DARIAH AAI Documentation

Federated Single Sign-On

The DARIAH Authentication and Authorization Infrastructure (DARIAH AAI) is based on SAML and Shibboleth in the European higher education identity inter-federation eduGAIN and its members. See AARC Federations 101 Training Module or the DASISH Training on AAI for a gentle introduction to the underlying concepts.

For setting up a service in the DARIAH AAI, you want to protect it with a Shibboleth Service Provider, e.g. by following this SWITCHaai Tutorial. Other SP software following the SAML v2 standard can be used as well. In order to integrate better with the DARIAH AAI, follow this presentation on DARIAH AAI. See below for Service Developer Resources.

DARIAH User Management

Researchers and students in organizations that do not operate an federated Identity Provider can request a DARIAH "homeless" account using the DARIAH SelfService. It can be accessed here: https://auth.dariah.eu/cgi-bin/selfservice/ldapportal.pl.

Useful links:

- Directly request a DARIAH homeless account
- Lost your password?
- DARIAH SelfService User manual

For administrators, there is a DARIAH User Administration which can be accessed here. It allows you to create and manage "homeless" and federated accounts, assign users to authorization groups, e.g. DARIAH Wiki spaces, and manage organizations in a country. See the DARIAH User Administration manual. If you have a question to the admins, please send e-Mail to register@dariah.eu.

For Service Developers

DARIAH AAI is integrated in Higher Education Federations using the SAML standard. This means any Web application should integrate with a so-called SAML Service Provider (SP). The SP will protect your application, driving the log-in process and providing your application with attributes about the user who has logged in using a SAML Identity Provider (IdP) at another organization. Be sure you understand these concepts well, perhaps using the Federations 101 article that is linked above.

In the following, we concentrate on securing your Web application using the Shibboleth SP, which is a widely used and flexible, programming language independent Apache- oder IIS-based module. However, there are other popular Open Source SAML SPs around, such as simpleSAMLphp, pySAML2, mod_auth_mellon, or Spring-Security-SAML, or even commercial ones.

Basics of setting up a Shibboleth Service Provider for your application

Refer to Example Shibboleth SP Configuration for some example configuration files for all use-cases described below.

The Shibboleth SP will do all processing of SAML requests and handling of SAML responses for you. An application only needs to decide on *when* login should occur, evaluate user attributes (provided as environment variables to the application) and base its access decisions upon it. For a first overview, you can follow this SWITCHaai Tutorial. It sums up to two steps:

- Install the Service Provider software on you application server. We highly recommend following the SWITCHaai SP Installation Instructions, as they are given for a variety of operating systems.

- The SP software must be configured. For a federation-independent walkthrough, please use the vendor documentation in the Shibboleth Wiki, https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPGettingStarted

If you are using puppet, there is a puppet module created by SUB Göttingen that has some DARIAH AAI specifics and can be found at https://github.com/DARIAH-DE/puppetmodule-dariahshibboleth.

Registering your Service with the DARIAH AAI IdP Proxy

Since Summer 2018, DARIAH has run an AAI Proxy. Any DARIAH service provider can use the Proxy's Identity Provider component for authentication. The Proxy's SP component, however, is registered in the eduGAIN meta-federation and will allow researchers with any IdP in eduGAIN to log in.

Here is what needs to be done to connect to the DARIAH AAI Proxy:

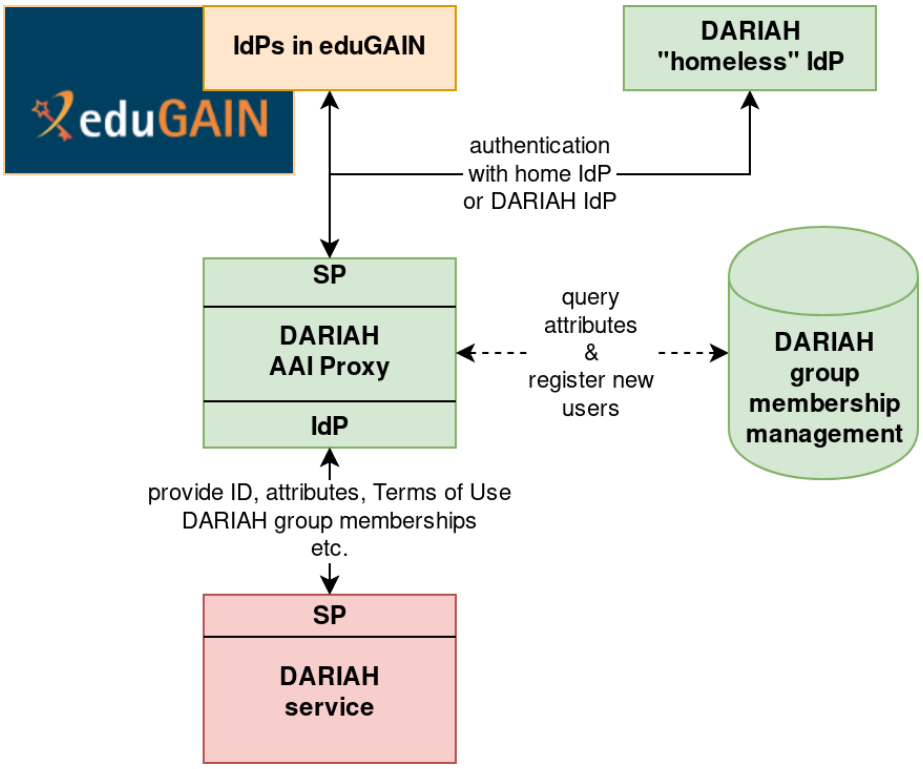
- Save DARIAH Proxy metadata (https://aaiproxy.de.dariah.eu/idp) to you local disk under /etc/shibboleth/ as "dariah-proxy-idp.xml" and load them using <MetadataProvider type="XML" file="dariah-proxy-idp.xml"/> in shibboleth2.xml
- Send your own SP's metadata (from https://your.sp.edu/Shibboleth.sso/Metadata) to register@dariah.eu with a request for entering them at the AAI proxy. Please state whether this service is a test or a production instance.
- Set the SP to direct login using <SSO entityID="https://aaiproxy.de.dariah.eu/idp"> in shibboleth2.xml
- Set REMOTE_USER="eppn unique-id"
- enable the attributes you need in attribute-map.xml, among them you specifically might want to consider
  - <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.13" id="unique-id"><AttributeDecoder xsi:type="ScopedAttributeDecoder"/></Attribute>

Get more out of DARIAH - the DARIAH AAI

The DARIAH AAI has been designed with several goals in mind.

- Goal 1: users of DARIAH services (SPs) should authenticate via their home organization (campus IdP).
- Goal 2: certain DARIAH services only allow particular user groups. This should be configurable centrally by the respective admins, for all DARIAH services.
- Goal 3: DARIAH needs some user information
  - 3.a) she agrees to DARIAH Terms
  - 3.b) she is a researcher (e.g. by her organization or e-mail)
- Goal 4: cope with a situation where users either
  - 4.a) have no campus IdP
  - 4.b) their campus IdP would not release Personally Identifiable Information (PII) to hitherto unknown SPs

Here's a diagram of how the architecture looks like.



Attributes available in the DARIAH AAI - Full list

See the following sections for a description of the dariah-specific attributes teh DARIAH AAI Proxy sends.

These attributes need to be available in the attribute-map.xml configuration file of your Shibboleth SP (usually under /etc/shibboleth).

Some of these attributes are already present there and you only need to remove the comments around them.
The DARIAH-specific attributes need to be added entirely. The order is not relevant.

Attribute name	Attribute oid	Example value	Description
eduPersonUniqueID	urn:oid: 1.3.6.1.4.1.5923.1.1.1.13	abc1234def6789fff000@dariah.eu	Unique identifier within the DARIAH AAI. Recommended for personalisation in services.
eduPersonPrincipalName	urn:oid: 1.3.6.1.4.1.5923.1.1.1.6	john.doe@example.org abc1234def6789fff000@dariah.eu	Is only different from eduPersonUniqueID for legacy DARIAH accounts; Might contain scopes different from "@dariah.eu" for these accounts.
eduPersonScopedAffiliation	urn:oid: 1.3.6.1.4.1.5923.1.1.1.9	student@example.org	Might contain scopes different from "@dariah.eu"
eduPersonAffiliation	urn:oid: 1.3.6.1.4.1.5923.1.1.1.1	member	Will only contain "member" since this attribute is unscoped.
eduPersonEntitlement	urn:oid: 1.3.6.1.4.1.5923.1.1.1.7	urn:mace-dir:common-lib-terms	
cn	urn:oid:2.5.4.3	John Doe	
givenName	urn:oid:2.5.4.42	John	
sn	urn:oid:2.5.4.4	Doe	
displayName	urn:oid: 2.16.840.1.113730.3.1.241	John Doe	
preferredLanguage	urn:oid: 2.16.840.1.113730.3.1.39	DE	
o	urn:oid:2.5.4.10	Example University	organisation
mail	urn:oid: 0.9.2342.19200300.100.1.3	john.doe@example.org	
schacCountryOfCitizenship	urn:oid: 1.3.6.1.4.1.25178.1.2.5	DE	
isMemberOf	urn:oid: 1.3.6.1.4.1.5923.1.5.1.1	textgrid-users	• Contains all DARIAH-specific groups the user is a member of • Can be used for authorisation by your application • Can be multivalued with individual groups semicola-separated
dariahRole	urn:oid: 1.3.6.1.4.1.10126.1.52.5.2	cn=National Representative, c=DE	• Contains the context of all roles the user has within DARIAH • Can be multivalued with individual roles semicola-separated
dariahTermsOfUse	urn:oid: 1.3.6.1.4.1.10126.1.52.4.15	Terms_of_Use_v5.pdf	• Contains all Terms of Use (ToU) documents the user has accepted • Can be used by your application to make sure, that required ToU has been accepted • Can be multivalued with individual groups semicola-separated

The following code is an excerpt of the attribute definitions done in the attribute-map.xml configuration file:

attribute-map.xml

```
<!-- eduPerson attributes -->
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.13" id="unique-id">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
                <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" id="affiliation">
                 <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
        <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" id="unscoped-affiliation"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" id="entitlement"/>

<!-- standard attributes -->
    <Attribute name="urn:oid:2.5.4.3" id="cn"/> <!-- common name -->
    <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
    <Attribute name="urn:oid:2.5.4.4" id="sn"/> <!-- surname -->
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.39" id="preferredLanguage"/>
    <Attribute name="urn:oid:2.5.4.10" id="o"/> <!-- organization -->
    <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.25178.1.2.5" id="schacCountryOfCitizenship"/>

<!-- DARIAH-specific -->
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" id="isMemberOf"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.10126.1.52.5.2" id="dariahRole"/>
    <Attribute name="urn:oid:1.3.6.1.4.1.10126.1.52.4.15" id="dariahTermsOfUse"/>
```

Receiving Attributes from Campus IdPs

Some attributes in the DARIAH AAI are *scoped*, i.e. they contain the domain of the issuing IdP:

- eduPersonScopedAffiliation (Apache Header: **affiliation**), sample value *student@your-university.org*
- eduPersonPrincipalName (Apache Header: **eppn**), sample value *uid1234@your-university.org* (only for legacy federation users, i.e. those who registered before Summer 2018)

The SP must be configured in a way that it will accept scopes different from *@dariah.eu* for these two attributes from the AAI proxy. This is done by commenting some lines in attribute-policy.xml:

/etc/shibboleth/attribute-policy.xml

```
[...]
 <afp:AttributeRule attributeID="affiliation">
            <afp:PermitValueRule xsi:type="AND">
                <RuleReference ref="eduPersonAffiliationValues"/>
<!-- accept any scope
                                <RuleReference ref="ScopingRules"/>
-->
            </afp:PermitValueRule>
 </afp:AttributeRule>
[...]
<!-- accept any scope for legacy users, i.e. comment the eppn policy fully
        <afp:AttributeRule attributeID="eppn">
            <afp:PermitValueRuleReference ref="ScopingRules"/>
        </afp:AttributeRule>
-->
[...]
```

Attribute Use Cases

Receive Central Authorization Information (the DARIAH isMemberOf-attribute)

DARIAH Administrators can assign users to groups, such as "texgrid-users", or "dariah-de-contributors". Such groups can be open for anybody, or upon request - see the DARIAH Self Service Documentation. The central DARIAH directory (DARIAH LDAP server) holds these authorization group information. Your service can use this multi-valued attribute in order to implement fine-grained access restrictions.

attribute-map.xml

```
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" id="isMemberOf"/>
```

Consequently your Shibboleth SP will provide all values it receives as an Apache environment variable with the name *isMemberOf*. Please refer to the according documentation on how to use this in your application. It is also possible to use this information as Apache Access rules.

isMemberOf is a multi-valued attribute, meaning, that it includes all groups the user is a member of with the individual values separanted by semicola **;**.

Example: consider a user that is member of the groups **textgrid-users** and **dariah-de-contributors.** The resulting value of isMemberOf would be "**textgrid-users;dariah-de-contributors**".

Process Role Information

DARIAH adminstrators can operate on a global or national level, or just for a single organization. See the DARIAH User Administration Documentation for the concept of the implementation. The central DARIAH directory (DARIAH LDAP server) holds this information as well. Your service can use this multi-valued attribute in a similar way.

attribute-map.xml

```
<Attribute name="1.3.6.1.4.1.10126.1.52.5.2" id="dariahRole"/>
```

The value of the *dariahRole* attribute will include all roles the user is a member of in the DARIAH LDAP server, once again separated by semicola and in their context, as they do on the LDAP directory.

Example: consider a user with three different roles in the DARIAH LDAP directory:

- **cn=DCO-admin**
- **cn=National Representative,c=DE**
- **cn=orgadmin,o=SUB,c=DE**

The resulting value of *dariahRole* would be "**cn=DCO-admin;cn=National Representative,c=DE;cn=orgadmin,o=SUB,c=DE**".

Assure your Service receives personal Data about the user

If your application needs personal data, e.g. the e-mail address, or displayName, the DARIAH AA can provide this. With the *e-mail* as an example, one would

- uncomment the already existing mapping for "**mail**" in attribute-map.xml

attribute-map.xml

```
<Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
```

Assure user has signed your Service's Custom Terms of Use

In order to check for your Service's Custom Terms, do the following:

- Send your Terms of Use document to register@dariah.eu, stating for which service it is used, which ToU version it is, and, if applicable, which authorization group you use
  - Your ToU document can be either a WWW link (then the URL should contain some version information)
  - Or a HTML-style marked-up file. Either you or Dariah staff will put the file in the *DARIAH Repository* for general access - see this example for the TextGrid service: https://repository.de.dariah.eu/1.0/dhcrud/21.11113/0000-000B-CB4A-E
- This is how you can see on the service side which ToU have been accepted by the user; **the actual check for ToU consent is being handled by the AAI proxy**.

attribute-map.xml

```
<Attribute name="urn:oid:1.3.6.1.4.1.10126.1.52.4.15" id="dariahTermsOfUse"/>
```

Setting Up direct Trust with the DARIAH homeless Identity Provider

The recommended way of connecting a service with DARIAH is via the AAI proxy, see above. However, in some exceptional cases you might want to set up direct trust with the DARIAH "homeless" IdP. This might apply if:

- a) You decidedly do not want members of the eduGAIN federation to use your service, or
- b) Your service already has connections to eduGAIN IdPs via a national federation, and you just want to add DARIAH and its special attributes

The recipe to configure your Shibboleth SP is as follows:

Case a): DARIAH without eduGAIN

- Save DARIAH homeless IdP metadata (https://idp.de.dariah.eu/idp/shibboleth) to you local disk under /etc/shibboleth/ as "dariah-homeless-idp.xml" and load them using <MetadataProvider type="XML" file="dariah-homeless-idp.xml"/> in shibboleth2.xml
- Send your own SP's metadata (from https://your.sp.edu/Shibboleth.sso/Metadata) to register@dariah.eu with a request for entering them at the DARIAH homeless IdP. Please state whether this service is a test or a production instance, and which of the available attributes your service requires
- Set the SP to direct login using <SSO entityID="https://idp.de.dariah.eu/idp/shibboleth"> in shibboleth2.xml
- Set REMOTE_USER="eppn unique-id"
- enable the attributes you need in attribute-map.xml. See above for the attributes that are available.

Case b): Connect to DARIAH IdP via eduGAIN directly

- The DARIAH IdP is a member of eduGAIN already. This means your service needs to be registered with eduGAIN and consume its metadata. Please consult your national higher education federation's documentation for this.
- To ensure the DARIAH IdP sends out the attributes you require, set up CoCo and/or R&S:
    - https://wiki.geant.org/display/eduGAIN/CoCo+Recipe+for+a+Service+Provider
    - https://refeds.org/category/research-and-scholarship
- If you require one of the special DARIAH attributes (see above), please send an e-Mail to register@dariah.eu, specifying which of the available attributes your service requires; and configure them in your attribute map.