AARC-G037

Guidelines for combining group membership and role information from multiple attribute sources

“Guidelines for combining group membership and role information in multi-AA environments”

Task AARC2-JRA1.4B

[Scope: the scope of this document is the multi-AA environment.

- A given user obtains attributes from multiple AAs. They may be a member of more than one VO, or may draw on attributes from different sources
    - (An IdP can be seen as a special case of an AA, issuing identity-related and organisational/community information. However, **the focus of this document should be on <u>** attributes used for authorisation **</u>**)
    - To which extent can IdP-as-an-AA be generalised to other AAs, e.g. does JRA1.3A generalise to non-identity attributes?
    - (Maybe, if it’s useful) also build on JRA1.4A’s description of communicating VO roles
- Suggested two parts of this document:
    - How users select AAs and obtain attributes from them
    - A SP needs to process information from multiple AAs (forwarded by the proxy if there is a proxy)
        - Proxy communicates information for which it is not authoritative
        - Change of scope: AA publishes information for its own scope, proxy for the infrastructure federation
- Cross infrastructure notes: infrastructures using each other’s AAs, cf. JRA1.1A
- AAttributes may be combined by:
    - The user (e.,g. VOMS, users collecting AAttributes from different AAs)
    - **The proxy (since AARC recommends using a proxy, we can focus on the proxy case)**
    - The SP (e.g. e-commerce, with traditional session-cookie setting where the session is the key for a service-based lookup of the stored attributes)
    - Connecting proxies - taking information from other proxies
    - ⇒ The focus of this document should be the PROXY
- How are attributes expressed
    - Not assuming SAML necessarily
    - OIDC - type of claims not (yet) defined - what should be done could be fed into the OIDC RE WG
        - How to translate SAML -> OIDC (cf. RAF for OIDC which is available now)
    - But using the schema as reference

Introduction & Background

The premise behind this work is that users may need or may obtain attributes from more than one attribute authority (AA). This could be because they have different roles maintained by distinct AAs, which in turn could be because the user is a member of more than one VO, or the user uses simultaneously more than one infrastructure and each infrastructure obtains attributes about the user from a separate AA.
In contrast to the attributes used to authenticate to the service/proxy, the general aim of this work is *attributes passed on to the service for authorisation/accounting* (such as membership, roles, etc.). There is obviously no hard boundary between the two.

Particular questions to be covered here are:

- Attributes from different AAs may have different purposes, or different LoA.
    - Different AAs may be operated at different qualities of operation (cf. JRA1.4D): for example, one may be run by an infrastructure as a core service, and another by the user community.
    - Different AAs may have different ways of assigning attributes. An established infrastructure may have documented processes, and a small community may have more ad-hoc processes.
- User choice
    - The user may choose to assert one attribute and not others (LSAAI & VOMS), Affiliation (ELIXIR)
    - ~~How are the AAs selected? And are they selected by the user - in which case, the origin AA of each attribute needs to be tracked so the RP can see where the attribute came from, and decide whether to trust it.~~
    - ~~Harking back to AARC1 discussions, when and how is the choice of AA made? When the service needs extra attributes and the user needs to “go out and find them” or initially when the user authenticates (e.g. to a proxy.) [this discussion should be analogous to “step up”?]~~
- Combining attributes
    - Keeping track of the origin of the attribute
    - Interpretation of combined attributes with overlapping semantics

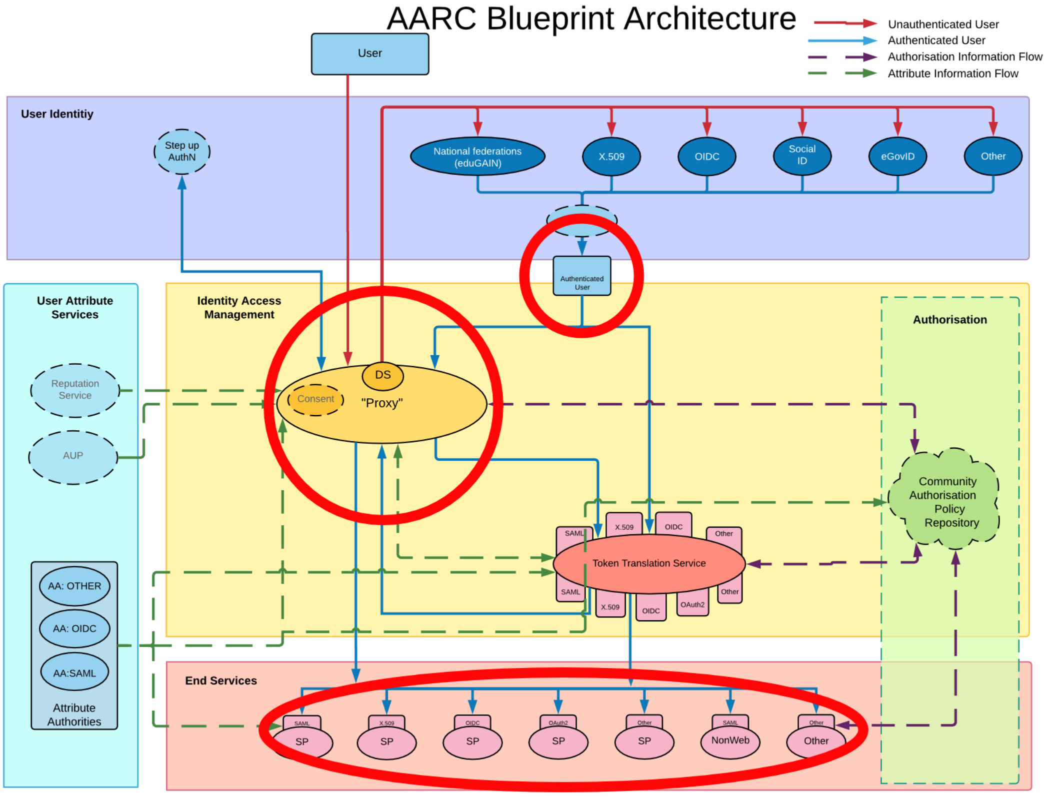

Figure XXX shows the typical locations of attribute aggregation in current deployments, mapped onto the BPA (in order from top to bottom):

- The (already authenticated) user collecting attributes from different AAs (e.g. VOMS)
- The proxy aggregates attributes (e.g. EUDAT B2ACCESS, EGI (through PERUN))
- The SP aggregates attributes (e.g. e-commerce or social media, with traditional session-cookie setting where the session is the key for a service-based lookup of the stored attributes)

Since AARC recommends that infrastructures use a proxy, we shall focus on the proxy in this document. For infrastructures not using a proxy, the same processes apply, but at the other points indicated in the architecture diagram.

Prior and Related Work

This document builds on the following earlier documents, which the reader [and the writer!] should consult.

First, a fundamental assumption is that the user is *authenticated* before they can ask for or obtain authorisation attributes/tokens (cf. DJRA1.2, section 4.1).

From AARC1-JRA1.4B, we import in particular the concepts of:

- Attribute push/pull models
  - Attribute aggregation SHOULD NOT be done by SP
- The need for attribute metadata:
  - User's consent to use

    - User’s visibility of the use of attributes
    - Expiry
    - Freshness
    - Linking records through persistent and unique identifiers
    - Attribute scope
    - Harmonisation and consolidation into the smallest useful vocabulary

AARC1-JRA1.4A-201710 describes guidelines for the expression of memberships and roles. As far as the present document is concerned, the main points are:

- If there is a need for harmonisation of attributes, then this should be done in a proxy and not by SPs
- Ensure attributes are scoped (when they reach the SP)
    - Use of eduPersonScopedAffiliation to express affiliation within research infrastructure
    - Mapping of eduPersonScopedAffiliation to voPersonExternalAffiliation to express affiliation within Home Organisation(s)

AARC-G009 (aka AARC1-JRA1.4H) discusses account linking and LoA elevation. This document had slightly different use cases for account linking, but is similar in the sense that (multi-valued) attributes can be aggregated, and a specific selection process needs to be applied given competing values for single-valued attributes. In particular, the choice may not be known at linking-time (or when the user authenticates to the proxy), but may have to be decided upon accessing the service (AARC-G009, section 4).

From AARC1-MJRA1.3 Design for the integration of an attribute management tool we note the distinction between push and pull; in the former the user chooses which attributes to send (resp., the proxy), and in the latter, the service fetches the attributes it needs.

(AARC2-DJRA1.2 - how authorisation is actually done, distilled)

Example work: EUDAT attribute LoA (background study
http://doi.org/10.23728/b2share.20c1c0c8ba254e768fbcb67724918936). EGI?
OGF-VOMSPROC

Using NIST IR 8112: provenance of attributes. EUDAT implementation.

Schemata (move down?)

eduPerson , … voPerson
(what’s missing in voPerson? - open to collaboration, open issue to request change
voPersonExternalAffiliation?)

Projects with multi-attribute authority requirements (use cases, why are we doing this);

ELIXIR - organisational affiliation [LSAAI is not a good example]
DAFNI - organisational affiliation
Two communities asserting roles about the same person
CheckIn / COmanage, GOCDB (site roles); VOMS example
eduTeams-EUDAT (eduTeams as a community proxy - does it have the same problem?) - may not be in scope, because it’s a single AA. So is LS AAI (using PERUN). -> Check with Arnout.
B2ACCESS is the proxy service for the EUDAT infrastructure.
Access to sensitive human data in ELIXIR
ELIXIR developed a solution to control access to sensitive data in distributed environments, which is based on rules managed by dedicated authorities. Access to a particular dataset is granted through the REMS service and is maintained by a “permission source” that is responsible for the dataset. Permission sources (e.g. European Genome-phenome Archive, EGA) issue data access permissions that are used by the policy enforcement points at the systems (for instance, private computing clouds). Permissions of a user are delivered as signed JWT tokens using the OAuth2 authorization server that is provided as part of ELIXIR AAI. The intention is that only one integration is needed both for each permission source and each PEP. For more details see https://docs.google.com/document/d/1rqCD75HRA99HKwq0s-OkWWBKiaEo2JO-_MYjlsyiSQ4/edit#

DARIAH (David)

In the DARIAH AAI some attributes are mandatory (e.g. givenName, surname, email). While some of these attributes technically are multi-valued, in a lot of cases it is not desired to actually release multiple values to a service (e.g. a service usually is only interested in one email address and might actually have technical limitations to deal with a multi-value email address). This also holds true for single-valued attributes.
A user can either register a DARIAH account directly or authenticate via his home organisation’s IdP via the DARIAH proxy. In the former case there is only one source for the attributes and the proxy will never be in a situation, where it needs to decide between multiple values. However, in the latter case, there potentially are two sources of attribute values. There are two scenarios where this could lead to problems:

- The user registers via his home organisation’s IdP. This IdP releases the R&S attribute bundle and the attributes are pre-filled in the registration form for the DARIAH account for convenience. The DARIAH AAI allows a user to change most of the attributes in a self service portal. The user might change some of these attributes (e.g. email , because he wants to receive emails regarding DARIAH on a seperate account). In this case there will be a conflict at the proxy, which still receives the old email from thedolph IdP used for authentication and the new email from the DARIAH AA.

- A similar conflict could be created by the home organisation’s IdP changing one of the released attributes (e.g. surname after marriage or email, because the old email is no longer valid). This could happen without a deliberate action of the user.

In both scenarios the proxy might be in a situation, in which it has to decide between two conflicting attribute values. Without additional information about the attributes (e.g. metadata about assurance, modification date, ...) the proxy can either rely on simple heuristics (e.g. always trust the DARIAH AA as authoritative source of the values) or prompt the user everytime a conflict is detected. The former option will not always be accurate (consider email, where the user might have deliberately changed his email or the old email is actually no longer valid). The latter option might be too intrusive. AARC-G009 gives some guidelines about this issues, but ultimately states, that this “needs to be decided by the infrastructure”. However, with the increasing number of proxies and account linking, some more concrete guidance could be justified.

EGI Check-in can combine information from different attribute authorities providing assertions about the user’s:

- VO/group membership information maintained, for example, in COmanage Registry, Perun and/or VOMS
- role(s) when operating EGI Resource Centres (also named "sites")

Both types of information are encapsulated in URN-formatted entitlement values, which are incorporated into the original SAML attribute assertion/OIDC claims sent by the user’s IdP before being passed on to the relying party where they are commonly used for authorisation purposes.

How attributes are combined in the proxy

- Metadata says what the origin is
- Scope it and aggregate it
- Select most trusted, *user-preferred* attribute (e.g. email address?)
    - SP expects single attribute, but the proxy may have it from multiple sources
    - LS pilot - group membership (DJRA1.2?)
- Communicating all attributes vs user selected vs those required by the particular access request

The NIST IR 8112 discusses the "currency metadata" of an attribute, in this example affiliation. There are three values, when the attribute was verified (LastVerified), when it was refreshed (LastRefresh), and when it "expires" (ExpirationDate). In the context of affiliation,

How (meta)attributes are communicated to the SP (or rather authorisation systems?)

Proxy can communicate single attribute (= easy).

Some attributes already have meta-attributes, like email verified. Some attributes are multi-valued in SAML. (AARC / voPerson / OGF / NIST Namespace with XML markup? - SAML attribute combines XML DIGSIG, XMLENC, SAML, + new namespace for metadata) Expression of attributes in XACML policy?

How meta-attributes are expressed in SAML and OIDC?
X.509 using GSI and RFC 3281 attribute certificates

Guidance

In the following guidance, unless stated explicitly, we discuss attributes abstractly, i.e. "name" or "role in community X", as opposed to the specific attribute such as eduPersonPrincipalName or commonName. For example, several specific attributes can carry the user name, such as givenName and sn (surname), commonName, displayName, etc. If several such are used, they SHOULD be consistent.

It is assumed that services in a given infrastructure need certain attributes to authorise authenticated users. In general, these attributes can be sourced from multiple sources: some from the IdP, additional attributes (possibly none) from the proxy, yet other (possibly none) from other attribute authorities. These other attribute authorities, if used, can be run by the communities, by the infrastructure that the user is trying to access, or by other infrastructures.

NIST IR8112 distinguishes between attribute schema metadata (ASM) and attribute value metadata (AVM). ASM is not about the schema specifically but about the (abstract) attribute. For example, ASM would include frequency with which the attribute is to be verified, and AVM the timestamp of the most recent verification.

Who can modify attribute?
If an attribute is unique, in the sense that it is asserted by a single source, the issues are:

- User's intent to assert the attribute
- User's consent to release the attribute
- Attribute metadata
  - Identification of the authority for the attribute
  - Freshness of attribute or frequency of verification

    - Allowed values

If an attribute is published by multiple sources,

Usability Issues

The flow for the proxy getting attributes

OIDC flow in EUDAT/GEANT (eduTeams->B2ACCESS)

B2ACCESS is the authentication and authorisation service for the EUDAT infrastructure. It is a single attribute authority and acts as a “fat” proxy: it is being deployed as an authentication gateway and also manages the user provisioning and their attributes. The attributes that are released by the B2ACCESS service to the downstream EUDAT services are the EUDAT scoped persistent identifier, email and common name. As the attribute set does not contain essential information (such as group or entitlement attributes) to make authorisation or accounting. Given that each service has to define such attributes internally. In order to provide accounting and authorisation attributes to the downstream services, B2ACCESS has been integrated with the GEANT's eduTEAMS service, which is an identity hub, group / membership management system and enables management of virtual research teams. eduTEAMS being an attribute authority within the EUDAT CDI, manages data project, service and resource identifiers for each EUDAT user in the infrastructure and released to the end service through B2ACCESS, which merges and persists the eduTEAMS and EUDAT specific attributes. Furthermore, the B2ACCESS and eduTEAMS are connected through OIDC protocol, whereby the latter is the authorisation server and using the code grant flow.

Push vs pull: proxy may get attributes pushed to it and

voPerson is both a set of recommendations and an ldap attribute schema (object class), intended to provide a common reference point for attribute management within a Virtual Organization (VO)

- Using the scope-*[1] label for attributes that are implicitly scoped based on their source. The suffix of the option (ie: anything after scope-) is a label describing the scope or source of the value.
- Using the role-*[2] label for correlating multi-valued attributes. Any value labeled with the same role label should be considered belonging to the same source.

Brief Use Cases (keeping it real…) - cf. NIST use cases

Use case should describe (a) what happens in the proxy and (b) how the authorisation is made on the information.

1. Expression of 'memberOf' - single format, scoped
2. Single service needs attributes from xple attribute authorities
    a. May be outside the infrastructure - not processed by the proxy
    b. From outside the infrastructure - mapped/processed by the proxy
    c. From different sources inside the infrastructure(?)
3. Service needs LoA of attrbute used for authorisation -
    a. e.g. assertion of affiliation (who asserted it, self asserted or org.)
    b. (and controlled value domain cf. DAFNI project)
4. and LS Pilot - community managed information

User Choice

[Out of scope? Cf. R&S Discussion]
Users choose group/role and/or project affiliation

- VOMS - authenticated user "decorating" credential
- Proxy presents information ot the user so they may select which attributes to present - front end or back end
    - LS Pilot - proxy lets user select attributes to present

BBMRI has a quite strict requirement on controlling access to dataset they process when it comes to multiple roles. A user needs to be granted to work with the dataset (which AFAIK is determined by a membership in a project that processes the data). Users are expected to

[1] https://github.com/voperson/voperson/blob/master/voPerson.md#scope--attribute-description-option
[2] https://github.com/voperson/voperson/blob/master/voPerson.md#role--attribute-description-option

use their permissions exclusively, i.e. even if a user is member of multiple projects with different access rights, they should not aggregate the access rights and always work with a single role/project. The arrangement has some impacts on how the choice is performed and where (Proxy/IdP and/or end-service).

IIRC, Openstack has a similar requirement actually (imposed by technical aspects not policy), requiring that the user always select one group/project they want to use when dealing with the VMs.