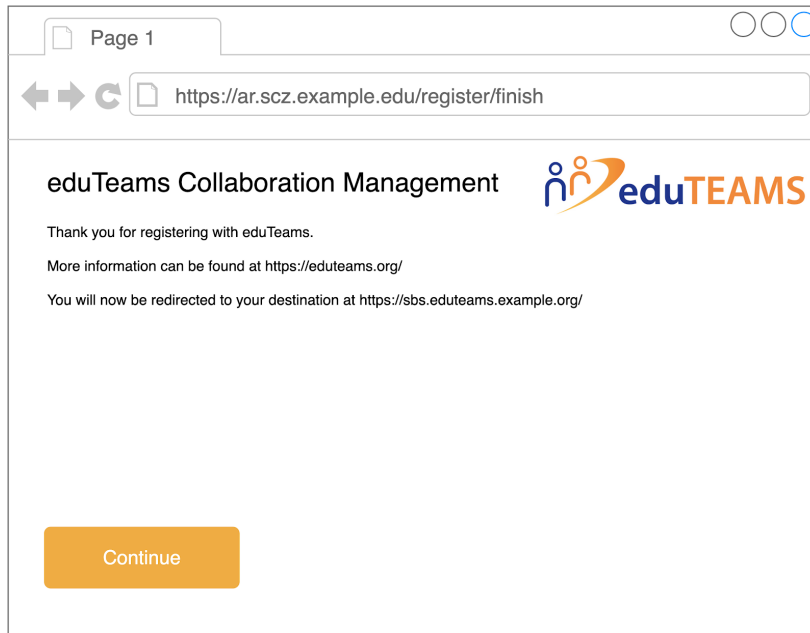
Note that the screen also shows the email address that doesn;t need to be verified as it was supplied by the IdP. The user-entered emails need to be verified: the AR sends a mail to each of the addresses. Within the mail, there is a unique code which the users need to enter here.

We choose this approach rather than a link in the email because it simplifies the user's flow, it simplifies the account handling (**what do we need to do if not all email addresses are verified yet?**), and it prevents any fishing attacks.

If one of the email addresses is correctly identified, the icon in front of the email address changes, and the system notes that the email is verified (bottom example).

The finish button should be greyed out until all addresses are verified.

8. Finally, the AR shows a thank you page and redirects the user back to their original destination ([see remark at item 6 of the login flow](#)).



## Data model

The diagram below shows a proposed data model. This is not meant as fixed model that we definitely need to implement, but rather as an informative diagram of the relationships between the different objects in the system.



