23-05-2017

Deliverable DJRA1.2: AARC Blueprint Architectures

Deliverable DJRA1.2

Contractual Date:	30-04-2017
Actual Date:	23-05-2017
Grant Agreement No.:	653965
Work Package:	JRA1
Task Item:	Task 2
Lead Partner:	KIT/GÉANT
Document Code:	DJRA1.2
Authors:	Christos Kanellopoulos (GÉANT), Uros Stevanovic (KIT), Marcus Hardt (KIT) and the rest of the JRA1 team

The research leading to these results has received funding from the European Community's Horizon2020 Programme under Grant Agreement No. 653965 (AARC).

Abstract

This document provides an overview of the two iterations of the AARC blueprint architecture, which have been incrementally developed during the two years of the project, and the associated guidelines and best practices. The AARC blueprint architecture provides a set of building blocks for software architects and technical decision makers who are designing and implementing access management solutions for international research collaborations.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The AARC blueprint architecture provides a set of building blocks for software architects and technical decision makers who are designing and implementing access management solutions for international research collaborations on top of eduGAIN. During the last two years, AARC has provided two iterations of the blueprint architecture, along with a set of guidelines and best practice documents for key areas.

In Year 1 of the project, AARC published the first iteration of the blueprint architecture, named AARC-BPA-2016 [AARC-BPA-2016]. It was informed by an analysis of the requirements of several communities and infrastructure providers (documented in [AARC-DJRA1.1]) and discussions of existing implementations and designs with implementers of nascent access management solutions in the context of international research collaborations. Defining four layers – User Identities, Attribute Enrichment, Translation and End Services – AARC-BPA-2016 laid the groundwork for the implementation of interoperable authentication and authorisation infrastructure (AAI) solutions by the research and e-infrastructures and has been widely accepted.

In Year 2 of the project, AARC published the second iteration of the blueprint architecture, AARC-BPA-2017 [AARC-BPA-2017]. This was an incremental version, building on AARC-BPA-2016, and driven by the early experiences of and feedback received from the adopters of the first iteration of the AARC blueprint architecture. The IdP/SP proxy model, introduced in AARC-BPA-2016, was further developed in AARC-BPA-2017 and it is now the basis of the blueprint architecture. In AARC-BPA-2017, the architectural layers were repositioned, to make their interactions and dependencies clearer to the reader; a new layer for authorisation was introduced; and more components and details for each of the pre-existing layers were provided. In addition, two of those layers were renamed: Attribute Enrichment became User Attribute Services, and Translation became Identity Access Management.

AARC-BPA-2017 is accompanied by a set of support documents:

- Guidelines on expressing group membership and role information in a consistent manner across research infrastructures / e-infrastructures [AARC-JRA1.4A].
- Guidelines on scalable attribute aggregation implementations [AARC-JRA1.4B].
- Guidelines on the implementation of credential translation via token translation services [AARC-JRA1.4C].
- Implementation scenarios and guidelines for credential delegation [AARC-JRA1.4D].
- Best practices for managing authorisation, specifically targeting practices for community-based authorisation [AARC-JRA1.4E].
- Implementation scenarios and guidelines for non-browser access [AARC-JRA1.4F].
- Guidelines for implementing SAML authentication proxies for social media IdPs [AARC-JRA1.4G].

- Use-case scenarios for account linking and level of assurance elevation via step-up authentication [AARC-JRA1.4H].
- Best practices and recommendations for attribute translation from federated authentication to X.509 credentials [AARC-JRA1.4I].

Each of these is summarised in this document.

While work on the AARC blueprint architecture will continue in AARC2, it has already been adopted by e-infrastructure providers and research infrastructures, including EGI, ELIXIR, EUDAT, GÉANT and INDIGO.

The AARC blueprint architecture, along with the guidelines for the implementers, can be found on the AARC website [AARC-BPA-Web].

1 Introduction

1.1 Background

The way researchers collaborate can vary significantly between the different scientific communities. Some are highly structured, with thousands of researchers who could be located virtually anywhere in the world. Typically, these are communities that have been working together for a long time, that want to share and have access to a wide range of resources, and have had to put in place practical solutions to make the collaborations work. On the other hand, there are also a number of smaller, more diverse research communities working within specific or across multiple scientific disciplines. Typically, these are either nascent communities being established around new scientific domains, or communities in specific domains that do not need to promote widespread and close collaboration among researchers. In between these two extremes are scientific communities of all varieties in terms of size, structure, history, etc.

During the last two years, the AARC project [AARC] has been working together with e-infrastructures, research infrastructures, research communities, AAI architects, and implementers to get a better understanding of their experiences and needs regarding sharing and accessing resources within research collaborations. The goal has been to collectively define a set of architectural building blocks and implementation patterns, the "AARC blueprint architecture", that will allow the development of interoperable technical solutions for international intra- and inter-disciplinary research collaborations.

The first version of the AARC blueprint architecture, named AARC-BPA-2016, was published in July 2016 [AARC-BPA-2016]. In that document, the authors analysed the architectures of existing designs and implementations and extracted a high-level architecture that encapsulated common patterns and building blocks. The second version, named AARC-BPA-2017, was published in April 2017 [AARC-BPA-2017] and built upon the first iteration. AARC-BPA-2017 extended the previous version and provided guidance on topics such as access to non-web based services, token translation services (TTSs), best practices for managing authorisation, harmonised expression of group membership and role information, attribute aggregation and credential delegation.

1.2 In this Document

This document is organised as follows:

- Section 2 provides an overview of the two versions of the AARC blueprint architecture, AARC-BPA-2016 and AARC-BPA-2017. The first version is presented only briefly, as it has been thoroughly described in *MJRA1.4: First Draft of the Blueprint Architecture* [AARC-BPA-2016]. This deliverable then focuses on

AARC-BPA-2017, beginning with a summary of the changes from AARC-BPA-2016, including a list of the new guidelines and best practices. Section 2 also outlines plans for subsequent iterations of the blueprint architecture.

The deliverable then presents extracts from the set of guidelines and best practices that accompany AARC-BPA-2017 (the shorter documents are grouped together in Section 3; the longer, more complex documents each have a section of their own):

- Section 3 covers expressing group membership and role information; attribute aggregation; token translation services; and implementing Security Assertion Markup Language (SAML) authentication proxies for social media identity providers (IdPs).
- Section 4 addresses managing authorisation in research infrastructures / e-infrastructures (RIs/EIs) leveraging federated access.
- Section 5 considers access to non-web services.
- Section 6 covers credential delegation.
- Section 7 considers account linking and level of assurance (LoA) elevation.
- To conclude, Section 8 offers an overall assessment of the blueprint architecture work.

2 AARC Blueprint Architecture

As was described in the milestone document *MJRA1.4: First Draft of the Blueprint Architecture* [AARC-BPA-2016], research infrastructures (RIs) and e-infrastructures (EIs) can already rely on eduGAIN [eduGAIN] and the underlying identity federations to authenticate their users. Figure 2.1 depicts the standard approach, in which different services are made available via eduGAIN through a participating federation. The top part of the figure (almond colour) shows the example of a "full mesh federation", while the lower part (green, blue and grey) shows a "hub-and-spoke" federation. Figure 2.2 shows an RI or EI that is connected to eduGAIN via a single service provider (SP), which thus acts like an SP-identity provider (IdP) proxy. The SP-IdP proxy component can augment attributes from the authenticator by introducing elements that are essential for the RIs/EIs, such as persistent, unique, non-reassignable identifiers, differentiated levels of assurance, community-managed access control based on group membership and community roles, etc. In this way, the RIs/EIs shield themselves from the heterogeneity of the global R&E federation space and are able to implement flexible and scalable access solutions that encompass federated access.

The purpose of the AARC blueprint architecture (BPA) is to provide a set of interoperable architectural building blocks for software architects and technical decision makers who are designing and implementing access management solutions for international research collaborations.

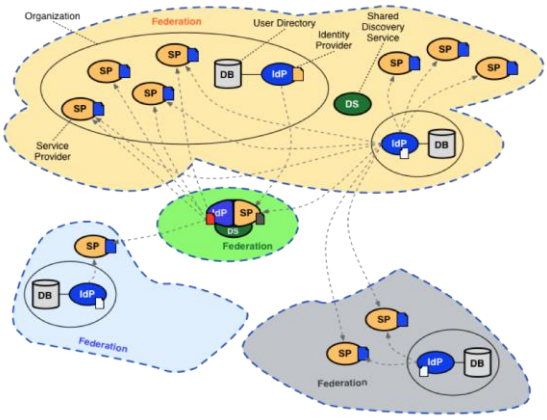

Figure 2.1: SAML inter-federation, as provided (for example) by eduGAIN

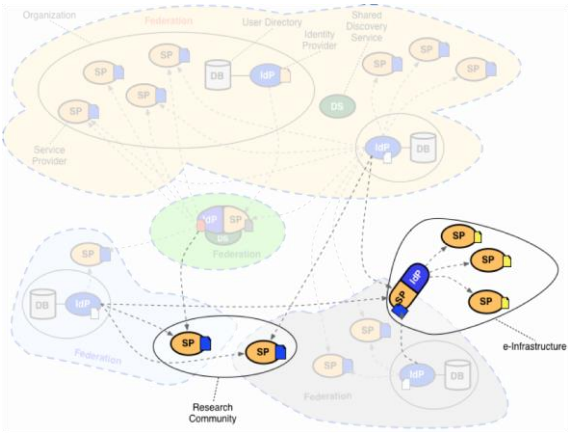

Figure 2.2: Examples of a research community and an e-infrastructure in eduGAIN

The deliverable *DJRA1.1: Analysis of user community and service provider requirements* [AARC-DRJA1.1] presents an analysis of the use cases and needs of the user communities and infrastructure providers and contains a detailed list of requirements. A summary of those requirements is shown in the matrix below.

Attribute Release	Attribute Aggregation	User Friendliness	SP Friendliness
Persistent Unique Id	Credential translation	Credential Delegation	User Managed Information
Levels of Assurance	Guest users	Step-up AuthN	Best Practices
Community based AuthZ	Non-web-browser	Social & e-Gov IDs	Incident Response

Figure 2.3: Requirements matrix

2.1 AARC-BPA-2016

As shown in Figure 2.4 below, the first version of the AARC blueprint architecture, AARC-BPA-2016, defines four layers, namely: User Identities, Attribute Enrichment, Translation, and End Services. Each layer contains one or more components. The figure is not a strict representation of deployment scenarios. Users and SPs are likely to be situated in a different organisational domain and even different countries; identities could be provided by services different from those that provide other (non-identity) attributes.

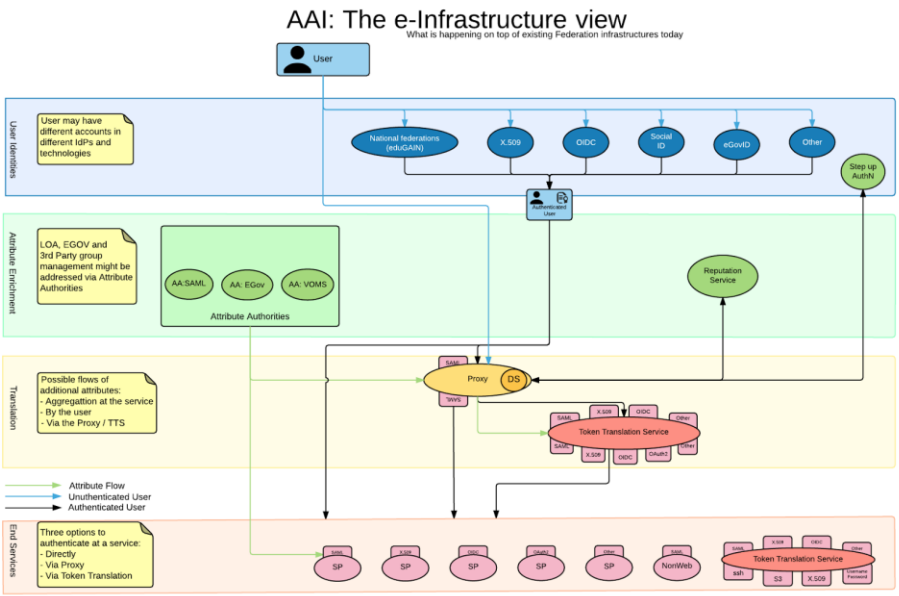

Figure 2.4: AARC blueprint architecture 2016

The **User Identities Layer** contains services for identification and authentication of users. In existing implementations in the research and education space, these services typically include Security Assertion Markup Language (SAML) identity providers, certification authorities and, more recently, OpenID Connect (OIDC) Providers (OPs). Although the focus of the services in this layer is to provide user authentication, often some end-user profile information is released as part of the authentication process.

The **Attribute Enrichment Layer** groups services related to managing and providing information (attributes) about users. Typically, they provide additional information about the users, such as group memberships and community roles, on top of the information that might be provided by services from the User Identities Layer. Services like these exist for all the authentication technologies mentioned above. The Virtual Organisation Membership Service (VOMS) is commonly used in X.509-based infrastructures, attribute authorities (AAs) in SAML-based implementations, and the "userinfo" endpoint in OpenID Connect implementations. This document uses SAML AA terminology.

The **Translation Layer** addresses the requirement for supporting multiple authentication technologies. The two types of services most often encountered in existing implementations are:

- **Token Translation Services**, which translate identity tokens between different technologies. Token translation can be implemented as a central service or offered at an SP's site.
- **SP-IdP-Proxy (proxy)**, which is an emerging pattern within research and e-infrastructures. This model is depicted in Figure 2.2. It is predominantly found in SAML installations. Towards the Identity Federations this proxy looks like any other SP, while towards the internal SPs it acts as an IdP.

The **End Services Layer** contains the services users want to use. Access to these services is AAI-protected (possibly using different technologies). These services can range from simple web services, such as wikis or portals for accessing computing and storage resources, to non-web resources such as a login shell, an FTP transfer or a workload management system.

This initial draft version of the AARC blueprint architecture addressed a subset of the requirements coming from the user communities and infrastructure providers, shown in blue in Figure 2.5.

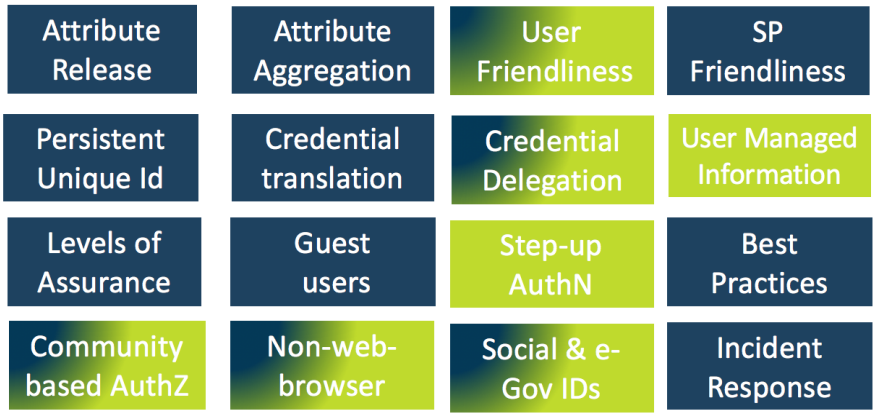

Figure 2.5: Requirements covered in AARC-BPA-2016

The requirements coloured blue-green were being actively worked on at the time the draft was published, and those shown in green would be addressed in the next iterations of the AARC blueprint architecture.

2.2 AARC-BPA-2017

The second version of the AARC blueprint architecture, AARC-BPA-2017, builds upon the previous one and provides a more detailed layered architecture, while retaining full backwards compatibility. As shown in Figure 2.6, AARC-BPA-2017 retains the same four layers, each of which includes one or more functional components, grouped by their complementary functional roles. The User Identities Layer and the End Services Layer are still there, while the Attribute Enrichment Layer has been renamed to User Attribute Services Layer and the Translation Layer has been renamed to Identity Access Management (IAM) Layer and has a prominent role in the architecture. AARC-BPA-2017 introduces a new layer for the centralised Authorisation.

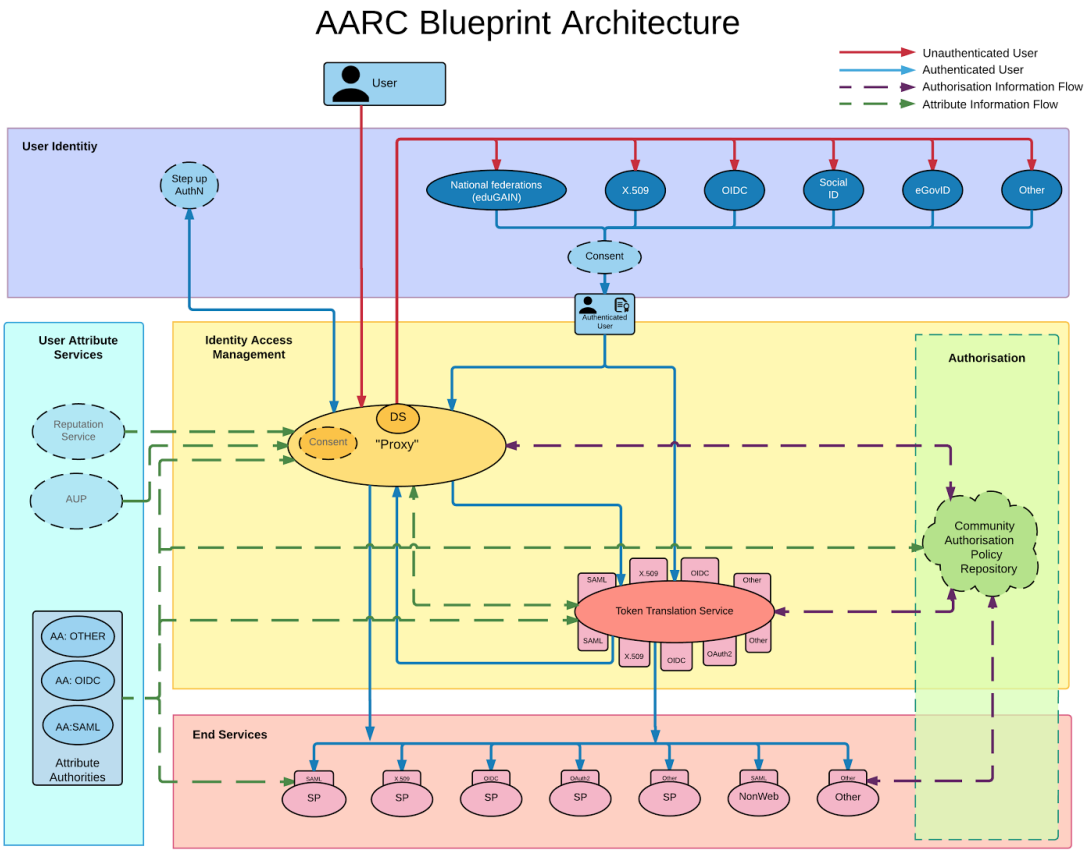

Figure 2.6: AARC Blueprint Architecture 2017

The **User Identity Layer** includes services which provide electronic identities that can be used by users participating in international research collaborations. Typically, those identity services are themselves outside of the administrative boundaries of the international research collaborations. Services in the layer are expected to use secure authentication mechanisms, to bind physical persons to their electronic identities, which are then made available to services in the other layers via secure protocols.

The AARC blueprint architecture is technology agnostic by design and has been verified against production implementations that use SAML 2.0, OpenID Connect, OAuth2, or combinations of these. Identity service providers might utilise single-factor or multi-factor authentication schemes.

AARC-BPA-2016 also incorporated the traditional flows, which directly linked service and identity providers, to denote the different possibilities and the change towards proxied implementations. AARC-BPA-2017 has kept only the flows relevant for the proxied architecture and thus users are shown to access services only via the RI/EI proxy component. This version of the architecture also introduces the points where user consent is expected.

The **Identity Access Management Layer** is a new layer, introduced in AARC-BPA-2017, which replaces the Translation Layer. The components in the Identity Access Management Layer are operated by (or on behalf of)

the RI/EI and thus reside within the administrative and policy boundaries of the RI/EI. The Identity Access Management Layer defines an administrative, policy and technical boundary between the internal services and resources of the RI/EI and any other external services and resources. In AARC-BPA-2016, this layer was marked as an optional layer, but in AARC-BPA-2017, the Identity Access Management Layer is an integral part of the architecture as its functionality is required for all the use cases that go beyond the basic web single sign-on (SSO) scenarios.

The components in this layer allow the RIs/EIs to (a) take full advantage of eduGAIN and the national identity federations, (b) reduce the administrative overhead of introducing new services and (c) have the flexibility to choose the appropriate security protocols and mechanisms for delivering internal services to their users[1]. Furthermore, the Identity Access Management Layer enables the implementation of a single point where the RI/EI can provide an IdP discovery service for all its internal services, along with other required functionalities, such as integration with community-based group management systems and consistent, scalable central management of user consent. Finally, it enables the infrastructure to provide guarantees that may not be met by the external IdPs alone, such as unique, persistent identifiers, multi-level-of-assurance (LoA) management, infrastructure-specific attributes, account linking, etc.

The Attribute Enrichment Layer has been renamed to **User Attribute Services Layer,** and it groups components related to managing and providing information (attributes) about users, such as group memberships and community roles, on top of the information that might be provided directly by the identity providers from the User Identity Layer. In the AARC-BPA-2017 architecture diagram (Figure 2.6), this layer has been moved to the left of the Identity Access Management and End Service layers to make clearer the interactions between components of this layer and the layers of Identity Access Management, Authorisation and End Services. In contrast with the proxy-managed attributes, responsibility for managing attributes provided by these services rests with the user communities[2].

Note also the presence of two components with dotted borders, the Acceptable Use Policy (AUP) and Reputation attribute services. The former denotes a service that specifically records whether the user has accepted the AUP of the infrastructure (as required by many NRENs, RIs, EIs, and by Sirtfi [SIRTFI]). The latter is a service that records the user's "reputation", as discussed in the milestone document *MJRA1.2: Design for Deploying Solutions for "Guest Identities"* [AARC-MJRA1.2]. These components will be further analysed in the next versions of the AARC BPA in AARC2.

[1] This flexibility is important because projects are encouraged to reuse existing software rather than re-implement from scratch, but the existing software components may use different AAI technologies.

[2] There may be a need for attribute scoping, harmonisation, or even translation, such that attributes from different communities are, as a minimum, not confused with each other. Further work in this area is expected to take place in AARC2.

The **Authorisation Layer** is a new layer introduced in AARC-BPA-2017. Authorisation of access to services can take place in many ways. Typically, in RIs/EIs, authorisation can be based on: (a) the group membership of the users, (b) the roles a user might have been granted within the collaboration, (c) the entitlements users might have been granted, (d) the affiliations of the users, (e) the strength of the authentication method used or the quality of the user information, or (f) combinations of these.

Although authorisation enforcement always happens on the service side, the AARC BPA allows the implementers to delegate many of the complex decisions to central components, which can significantly reduce the complexity of managing authorisation policies, and their evaluation to each service individually. For example, the decision as to whether a user can access a specific service can be taken centrally and then communicated to the service by adding a service-specific attribute to the user's attributes. In this way, a service can rely on the infrastructure to make the authorisation decision based on several appropriate factors. Authorisation is a topic that will be thoroughly analysed in the next versions of the AARC BPA and it is a key topic in the Architecture work package of AARC2.

As in AARC-BPA-2016, the **End Services Layer** contains the services users want to use. Access to these services is AAI-protected (possibly using different technologies). These services can range from simple web services, such as wikis or portals for accessing computing and storage resources, to non-web resources such as a login shell, an FTP transfer or a workload management system. A notable change in AARC-BPA-2017 is the removal of the Token Translation Services from the End Services Layer. Credential translation or token translation can happen centrally and/or within a service, but the latter is outside the scope of the AARC blueprint architecture.

AARC-BPA-2017 addresses most of the requirements that were still open in AARC-BPA-2016. Namely, AARC-BPA-2017 provides:

- Guidelines on expressing group membership and role information in a consistent manner across RIs/EIs [AARC-JRA1.4A].
- Guidelines on scalable attribute aggregation implementations [AARC-JRA1.4B].
- Guidelines on the implementation of credential translation via token translation services [AARC-JRA1.4C].
- Implementation scenarios and guidelines for credential delegation [AARC-JRA1.4D].
- Best practices for managing authorisation, specifically targeting practices for community-based authorisation [AARC-JRA1.4E].
- Implementation scenarios and guidelines for non-browser access [AARC-JRA1.4F].
- Guidelines for implementing SAML authentication proxies for social media IdPs [AARC-JRA1.4G].
- Use-case scenarios for account linking and LoA elevation via step-up authentication [AARC-JRA1.4H].
- Best practices and recommendations for attribute translation from federated authentication to X.509 credentials [AARC-JRA1.4I].

These are summarised in sections 3 to 7 below.

Figure 2.7 below shows the progression of requirements addressed from AARC-BPA-2016 to AARC-BPA-2017.

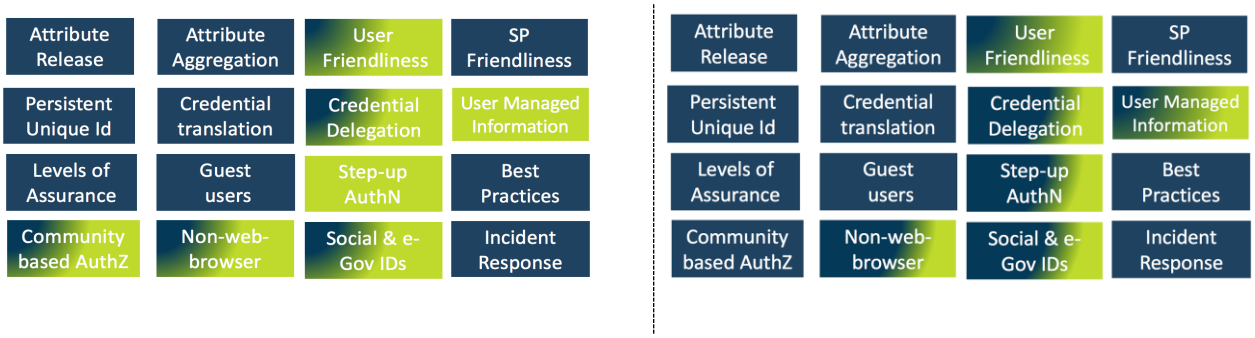

Figure 2.7: Progression of the requirements covered from AARC-BPA-2016 (left) to AARC-BPA-2017 (right)

The requirements shown in blue have been addressed by the AARC BPA. Those coloured blue-green have been partially addressed, but there is still work to be done. Those shown in green will be addressed in the next iterations of the AARC blueprint architecture.

2.3 Next Iterations of the AARC BPA

AARC2 will continue the work on the blueprint architecture. As two years have passed since the initial requirements capture, the project will collect updated feedback and requirements from RIs/EIs, service providers and the scientific communities about cross-infrastructure interoperability, taking into consideration (a) the AAI technologies enabled in the different infrastructures and how the components fit in the architecture developed in AARC, and (b) the authorisation mechanisms integrated with the services and the sources of authorisation information operated by the communities.

The next iterations of the AARC BPA will:

- Address aspects relating to the integration of the blueprint architecture and its components into the existing AAIs.
- Explore tools and services for interoperable infrastructures and integrate additional technical components into the AAI design to support a wider range of use cases than to date.
- Explore service provider architectures and authorisation in multi-SP environments.
- Provide models for the evolution of the AAIs for research collaborations, ensuring cross-sector interoperation.
- Provide guidelines for scalable VO platforms.

3 Guidelines and Best Practices for AARC-BPA-2017

AARC-BPA-2017 is accompanied by a set of guidelines and best practice documents, which are intended to help RIs/EIs to implement interoperable AAIs. The following subsections and sections contain extracts from the core parts of each guideline and best practice document. The full documents can be found on the AARC BPA website home page [AARC-BPA-Web].

This section contains guidelines on:

- Expressing group membership and role information.
- Attribute aggregation.
- Token translation services.
- Implementing SAML authentication proxies for social media IdPs.

3.1 Guidelines on Expressing Group Membership and Role Information

The guidelines presented in this subsection (documented in full in [AARC-JRA1.4A]) have been defined based on experiences from multiple parties in the AARC project and have subsequently been discussed and tested through the Service Activity 1 Pilots (SA1) attribute management pilot [AARC-SA1-AMP]. Furthermore, it should be noted that a group membership representation scheme following these recommendations has already been adopted to enable cross-infrastructure exchange of group information between the EGI and the ELIXIR AAI.

- Centralised harmonisation of group membership information

  Adopt a proxy-based AAI, to delegate to the proxy component the complexity of dealing with different group membership representations that originate from diverse IdPs/AAs. As a result, the end SPs will not have to handle the harmonisation of group membership information as this will be performed in a centralised fashion by the SP proxy.

- Compatibility with existing group information models

  Adopt a group representation scheme that can be easily translated to/from standardised or widely used group data models, such as SCIM, VOOT or VOMS, and POSIX systems, if required.

- Scoping of group membership information

Specify the scopes where the identified group membership information is valid. These scopes should include:

- The authoritative source for each piece of group membership information.
- The VO associated with the identified group.
- The entire chain of group components, from the root parent group to the identified child group (in the case of group hierarchies).

The rationale behind scoping is to prevent clashes between groups that are managed by different VOs/administrative domains. This eliminates the need for syntactic and semantic group information harmonisation among different communities. An added benefit is that scoping allows easy filtering of group values that can be used by SPs for quick authorisation decisions.

- Use the eduPersonEntitlement attribute

  When using SAML, different standardised possibilities are available to convey group membership information. Specifically, both the isMemberOf [SWITCH-IMO] and the eduPersonEntitlement attribute [I2-EPE] can be used for representing group membership. However, eduPersonEntitlement values (formatted as URIs, either URNs or URLs) are, in addition, used to indicate rights to resources. In the case of OpenID Connect there is currently no standard claim to carry group membership information. However, the REFEDS OpenID Connect for Research and Education Working Group [OIDCre] is already investigating the standardisation of new claims for expressing the attributes defined in the eduPerson schema [I2-EP].

  It should be noted that while eduPersonEntitlement is not part of the REFEDS "Research and Scholarship" (R&S) [REFEDS-RS]) attribute bundle, an SP may request it if necessary [REFEDS-RS-1], without violating compliance with the R&S entity category. However, SPs are still encouraged to stick to the R&S bundle wherever possible.

- Use of valid URIs, either URLs or URNs, for representing group membership information

  As of 2015, MACE [MACE] encourages the use of URLs in preference to URNs [MACE-SR].

  Benefits of using URLs instead of URNs include:

  - Legitimate URL values are globally unique if a suitable (sub)domain is used and a delegation model is in place for defining paths under that root domain. No one else has the legal right to create values under that (sub)domain, so any assignments made under that subdomain will be globally unique.
  - If the URLs resolve to web pages, it is possible to make the assigned values self-documenting by posting a definition of the value at that URL.

    In practice, however, the relevant domain that is used for resolvable URLs is often the domain of corporate public relations departments and as such is not easily maintainable by technical staff responsible for the AAI.

  - URLs do not require a formal registration for a subtree, as is required for URNs.

  Benefits of using URNs instead of URLs include:

- URNs are currently more commonly used for expressing eduPersonEntitlement values by existing IdPs/AAs/federations.
- URNs can easily support scoping following a hierarchical structure when necessary. Using the namespace identifier registry delegation model, URN values can thus be managed in a distributed fashion by different issuing authorities, communities/VOs, group management systems.

The use of URLs and URNs each has its merits. Given the wide adoption of URNs, however, these guidelines suggest the use of URNs for expressing group and/or role membership. An e-infrastructure, research infrastructure or research collaboration could adopt the following eduPersonEntitlement formatting specification for representing group membership information:

urn:mace:<namespace>:<authority>:group:<group>[:<subgroup>*][:role=<role>]

where:

- <namespace> is a registered URN namespace ensuring global uniqueness
- <authority> is the FQDN of the authoritative source for the entitlement value
- the literal string “group” indicates an eduPersonEntitlement value expressing group membership information
- <group> is the name of a Virtual Organisation (VO), research collaboration or a top-level arbitrary group
- an optional list of <subgroup> components represents the hierarchy of subgroups in the <group>
- the optional <role> component is scoped to the rightmost (sub)group; if no subgroup information is specified, the role applies to the top level group/VO

3.2 Guidelines on Attribute Aggregation

A user’s home institution IdP can provide attribute information to the relying party/service provider he/she is accessing. However, many federated IdPs will not send enough information to meet the requirements of all RPs/SPs. Some IdP operators will not release the required information: they may have data protection concerns, restrictive policies or just slow procedures. Other IdP operators may not have the information at all, and are unable or unwilling to create or manage it. Research collaborations may have their own data for users and groups that they wish to use alongside federated authentication. It is common for VOs to create their own group and entitlement information for access control and management. This subsection presents attribute aggregation guidelines that can be applied in international research collaborations. (The guidelines are documented in full in [AARC-JRA1.4B].) Attribute aggregation can take place at proxy, SP or TTS services, in line with the AARC BPA.

- Persistent, unique identifiers are critical when linking records
  - **Institutional identifiers**: eduPersonPrincipalName (ePPN) [I2-EPPN] is widely available and required by the REFEDS R&S entity category. It should be a good key to link records from different sources.

However, recycling/reallocation of ePPN at some institutions creates a data protection risk. Migration to the use of eduPersonUniqueId [I2-EPUI] is preferred and should be supported by R&S.

- **Social/professional identifiers**: ORCID identifier [ORCID] (presented as eduPersonOrcid) [I2-EPO] appears to be a viable way to link to user-asserted data, and to indicate that accounts at different organisations are used by the same person.

- Explicit consent for data sharing should be obtained
  - It is important that users are aware of what personal information is being stored and accessed at a second service.
  - Consent to share an identifier is not consent to aggregate data using that identifier. For example, a user may give consent for their ORCID identifier to be shared by their IdP, but may need to give further consent for aggregation of their ORCID data.
  - The user should be informed about the attributes that will be aggregated. The user's consent to release attributes, which is usually collected by the authentication service, must be obtained in compliance with the General Data Protection Regulation (GDPR) [GDPR].
  - Unnecessary data collection should be avoided. Again, this is in accordance with the GDPR.
- Attributes stored at an AA, IdP or SP post-aggregation should expire
  - Deprovisioning is very important. Failure to deprovision can create privacy and security risks for both individuals and organisations.
  - IdPs and AAs should ideally provide expiry dates for attributes with each assertion – schacExpiryDate is an appropriate existing attribute type for this purpose.
  - Aggregators should expire cached or stored records in accordance with any expiry information from the originating IdP.
  - Aggregators should expire records with no explicit expiry date either in accordance with existing data protection guidelines for their organisation, or within 3 months of an update.
- Check attributes supplied by the user's SAML identity provider or OIDC provider and redirect users to aggregation sources if additional information is required
  - The Shibboleth SP AttributeChecker feature allows SPs to redirect to another source if inadequate data is sent by the user's IdP. This can be used to redirect a user to register with an attribute authority to provide (and give consent to) additional attributes.
- Consider moving aggregation "business logic" away from the SP

  If aggregation is done at the SP/RP from similar, reliable, equally trusted IdPs (maybe from within the same federation), then the aggregation can be kept simple and there is no need for more advanced logic. Attributes can simply be gathered and passed on to the application or HTTP server's access control.

  The future direction of federated identity management (FIM) (especially regarding assurance levels) requires some business logic so that data can be harmonised depending on its source IdP. At the moment, not all SP software can dynamically rewrite attribute data.

  Complex aggregation rules should be moved outside the SP software:

- Rules can be moved into a proxy (especially appropriate for Push aggregation).
- Rules can be moved into the application (the best option for Pull aggregation).

- Scoped attribute values
  - Use of @domain scoping is limited by the strict scope-origin filtering that should be done by SAML SPs for security. A proxy may not be able to pass to an SP an attribute that is scoped to a source IdP, as the SP will, by default, only trust the original IdP to provide attributes with that scope.
  - If information about the source IdP is not required and attributes have been harmonised, then scopes of attributes from suitable sources can simply be rewritten to originate at the aggregator. For example, student@aa1.edu would become student@proxyidp.com. Locally unique identifiers, such as ePPN, must not be used to create new aggregator-scoped identifiers, and if a new identifier is created, the source identifier must always be traceable.
  - Registering all the origin AA and IdP's scopes in the aggregator's metadata is also possible, and may be practical even for large numbers of source IdPs for a proxy service only supporting SPs outside of a federation (such as within a research organisation). Federations are unlikely to allow aggregating proxies to share scopes with institutional IdPs, as the aggregator would be able to impersonate any IdP it shares a scope with.
  - URIs containing domains are naturally scoped. See *Guidelines on expressing group membership and role information* [AARC-JRA1.4A] for examples involving groups.
  - The aggregator must verify that scopes entering the aggregator are from valid IdPs, and belong to the legitimate source.
- Be cautious when using eduPersonEntitlement to store URIs
  - The SAML eduPersonEntitlement attribute [I2-EPE] is intended to contain one or more URIs that indicate a specific entitlement to a resource. The very flexible nature of URIs makes eduPersonEntitlement an often effective workaround to some of the aggregation limitations of SAML assertions.
  - However, this may lead to eduPersonEntitlement being used to represent the aggregated values of many other attribute types such as groups, organisation membership, roles and institutional affiliations, rather than abstract resource access rights. This can create maintainability problems.
  - Try to create useful entitlements at the aggregator that are derived from source attributes, rather than storing other aggregated source attributes in eduPersonEntitlement.
  - Store values aggregated as URIs in more appropriate attributes if a suitable attribute is available and the original data is needed, rather than an entitlement. Examples include identifiers, affiliation, assurance levels, and groups.
  - Research communities can create their own local schemas and new attributes to store aggregated values.
  - Groups are frequently used to indicate shared access entitlements, and so membership of such a group can often be safely expressed with a simple entitlement URI.
  - Care must also be taken to check and filter values when passing eduPersonEntitlement through the aggregator to SPs.

- Filter attributes according to source
  - High-assurance, low-assurance and user-asserted attribute data should not be mixed without careful filtering.
  - Filtering may also be needed to remove unknown or inconsistent values (if normalisation is not possible).
- Attribute vocabularies should be harmonised by the aggregator
  - The aggregator should, whenever possible, tidy and simplify the wide range of possible attribute values into a smaller, known, and more consistent set. This is especially important if a diverse set of IdPs and AAs are being used.
  - The aggregator should become a Single Source of Truth (SSOT). There is a risk that an SP using both processed attributes (from an external aggregator, or that it has aggregated itself) and attributes taken directly from an origin IdP may use unprocessed data by assuming it comes from the aggregator. It should be safer for an SP to only use certain attributes from a single trusted aggregator.
  - As already mentioned, creating new harmonised entitlement values from various source attributes may be more efficient and reliable than processing the source attributes and passing them on to RPs/SPs.

3.3 Guidelines on Token Translation Services

In federated environments, it may happen that there are technological incompatibilities between the source of the user identity (e.g. IdP) and the service that user would like to access. For example, grid environments use X.509 certificates for the authentication and authorisation of users, while current R&E identity federations are based on SAML 2.0. Furthermore, commercial entities (e.g. social networks, cloud solutions) are increasingly relying on OIDC. (Of course, the examples of technological solutions mentioned above are not an exhaustive list.) To increase the adoption of federated identities, maintain interoperability with legacy services or easily deploy new ones, there is a need to provide mechanisms that enable translation between different protocols or technologies. The term “token translation service” (TTS) is a broad term used to denote such mechanisms. These guidelines in this subsection are documented in full in [AARC-JRA1.4C].

- Consistency of user information

  While there are already solutions that translate SAML to OIDC and vice versa, or OIDC to X.509, SAML to X.509, or OIDC to SSH keys, one important point to watch is how information is translated between technologies. TTSs need to properly translate information included in the original token, to information included in the translated token. The different parts of the token or of the information need to be carefully considered, i.e. which token part is used for user authentication (“who are you”) and which part is used for authorisation (“what roles/rights are granted to you”) and how these are translated across different technologies. Best practices and recommendations for translating between federated

authentication and X.509 certificates are listed in [AARC-JRA1.4I]. For SAML <-> OIDC mapping, there is an ongoing effort from the OpenID Connect for Research and Education Working Group (OIDCre) [OIDCre-SAML-OIDC]. Furthermore, in AARC, an effort was devoted to guidelines for implementing SAML authentication proxies for social media IdPs [AARC-JRA1.4G].

- Deployment considerations

  It is generally easier to deploy a "standalone" token translation service with already established services, than to implement it as an "embedded" translation operation. With the former, there is no need to modify existing service operation, and the additional step is added on top of the existing authentication flow.

- Security considerations

  In general, all industry security standards should be followed when executing token translation. This may include employing transport layer security (TLS) in browser communication and between services, safe storage and deployment of credentials (such as SSH and certificate private keys, OAuth2 bearer tokens, etc.). The TTS must avoid the possession of users' institutional credentials at any point.

- Transparency, data protection and data minimisation

  The user should be informed about the attributes that will be released through the TTS. The user's consent to release attributes, which is usually collected by the authentication service, must be obtained in compliance with the General Data Protection Regulation (GDPR) [GDPR]. When the attribute set that will be finally released to the end service changes because of the TTS, the user should be informed as well. Furthermore, the TTS should only request the minimum of data needed for its operation. Unnecessary data collection should be avoided. Again, this is in accordance with the GDPR.

3.4 Guidelines for Implementing SAML Authentication Proxies for Social Media IdPs

One of the major goals of the blueprint architecture is to support users in research collaborations who do not have a federated identity via their home organisation. Moreover, there are cases in which an individual researcher is not affiliated with any of the traditional home organisations. To cater for these cases, the AARC blueprint architecture enables research communities and infrastructure providers to connect to identity providers that are not part of any of the eduGAIN participating federations. Such guest identity providers include social networks, which typically use OpenID Connect/OAuth2 for authentication and authorisation. This subsection provides recommendations and best practices for implementing authentication proxies that can connect social media identity providers with federated SAML 2.0 service providers.

The guidelines presented in this subsection have been defined based on experiences from multiple parties in the AARC project and have subsequently been discussed and tested through the SA1 pilot on using social media as guest identities for federated access [AARC-SA1-SCP]. They are documented in full in [AARC-JRA1.4G].

- In order for the proxy to support the REFEDS Research and Scholarship [REFEDS-RS] attribute bundle, the RI/EI needs to make sure that the social authentication application (typically an OAuth2/OIDC client) is properly configured to request the required data elements from the social IdP. The RI/EI should therefore set up appropriate permissions and request scopes to allow users to authorise their social IdP to release information such as the shared user identifier and email address.
- In SAML, the recommended user identifier is the eduPersonUniqueId (ePUID) [I2-EPUI], which is a long-lived, non-reassignable, shared identifier. While ePUID is formatted like an email address, it is not intended to be a person's published email address or to be used as an email address. In fact, the released email address should never be used for the user's ePUID as social identities can have multiple email addresses at different points in time.
- In the case of OIDC-compliant social IdPs, the subject (sub [OIDC-Sub]) and issuer (iss [OIDC-Iss]) claims can be used together as a stable global identifier for the end user, since the sub claim is locally unique and never reassigned within the issuer for a particular end user. Therefore, the combination of the iss claim and the sub claim is appropriate for calculating a SAML ePUID. Any algorithm with the following properties can be used to calculate ePUIDs:
  - Distinct combinations of the iss and the sub claim MUST result in distinct ePUID values.
  - The algorithm MUST be deterministic.
  - The "uniqueId" portion MUST contain only alphanumeric characters (a-z, A-Z, 0-9).
  - The "uniqueId" portion MUST be less than or equal to 64 characters.
  - The "scope" portion MUST be the administrative domain of the social IdP proxy where the identifier was created and assigned.
  - The "scope" portion MAY contain any Unicode character.
  - The "scope" portion MUST be less than or equal to 256 characters.

  Based on the properties above, this guideline propose the following algorithm:

  $$\text{ePUID} = \text{SHA-256}\ (\ \text{sub} \mid\mid \text{iss} \mid\mid \text{salt}\ ) \mid\mid \text{'@'} \mid\mid \text{scope}$$

  where the sub claim is concatenated with the iss claim and a static salt value. The concatenated string is then hashed using SHA-256. The result is then scoped at the administrative domain of the authentication proxy where the identifier was created and assigned.
- In the event that a non-shared (targeted) user identifier is released by the social IdP, then different SAML authentication proxies will receive distinct user identifier values for the same end user. This will result in distinct ePUID values, even if the same generation algorithm is being used. However, it should be noted that the services behind a given authentication proxy will still be able to identify users consistently, since the proxy-specific user identifier will remain the same.

4 Best Practices for Managing Authorisation

Once a user is authenticated, resource access is granted by authorisation policies enforced by service providers (SPs). National identity federations and eduGAIN, through identity providers (IdPs), provide a well-established authentication service for a home organisation's (HO) users. However, nowadays, research collaborations (RCs) are composed of members belonging to different federations. An RC usually relies on digital resources and services both to do research work and to communicate and manage the collaboration itself, but it is the RC that is responsible for defining the access rights to its resources. Therefore, authentication and authorisation processes for resource access are very important for RCs.

Authorisation policy enforcement always happens on SPs (even though not always on just the resource SP alone, e.g. in the case of an IdP/SP proxy). When an SP has all the information to build a consistent authorisation policy for each user, this is not a problem. However, there are several cases where this is not possible or desirable, especially when RCs are involved. SP-managed authorisation, or application-based authorisation, can face issues related to (multiple) source of authority, scalability, and RCs interaction.

HO IdPs may be used as an authoritative source of information for determining resource access. A common example is the "common-lib-terms" entitlement, which is used by many institutions to signal to a publisher that the user is authorised to access the SP resources (and that the institution will pay for its use).

As already stated, in the case of RCs there is not just one HO on which to rely for authoritative information.

SPs cannot easily collect information to manage different access rights for thousands of users, and often the IdPs of R&E entities such as universities deal with tens of thousands of users.

A user may be a member of many RCs, and it is impractical for an institution to manage RC-specific entitlement information on a per-user basis. A more practical approach for RCs is to create a virtual organisation (VO) and to manage its entitlements itself. The information on which authorisation policies rely can then be collected using an attribute management system and its information exposed via attribute authorities (AAs).

This section, which is based on *Best practices for managing authorisation* [AARC-JRA1.4E], covers the following topics:

- Authorisation information sources.
- Authorisation attributes.
- Additional considerations.

4.1 Authorisation Information Sources

As noted above, while authorisation policy enforcement takes place at the SP, the information that needs to be evaluated by those policies, usually attributes and roles, will often need to be sourced from different providers. Two such authorisation information sources are considered below: identity providers and attribute authorities.

4.1.1 Identity Providers as Authorisation Information Source

When IdPs are used as the source of information for authorisation purposes, user information is encapsulated in attributes and transmitted to SPs along with the authentication assertion, as shown in Figure 4.1.

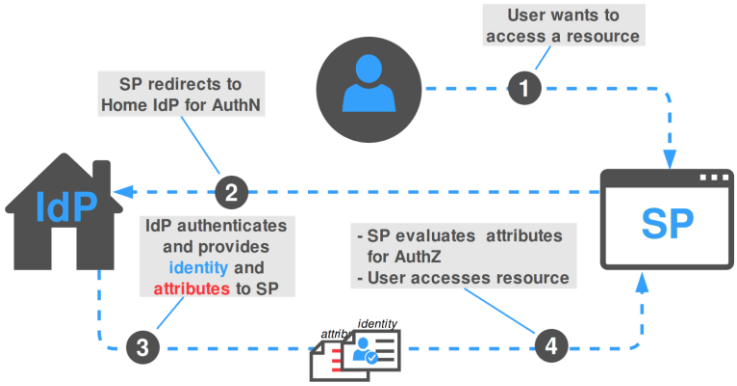

Figure 4.1: IdP as authorisation source: SPs leverage attributes coming from IdP

4.1.2 Attribute Authorities as Authorisation Information Source

AAs can store additional user attributes[3] including, but not limited to, group membership, virtual organisation (VO) affiliation and/or role.

In SAML authentication flows, AAs do not participate in the authentication process, so to use them as a source they have to be queried directly. Typically, the user identifier from the authentication is leveraged as a user identifier to retrieve the additional attributes at the AA.

Depending on who queries the AA and at what time, three general models can be outlined. Each of these is described below.

[3] Many virtual organisations also issue an extra identifier for the user.

4.1.2.1 *Identity Provider*

The HO IdP can collect additional attributes from AAs with the SAML attribute query, or by employing custom connectors available on the AA (if the AA is managed by the home organisation IdP, direct database/directory queries are often employed). In this model (Figure 4.2), the IdP will aggregate all the attributes and push them to the SP, so that the SP does not need to be aware of the existence of the AA.

Importantly, this model is not suitable for RCs: it would require too great a coordination effort to have all the RC members' HO IdPs query a common AA, aggregate the attributes, and finally release them to the SP(s) used by the RC.

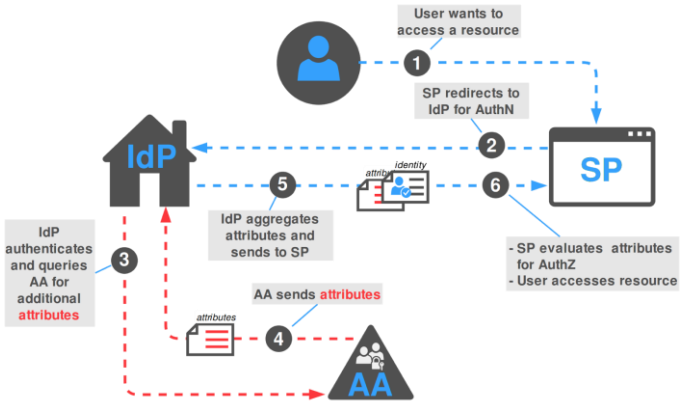

Figure 4.2: AA as authorisation source for IdP: IdP aggregates AA attributes and pushes them to SP

4.1.2.2 *Service Provider*

In this model (Figure 4.3), SPs are aware of the existence of an AA that has to be queried to retrieve attributes useful for the enforcement of the authorisation policies.

In the SAML world, SPs use the SAML attribute query to pull the additional attributes after receiving the authentication assertion from the IdP along with a user identifier. In order to query the AA, the SAML attribute query must contain a user identifier linked to the user's attributes.

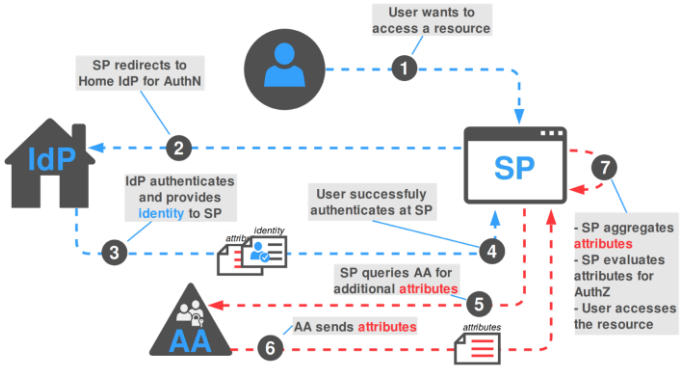

Figure 4.3: AA as authorisation source for SP: SP queries AA for attributes

4.1.2.3 *IdP/SP Proxy*

IdP/SP proxies sit in between the IdP, which performs the authentication, and the SP that will receive the authentication assertion and the user attributes. Proxies with attribute aggregation and external attribute query features can thus modify the attributes set that is part of the authentication flow. Additional attributes, which will eventually be used by the SP to enforce the authorisation policies, can be retrieved from AAs and aggregated into the original set.

In this model (Figure 4.4), the attributes coming from the IdP and the AA are pushed to the SP. Neither the IdP nor the SP needs to be aware of the existence of the AA, but they both should have a trust relationship with the proxy.

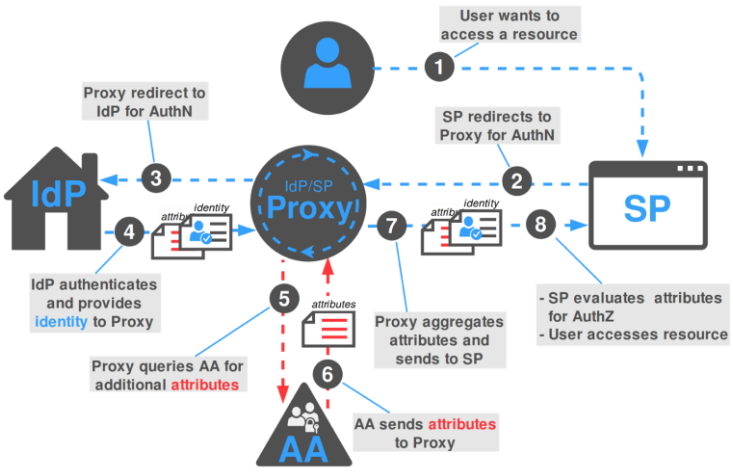

Figure 4.4: AA as authorisation source for IdP/SP proxy: proxy aggregates AA attributes

4.2 Authorisation Attributes

This section considers three types of information that can be used for authorisation: affiliation, entitlement and level of assurance.

4.2.1 A Basic Example: Affiliation as Authorisation Data

In a simple scenario, affiliation information, usually released by IdPs, can be used for authorisation. In the context of the SAML 2.0 protocol and its use in eduGAIN, this information is transmitted employing eduPersonAffiliation (ePA) [I2-EPA] or eduPersonScopedAffiliation (ePSA) [I2-EPSA], and preferably the latter.

It is important to understand that affiliation can be authoritatively asserted only by the organisation to which the user belongs, the so-called home organisation.

Table 4.1 below shows a practical example of how to use ePSA to enforce authorisation policies, taking an SP that has two service levels: base and advanced. To qualify for each service, ePSA attribute values are evaluated at login time. In the example shown below, not all the values of ePSA let the user access the protected, advanced services and, as a minimum, the member affiliation is needed to successfully log in.

User ePSA Values	Policy		
	Login	Base Service	Advanced Service
affiliate@foo.bar	NO	N/A	N/A
member@foo.bar, student@foo.bar	YES	YES	NO
member@foo.bar, staff@foo.bar	YES	YES	YES
member@foo.bar, faculty@foo.bar	YES	YES	YES

Table 4.1: Affiliation as authorisation data

It is worth mentioning that affiliation, as defined in the eduPerson schema for the attribute ePSA [I2-EP], carries some role information also, but in a very broad and general sense. In the context of research collaborations, affiliation can be used for coarse-grained authorisation management. When resource access is based on more fine-grained authorisation policies, entitlements (see Section 4.2.2 below) should be preferred.

4.2.2 Entitlements

Entitlements indicate a set of rights to specific resources. In SAML 2.0/eduGAIN, entitlements are stored in the attribute eduPersonEntitlement (ePE) [IP-E2E]. Entitlements can be information targeted on either the resource, or on the user's groups membership and roles.

In the first case, entitlements can represent the specific right of a user to access a resource. In sensitive research fields, such as biomedicine, access to data is subject to approval, and permission to access the data can be conveyed through one or more entitlements.

When group- and role-based access control policies are needed, the membership information can be transmitted with the attributes ePE, or isMemberOf.

Being a URI, eduPersonEntitlement can be used for fine-grained representation and transmission of a variety of information, including groups membership, roles, and scope.

Table 4.2 below shows an example of how to use ePE to enforce authorisation policies, taking an SP that has two access levels: user and manager.

User ePE Values	Policy	
	User Access	Manager Access
urn:mace:<namespace>:<authority>:group:vo.example.org	YES	NO
urn:mace:<namespace>:<authority>:group:vo.example.org:role=manager	YES	YES

Table 4.2: eduPersonEntitlement as authorisation attribute for groups and roles

Entitlements are suitable for conveying authorisation information for research collaborations and VOs.

A standardisation effort regarding the use of entitlements to best represent groups membership and roles has been carried out within AARC [AARC-JRA1.4A]. AARC2 will address more advanced scenarios related to distributed authorisation.

4.2.3 Level of Assurance

The authorisation process can also be related to the level of assurance (LoA). Assurance information can be transmitted leveraging attributes, such as eduPersonAssurance [I2-ePAs], and the SAML authentication context class.

While some identity federations[4] and e-infrastructures[5] have deployed their own LoA scheme, the REFEDS Assurance Working Group and AARC are working to define common LoA recommendations [REFEDS-AWG].

4.3 Additional Considerations

This section addresses three further considerations relating to authorisation management: trust relationships, delegated authorisation management, and authorisation attributes and token translation.

4.3.1 Trust Relationships

In some of the models related to using attribute authorities as the authorisation information source, the trust relationships among all the components are heavily dependent on the architecture and the attributes flow. Another important factor to consider is the ownership of each component, since it can reshape the circle of trust (for example, if the AA is owned by the same home organisation that owns the IdP).

In the case of using an AA as the authorisation source for an SP, all the trust relationships should be considered direct, as shown in Figure 4.5. For IdP to SP and AA to SP, the nature of the trust relationships is apparent, since they need to exchange information. The trust relationship between the IdP and the AA is also shown as direct, since the AA should be linked to the IdP with a shared user identifier.

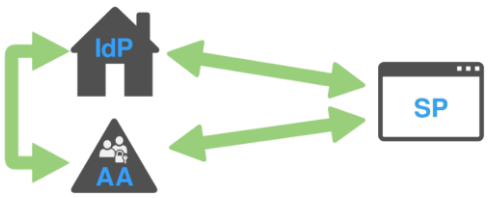

Figure 4.5: Trust relationships for AA as authorisation source for SP

In the case of using an AA as the authorisation source for an IdP/SP proxy, the trust relationships between the IdP/SP proxy and all the other components are direct, but those between the SP on one side and the IdP and the AA on the other are indirect, as shown in Figure 4.6. In theory, because of the proxy-based attributes flows, the SP does not need to be aware of the existence of the AA, or even of the home organisation IdP that has originally authenticated the user. In real life, trust relationships between the HO IdP and the SP often pre-exist, either because the SP services are contract-based, or because they are both connected through a national identity federation or through the eduGAIN inter-federation service.

[4] Many eduGAIN identity federations have defined their own LoAs. As an example, see the SURFnet LoAs [SURF-LOA].

[5] EGI is an example of an e-infrastructure that has defined its own LoAs for authorisation purposes. See [EGI-LoA].

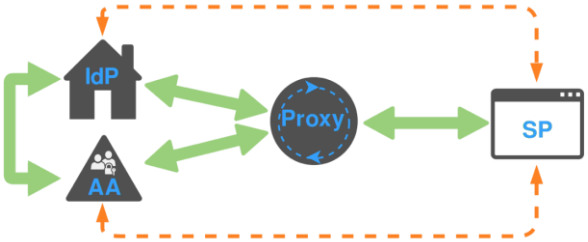

Figure 4.6: Trust relationships for AA as authorisation source for IdP/SP proxy

4.3.2 Delegated Authorisation Management

In the context of research collaborations, it is necessary, or at least desirable, to delegate the management of the resource authorisation entitlements to people who are not the direct resource owners. This is a very common scenario in RCs. An RC typically has a VO/group management system in which the members of the collaboration are assigned to groups and/or roles. Access to the resources is typically granted based on the group(s) that the user belongs to in the collaboration, and/or based on the role(s) (s)he holds. In small collaborations, with a small number of members and resources, authorisation management can happen centrally. In larger collaborations, the internal structure of the collaboration is much more complicated and this might require the group/role membership management to be distributed and delegated as needed, to map the collaboration structure.

4.3.3 Authorisation Attributes and Token Translation

Token translation services can be used to connect IdPs and SPs that employ different authentication protocols, but while syntactic and semantic differences among protocols can be addressed, the differences in context reference are more difficult to overcome.

In a scenario where SAML 2.0/eduGAIN is the destination authentication protocols/context reference and OIDC/Google is the source reference, some attributes and concepts will not be available at the origin, so they will not be available at the destination. Examples of attributes that cannot be easily translated are:

- Affiliation.
- Entitlements.

To overcome this issue, proxies with attribute aggregation and external attribute query features can leverage AAs to collect additional attributes to be used as authorisation entitlements.

5 Access to Non-Web Services

There is a need for non-browser-based federated access to resources ([AARC-DJRA1.1], requirement R14) and this type of access brings more technical problems to solve than web-browser-based access. Involving a third party (IdP) in the process of authentication and authorisation may be realised by two methods: by having the IdP issue and sign assertions prior to accessing the resource (e.g. in the form of X.509 certificates) or by querying the IdP while accessing the resource. The first method is not always feasible, or recommended, as it requires using modified software and handling of the signed assertions by the users (e.g. certificates, SAML assertions, OAuth2 token). The second method is often used for web-browser access, as features of the HTTP protocol (e.g. redirection) and client-side scripting allow its implementation. This section focuses on non-browser access to resources, which to date has often been realised using typically non-federated methods of authentication, meaning that legacy client or server software must be modified to support federated identity management.

Any writable access to resources usually require local, non-transient identities like those used by the operating system, databases or other services (e.g. iRODS). Such an identity typically requires prior provisioning (a local account must be set up) and deprovisioning if it is no longer used. During a sign-on the global identity of the user must be securely mapped to the local one. Additionally, local privileges are decided by group membership or role binding. This process may be split into two parts: token translation from the federated authentication protocol to the local authentication protocol, and translation from federated attributes to the local account name and local rights.

In all cases, the challenges involved can be grouped into the following categories:

- Account provisioning.
- Authentication:
  - Federated credentials or
  - Local credentials (still require a check if the remote credentials are up to date)
- Authorisation usually boils down to attribute mapping to local account and groups.
- Account deprovisioning means removing an account and freeing resources (e.g. data stored) that are no longer used. The analysis of this problem is outside the scope of this document.

This section, which is based on *Guidelines on non-browser access* [AARC-JRA1.4F], considers solutions to non-browser access using SSH/SFTP and HTTP APIs.

5.1 SSH/SFTP

SSH allows secure (encrypted) shell access to an operating system account on a remote machine over an unsecured network and SFTP allows file transfer using SSH. This technology is also used for port forwarding, proxying or restricted command execution (as used in GitHub and SVN). Typically, the authentication is based on an account name (username) and password or a pair of cryptographic keys. Additionally, if both client and server applications support Generic Security Service Application Program Interface (GSS-API)[6], another authentication mechanism may be provided. In all cases, the account on the resource server must already exist (locally or in a database store). In addition to the authentication mechanisms inside the SSH server, operating system authentication mechanisms may be used (e.g. Linux PAM).

This section explores the use of SSH/SFTP by considering the following scenario: there is a need for a service providing shell access or secure file transfer to a server. The service needs to authenticate the user and map him to a local account and groups. The mapping must be based on user attributes. The local accounts must be provisioned automatically. The solution must leverage federated access.

The above scenario may be affected by some use-case-specific constraints, e.g. policies, already-existing services, allowed protocols and software, etc.

Two possible solutions are described below: GSI-enabled SSH, and SSH key provisioning with web portal. The limitations previously mentioned should be kept in mind when selecting the appropriate one.

5.1.1 GSI-Enabled SSH

The Grid Security Infrastructure (GSI) is a specification for secure communication between software components, where authentication is based on PKI and credential delegation. GSI-OpenSSH is a modified version of OpenSSH that adds support for GSI, providing a single sign-on remote login (gsi-ssh) and file transfer services (gsi-scp and gsi-sftp). This solution requires both modified client software and modified server software. The users must have an X.509 certificate, so that the solution fits specific use cases.

PKI allows authentication to be delegated to an external entity. Typically, the user obtains a long-lived certificate signed by a CA related to their institution or community and uses it to create a short-lived proxy certificate used for the authentication. Another possible approach is to obtain a short-lived certificate from an online CA[7] that uses another method of authentication.

[6] Note that this support may be limited depending on the SSH implementation, e.g. OpenSSH supports only Kerberos GSS-API and the “gssapi-with-mic” authentication method, which is not sufficient for GSI, described in Section 5.1.1.

[7] See the AARC CILogon TTS pilot [AARC-CILogon] and the RCauth.eu service [RCauth] as a reference.

In order to obtain a (short-lived) certificate from an online CA on the command line, several possibilities are now available. Until recently, the only options were for the user to manually export a credential from the web browser or download directly from the online CA. The user would then need to manually put the credential in the right place. Recently, using the enhanced client protocol (ECP) interface of the CILogon service, the Open Science Grid and LIGO have started using a SAML-ECP-based command-line tool to automatically download and handle the credential[8]. The approach reduces the certificate + key to an opaque token, much like Kerberos[9].

The certificates may hold attributes signed by some attribute authority (e.g. VOMS) to be used while mapping to a local account. The mapping and account provisioning mechanism is pluggable in GSI and one, commonly used, is LCMAPS, which maps users to accounts from pre-created pools using VOMS attributes.

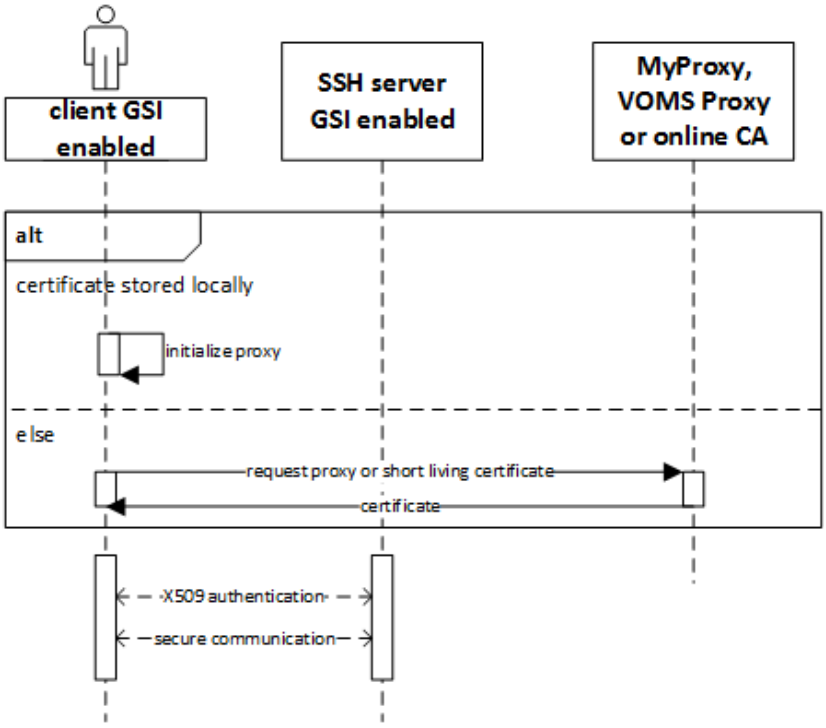

Figure 5.1: Sequence diagram for GSI SSH

[8]For a description of the CILogon ECP profile and the tool that uses it, cigetcert, see [CILogon ECP] and [CIGETCERT].

[9] An alternative approach, intended to circumvent the problem of limited ECP availability and to be implemented by the RCauth, is to use SSH-key authentication to retrieve and automatically handle a short-lived credential from a MyProxy server, where the key is previously uploaded to a web portal (see Section 5.1.2).

5.1.2 SSH Key Provisioning with Web Portal

This approach requires an additional step (logging into the web portal) to be taken by the user prior to accessing the non-browser resource, in particular an SSH resource. The flow is as follows (summarised in Figure 5.2):

- The user logs into a dedicated web portal that performs authentication and authorisation. It may leverage HTTP features and the existing variety of implementations of authentication methods for web solutions.
- The user uploads or generates credentials to be used while accessing the non-web resource. Typically, a public cryptographic key is uploaded or a pair of keys is generated. In addition, an account name on the resource is generated or selected by the user. Mapping of the user identity to that name must be persistent over multiple logins.
- The service provisions or updates an account on the resource and maps the credentials in the background. This is achieved by, for example, modifying an LDAP that provides accounts on the resource or by using an API to a VM hypervisor to set the public key in a VM. Additionally, group or VO membership on the portal may be mapped to local groups on the resource in order to achieve fine-grained authorisation.
- The user accesses the resource using the configured credentials.
- Access to the resource is enabled while the user has a valid login session opened in the portal. The access can be locked, for example by removing the public key from the LDAP or the VM. Note that this is not account deprovisioning.

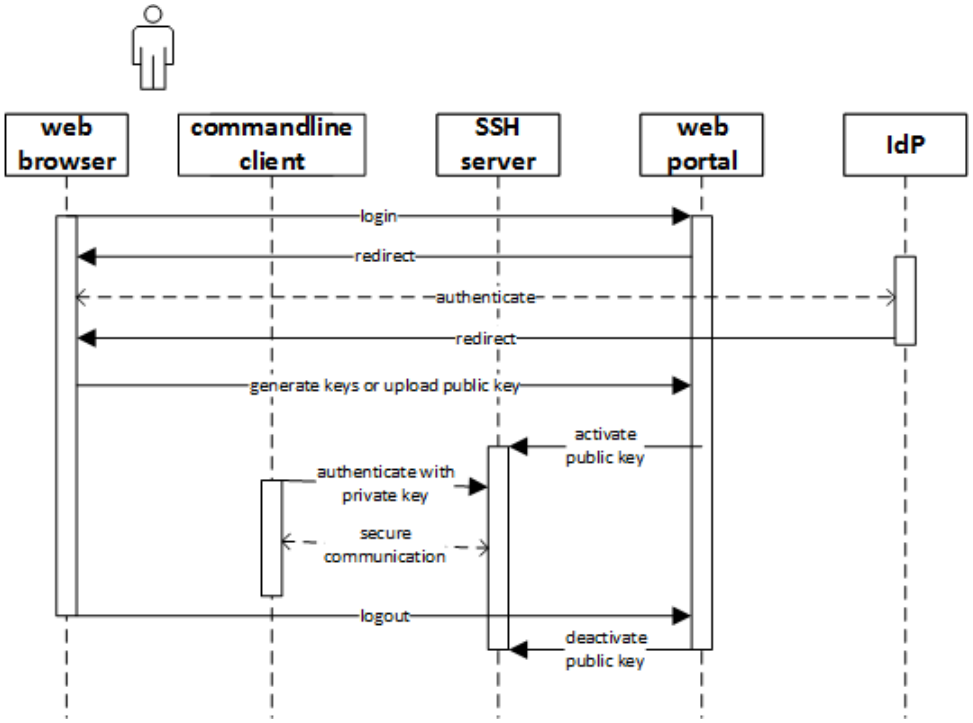

Figure 5.2: Sequence diagram for key provisioning with web portal

The AARC project has successfully piloted two solutions that follow this approach: COmanage SSH key management [AARC-CO-SSH] and WaTTS TTS SSH pilot [AARC-WTS-SSH].

5.2 HTTP APIs

HTTP APIs allow the implementation of applications that access services with the HTTP protocol. These applications may be run from a web browser or standalone, without a GUI or even in batch mode. Non-browser applications have limited ability to involve a third party in the authentication and authorisation process (using HTTP redirect) and to interact with the user.

This section explores the use of HTTP APIs by considering the following scenario: there is a service providing an HTTP API to let the clients access and manipulate certain resources (e.g. IaaS cloud or cloud storage). A user community (UC) wants to use the service from a specific application (e.g. batch script or office software) and also wants to leverage federated access. The HTTP API of the service needs to authenticate the user and take authorisation decisions based on the user's attributes (e.g. grant access to some VMs or files for users and groups).

The HTTP APIs support a variety of methods for authentication and authorisation. Note that the above scenario may have two variants: (a) the service already exists and the methods cannot be changed, and (b) the service is in the design phase and a preferred method can be selected.

The following technologies may be considered:

- OIDC/OAuth2.
- X.509 certificates.

On the other hand, the user identity and attributes taken from the user community may be provided by a different technology. Typically, for the above scenario it might be:

- SAML IdP.
- X.509 certificate.
- OAuth2 Authorisation Server (usually guest identities).

Since the AAI technologies and domains used by the service and identity or attribute provider may be different, token translation may be necessary. Two possible solutions are described below.

5.2.1 Accessing HTTP APIs Using OIDC/OAuth2

The solution is based on the following flow (summarised in Figure 5.3):

- The user goes to a token generation service (i.e. part of the token translation service) using their browser.
- To access the service the user has to authenticate at their home IdP (standard browser-based flow).
- After successful authentication/authorisation the service can generate an API token for the user, i.e. an OIDC access token.
- The OIDC access token can be used by command-line clients to authenticate against HTTP APIs that support OIDC/OAuth2. The token must be manually copied from the browser and pasted into the command line.
- The HTTP API service can use the OIDC access token to retrieve user information from the OIDC provider component of the proxy service.

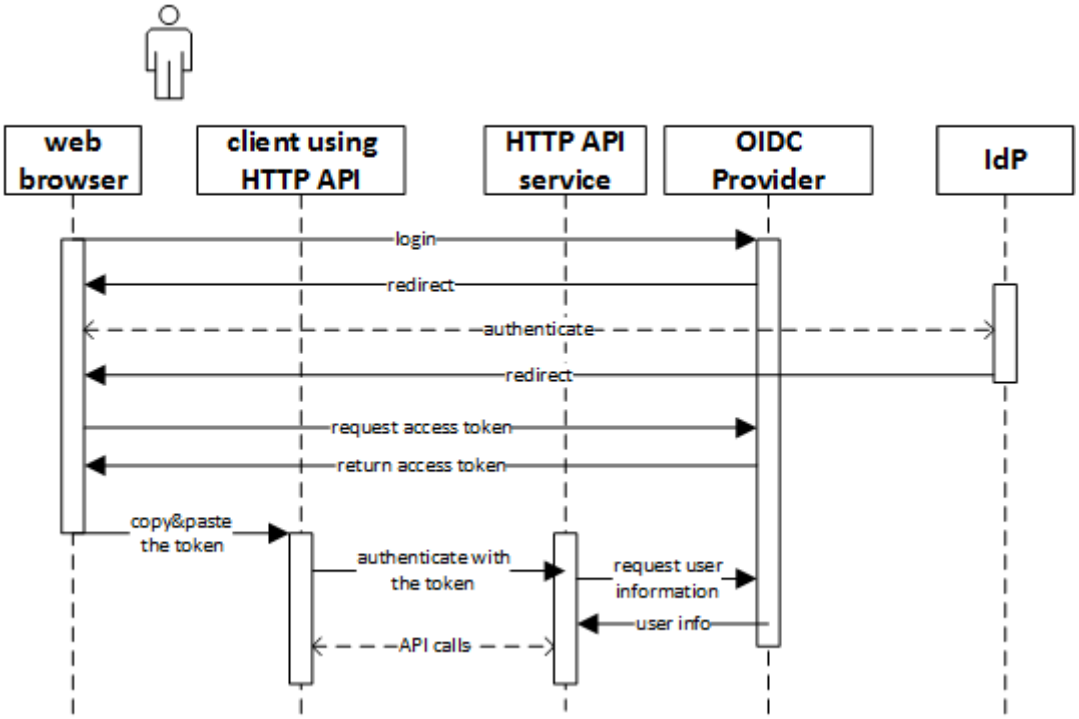

Figure 5.3: Sequence diagram for accessing HTTP APIs using OIDC/OAuth

5.2.2 Accessing HTTP APIs using X.509 Certificates

If the user already has an X.509 certificate and may use a derived proxy in the API, neither token translation nor the interaction is required. If he/she does not, then X.509 short-lived certificates can be retrieved by a token translation service accessible via a web browser.

This API can also be used in delegation scenarios, where the first service to be accessed (for example, a web portal) needs to access other services on behalf of the end user. Such delegation is done using RFC3820 proxy certificates. An example of such a situation is third-party transfer copies between GridFTP-based storage elements all handled from within a web portal, where the portal has obtained an X.509 (proxy) certificate for the user via a portal delegation scenario such as that used in the CILogon-based AARC pilot.

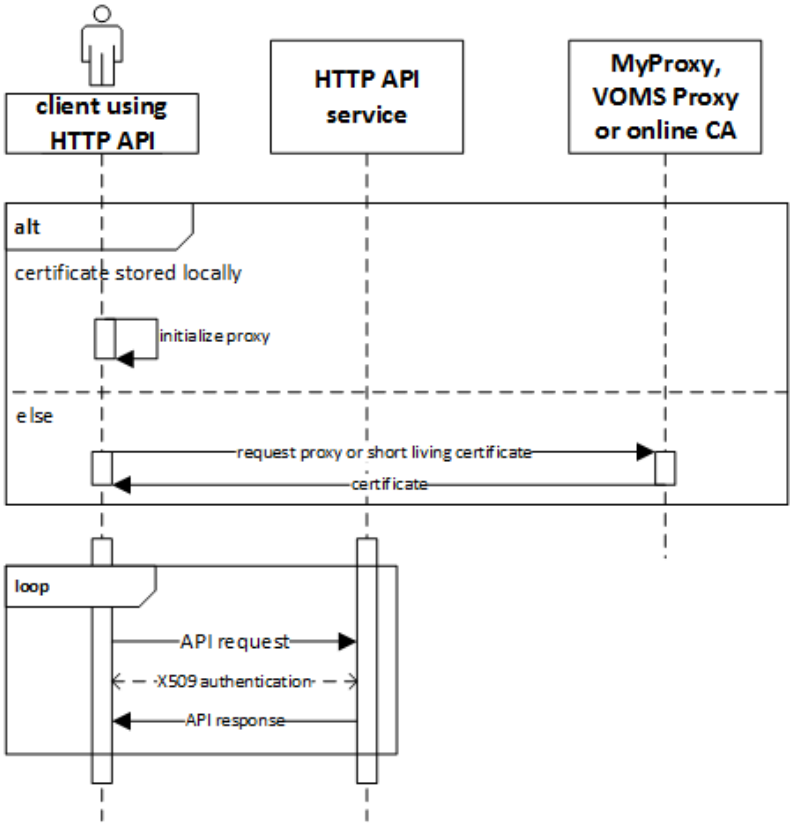

Figure 5.4: Sequence diagram for HTTP API using X.509 certificates

5.2.3 Accessing HTTP APIs using service specific API tokens/passwords

This type of solution is used by online services, e.g. Github [GITHUB-1], Google [GOOGLE-1] and Apple [APPLE-1]. They can provide access to an API which otherwise would require user interaction, e.g. via a browser or by a second factor authentication. They are used e.g. as a password via a basic authentication header. To obtain them, the user should visit the web interface, where they can be generated and subsequently copied from. A token will typically have certain rights attached to it, which can be adjusted or revoked from the same web interface. As such, these tokens are very similar to OAuth token and the flow is much like the one described in 5.2.1 for OIDC/OAuth2.

6 Credential Delegation

Federated single sign-on (SSO) has focused on providing authentication and access control for websites accessed directly by the user's browser. For example, logging in to a library resource, or accessing a virtual learning environment (VLE). The user's IdP passes information to the service he/she wishes to access. This assertion is usually strictly limited to being valid only on the website it was created for.

However, in distributed environments it is often necessary for a remote service to access other services on behalf of a user, or for a software agent to act on behalf of the user. In this case, it is necessary to securely delegate the user's "rights" from the website he/she originally accessed to a wide variety of other applications, such as mobile applications, intranet services and HPC clusters.

The oldest form of credential delegation is to store and reuse the user's login password. This is neither possible nor safe in a distributed environment. Current delegation credentials include signed assertions, session tickets, "tokens" of various types, and proxy certificates.

This section, which is based on *Guidelines for credential delegation* [AARC-JRA1.4D], considers requirements for delegation, to aid (a) the selection of the optimal technology for implementing delegation, and (b) deployment and operation of the technology. It covers:

- Types of delegation.
- Delegation features.
- Guidelines for delegation.
- Risks associated with delegation.

While the examples use specific technologies, they should also illustrate the situations where delegation is useful.

6.1 Types of Delegation

The term "delegation" is sometimes used to cover slightly different scenarios:

- Delegation of **rights** to another person. Combined with role-based access control (RBAC), this becomes entirely an authorisation question and is outside the scope of this document. The delegatee is typically a person, but could also be a service.

- Delegation of **access** to another person – a client, service, or person requests and obtains the right from the "owner" of a resource to access the resource, as in OAuth2. The token is typically issued for a limited time and purpose.
- **Credential** delegation (sometimes called impersonation). The most primitive example is copying the username/password as mentioned above. More useful examples include Kerberos "proxiable" tickets (RFC1510) and GSI (RFC3820)[10]: a remote service (typically a host/service, not a person) obtains a full or limited credential with which it can act on behalf of the user.

The first point is out of scope for this document (it will be covered in AARC2); the more widely used second and third points are covered here.

6.2 Delegation Features

This section summarises feature-related points to consider when selecting technology/implementations for delegation (beyond the usual questions of maturity of technology, interoperation of implementations, etc.)

The discussion considers a scenario in which a delegatee acts on behalf of a delegator (and by assumption is authorised by the delegator), and a token is issued to the delegatee by some authority to enable it to do so.

1. Can the participants (delegator, delegatee) be humans/automated?
   a. For the authentication of human participants, is federated identity management (FIM) supported?

   (Note that the resource, to which access is delegated, is usually non-human.)
2. How does the delegation integrate with existing authorisation?
3. How is the token validated?
   a. By the delegatee?
   b. By the resource accessed by the delegatee?
   c. Can the token be revoked, such that validation by the resource will fail?
4. Is the technology web-based, or does it support non-web access?
   a. If web-based, does it work with basic HTTP clients such as curl (plus perhaps some other standard components, such as XML or JSON parsers)? Or does it require that the client be a browser (needs JavaScript, user intervention)?
   b. If it supports non-web access, does it also work with web servers/clients?
5. Does it support onward delegation (from the delegatee to a second delegatee)?

[10] Technically, it is also worth mentioning SAML enhanced client protocol (ECP), but it is not widely supported in the relevant identity federations.

a. With the originating user's permission?
b. Without?

6. Can the scope of the delegated credential be limited? Options include:
   a. Limited time (possibly pre-dated, valid in the future).
   b. Locked to a particular delegate.
   c. Not further proxiable.
   d. Limited number of uses.
   e. Limited set of activities it can be used for.
7. Is the delegatee's use of the delegated token logged/audited?
8. How is the delegated credential stored and protected by the delegatee?

6.3 Guidelines for Implementing Delegation

For the integrator/developer/architect, faced with the need to delegate within their infrastructure, a decision needs to be made regarding the technologies to use. Different projects solve the same problem in different ways, either because they have to integrate with different existing infrastructure or because they require different features. Suggested steps to follow are:

- Decide which features are needed (see sections 6.2 and 6.4).
- Select the technology, preferring standards-based and interoperable, then mature.
- Look up best practices for running it and implement them.
- Check the risks (see Section 6.5).
- Ask for help from the experts.

The key point is to select the technology based on the features required; once the selection has been made, the technology and the features (including the features not supported by the technology) may give rise to specific risks, which should be assessed.

6.4 Example of Feature Selection

Table 6.1 illustrates technology selection based on features, focusing on three very common and widely supported technologies: GSI proxies, OIDC/OAuth2 and Kerberos. There are other technologies with similar or different feature sets; in particular, the editorial decision has been taken to not cover SAML ECP[11] [SAML-ECP-

[11] The delegation part of the specification was not included in the main Shibboleth implementation.

1], Macaroons [MACAR], iRODS tickets [IRODS], Dynafed [DYNAF], and other relevant but less widely used technologies.

Feature Required	GSI Proxies	OIDC/OAuth2	Kerberos
Human/automated participants	Both	Limited[12]	Both
Integration with existing authorisation	Fully integrated with VOMS	Own authorisation	Via LDAP/Active Directory
Token validated?	Verify digital signature of the chain and check for revocation	Call-out to AS	Digital signature
Web/non-web (primarily delegatee/resource)?	Mainly non-web	Mainly web	Both
Onward delegation?	Yes, chained (RFC 3820)	No, but see [AARC-JRA1.4D] for discussion	Yes, at least within realm
Delegation can be restricted (in the sense of Feature 6)	Partly	Depends on token type	Partly
Logged/audited use?	Yes	Yes (in AS)	No
How is it protected?	Filesystem	Filesystem	Filesystem
Revocation	Yes	Implementation dependent	No
Time limitation	Proxies are conventionally short-lived (O(12h))	Implementation dependent	Yes

Table 6.1: Example feature selection for credential delegation

[12] The "Resource Owner" (cf. RFC 6749) generally has to be human (i.e. the "End-User" of [OIDC-GUIDE]), with a browser; however, see also Section 5 Access to Non-Web Services.

Although the table covers only the high-level questions, rather than addressing the minutiae of the discussion, it should illustrate the selection factors for different delegation technologies, and it could point the way to further work on delegation, if needed.

6.5 Risks Associated with Delegation

Table 6.2 below summarises several general risks associated with delegation. It does not cover the technology-specific risks. (For example, if OAuth is used with bearer tokens (RFC 6750), there is a risk that a stolen token could be misused, as it intentionally does not support Feature 6.b. (Locked to a particular delegate), but it thus allows delegation to unregistered clients. If the use of the token (through the validation) is logged by the Authorisation Server (Feature 7), it could help mitigate this risk.)

	Description	Type	Owner	Mitigation
0	Delegation is not supported by protocol or infrastructure	Proto	Infrastructure	See earlier subsections of this section and the guidelines on token translation services (Section 3.3).
1	Delegated credential is compromised	Op	Infrastructure	Use revocable and/or short-lived credentials. Adopt best practices for operational security (see [SIRTFI-1] sections OS, IR, and PR.)
2	Delegated credential used for non-user-approved activities, by being too widely applicable, or by being forwarded/delegated without the user's consent	Op	User	Credentials with limited use/ purpose/locality and/or which are revocable. Auditable (user-visible) use of delegated credential.
3	Delegated credential is delegated further (without authorisation)	Op	User	Special case of Risk 2 (or Risk 1, depending on point of view). See [AARC-JRA1.4D].
4	Lack of clarity in interpretation of rights of delegated credentials, particularly credentials delegated multiple times (combining restrictions)	Tech, Policy	Infrastructure	For example, see the work by the OGF VOMS attribute PROCessing working group [VOMS-PROC] on VOMS Attribute Certificate Parsing Rules for Chained Identity Credentials.
5	Delegated credential not capable of inheriting same (or selected) authorisations as user credential	Tech	User	Bugfix – may need changes to SP to support delegated credential

	Description	Type	Owner	Mitigation
6	User cannot fine-tune limits on delegated credential so sets the most general limits or, if possible, turns them off	Tech, Usability	Infrastructure	Test with real users

	Description	Type	Owner	Mitigation
7	Delegations don’t work (or features are lost) in a federated environment (e.g. beyond scope of IdP)	Tech Proto	User	Needs research

Table 6.2: Risks associated with delegation

7 Account Linking and Level of Assurance Elevation

In the context of research collaborations, the user is typically assigned an identity by the infrastructure. This "infrastructure identity" consists of a personal, unique, non-reassignable, non-targeted identifier, and additional attributes containing profile information about the user, as well as group membership and role information. In this context, identity linking (also known as account linking) refers to the process of connecting the user's infrastructure identity with their external identities, i.e. identities created and assigned by Identity Providers that reside outside of the administrative boundaries of the infrastructure, such as institutional IdPs or social media IdPs. The identity linking process allows the user to access infrastructure resources as their infrastructure identity regardless of their external identity, used for authentication. It should be noted that the infrastructure identity can be used to obtain different types of credentials for accessing resources, for example, X.509 certificates, SSH keys or other access tokens. In fact, the user may not be aware of the credentials being used to access a specific resource, since in some cases the credentials are translated behind the scenes by the infrastructure.

This section, which is based on *Account linking and LoA elevation use cases and common practices for international research collaboration* [AARC-JRA1.4H], covers the following topics:

- Account linking use cases.
- Account linking process.
- Reconciling identity information.
- Level of assurance elevation.

7.1 Account Linking Use Cases

This subsection presents the main use cases for account linking:

- Consistent user identification/representation.
- Accounting of resource usage.
- Traceability and security incident response.

7.1.1 Consistent User Identification/Representation

An end user typically maintains accounts on different external authentication providers, including:

- Identity providers managed by the user's home organisation.
- Guest identity providers, including social media (such as Google, Facebook and LinkedIn), research community/collaboration-specific identity providers, and national e-government identity providers.

Regardless of the authentication provider being used, the end user wishes to be identified consistently when accessing infrastructure services and resources. This is the main use case for account linking as it supports the reconciliation of identities from various authentication providers, allowing a user to authenticate using any of their identities and still be recognised by the infrastructure with the same user profile. For example, the user can register to the infrastructure with their organisational ID, thus providing a strongly verified credential, but linking their social media ID – which can remain activated in the browser – they can easily "log in" to an infrastructure portal without having to authenticate.

7.1.2 Accounting of Resource Usage

Infrastructure providers may need to track the resources consumed by individual users. Accounting usage data usually associates a usage record to a user ID. Linking the user IDs from different IdPs and the certificate DN of the X.509 credentials allows the collection of all the data that is associated to the user and the different IDs that have been used to access the services. If this account linking is done at infrastructure level, the services do not need to change the way accounting data is produced (e.g. attach several user IDs to one usage record); the data can be merged at the time an accounting report for the user is generated. Assuming the accounting report generator has access to the information about the linked accounts – or just uses the unique identifier of the infrastructure identity – a user will be able to obtain an overview of all the resources consumed using any of their credentials.

7.1.3 Traceability and Security Incident Response

In a federation of multiple service providers that allow different authentication mechanisms, the same users may use different credentials to access different services. In the case of a security incident affecting a service provider member of a federation, the federation may want to prevent a user involved in the incident from accessing services while the incident is being investigated. The service providers need to know which identities have been linked to the account involved in the security incident, either to suspend these identities or simply to investigate whether there are other suspicious activities associated with the user's account. Without account linking, the user's accounts would be registered in the federation as different users, making it very difficult to associate the activities performed with the different credentials.

(See also the infrastructure risks associated with the use of guest identities, summarised in [AARC-MJRA1.2], Appendix A.)

7.2 Account Linking Process

Account linking typically takes place as part of the user enrolment process, either explicitly or automatically, as described in the subsections that follow.

7.2.1 Explicit Linking

In the explicit linking flow, the user requests that an additional identity be linked to their existing infrastructure identity. This flow requires the user to authenticate first with any of the identities already linked to their infrastructure identity (or with the infrastructure identity itself), and then to re-authenticate using the login credentials of the additional identity they want to connect. It should be noted that the administrators of the infrastructure identity management system can also manage identity links, usually to resolve enrolment issues, e.g. duplicate user registrations.

7.2.2 Automatic Linking

The automatic linking process is triggered when one attribute, or a combination of attributes, of one identity correlate to one or more attributes of another identity that is already associated with a registered user. The correlation process may require exact matching of attribute values or tolerate some differences. In the latter case, this could allow for inconsistently capitalised or similar identity values. Automatic linking can prevent an individual from registering distinct infrastructure identities, either accidentally or on purpose. It can therefore be useful in an infrastructure with a strict policy against maintaining multiple user accounts. However, the risk here is that identities which should not be linked may accidentally be matched by this process. Therefore, automatic linking should not be considered unless either the correlation process requires an exact matching on attribute values expressing user identifiers that are personal, globally unique and non-reassignable, while also considering the level of assurance (LoA) associated with the matching attribute(s), or the resulting account is directly derived from the user identifiers that are personal[13], globally unique and non-reassignable. Examples of attributes that may be considered for automatic linking include subject distinguished names of personal X.509 certificates and ORCID identifiers [ORCID]. In other case, such as when detecting the same email address, the account linking process may be automatically triggered, yet it would require explicit user intervention before being applied due to the undefined reassignment practise for such attributes.

[13] A personal identifier is intended for use by a single person, as opposed to shared (or guest) user accounts such as "libraryuser1@university.org".

7.3 Reconciling Identity Information

Account linking requires merging attributes from different identities into the user's infrastructure identity. For multi-valued attributes of the infrastructure identity, the merging process can be based on a simple aggregation strategy, whereby the attribute values from all linked identities are copied to the user's infrastructure identity. However, even then, a user may be allowed to choose their preferred value, e.g. a preferred email address. In the case of single-valued attributes, merging requires selecting a single value from all linked identities. Whether this choice is left to the user, or is selected based on assurance, or some other policy, needs to be decided by the infrastructure.

Some infrastructures recognised the need for managing provenance of attributes in account linking and surveyed existing work in attribute metadata [NISTIR-8112] to maintain the provenance, LoA [EUDAT-Attr], and user consent for the more important attributes. Other infrastructures, such as ELIXIR and BBMRI, have identified use cases that require attribute value selection upon access to resources; for example, to support a user with multiple roles/affiliations (e.g. multiple home organisations or projects they are affiliated with) who wants to log into a service that expects a single role/affiliation. Note that one should be careful about giving the user a choice in the presentation of authorisation attributes to services, in case users elevate their privileges beyond the level to which they are authorised. The classic example is if an IdP asserts membership of a "restricted" group which is denied access to a service, and the user can choose to not assert this membership by selecting membership information from another (linked) IdP.

7.4 LoA Elevation

Each of the external identities linked to the infrastructure identity is usually associated with a different LoA based on various properties of the authentication provider. Currently, the LoA assigned to the infrastructure identity is typically derived from the LoA associated with the authentication provider used by the user when accessing infrastructure resources. However, many infrastructures have identified the need to support the re-evaluation of the infrastructure identity LoA based on the LoA information associated with all linked identities. Specifically, the (re)evaluation model should take into account all four aspects of the LoA (see also [AARC-DNA3.1]) associated with each linked identity:

1. Identifier uniqueness (including the reassignment policy in place)
2. Identity proofing and credential issuance, renewal and replacement
3. Authentication
4. Attribute quality and freshness (primarily pertaining to the home organisation and affiliation information)

Another aspect that may be considered is the operational security of the Identity Provider that may have an impact on the LoA of the asserted identities. Sirtfi [SIRTFI] is a framework that can be used to indicate the operational security of the Identity Provider.

Models for addressing such LoA re-evaluation include:

- Linked high-LoA identity.
- Step-up authentication.
- Origin information.

Each of these is considered below.

7.4.1 Linked High-LoA Identity

The following LoA elevation flow considers all components of the LoA associated with linked identities: A user registers for an infrastructure identity with a low-LoA identity, e.g. from a social media identity provider lacking identity vetting. Subsequently, the user links their high-LoA organisational identity to their infrastructure identity. However, by linking the two identities, the user has proved that they are the same user. Assuming both providers meet the same requirements with respect to the uniqueness of the identifiers and the authentication strength, the infrastructure may assign a high LoA when the user logs in using the social media identity, since it has been linked to a high-LoA organisational identity that makes up for the lack of identity vetting.

7.4.2 Step-Up Authentication

There are also types of linked identities that do not support sufficiently strong authentication methods for high-risk access use cases, despite being otherwise trustworthy (e.g. from the point of view of the identity vetting process). In such cases, the user may register a second authentication factor to enhance the strength of the authentication method and effectively the associated LoA (step-up authentication).

7.4.3 Origin Information

Another model for determining the LoA of a linked identity is by examining the origin information associated with the asserted attributes. While there is currently no standard way to convey such information, some identity providers have defined attribute value metadata aiming to support cross-organisation confidence in attribute assertions. One such example is ORCID [ORCID], which allows researchers to create their account either by self-registration or through institutional login (through the eduGAIN inter-federation service). In general, all the information related to affiliations, publications, awards and grants is provided by the users themselves, so the assurance level is rather low. To provide validated assertions about the users' data, ORCID started the "Collect

& Connect" program [ORCID-CC], through which accounts can be connected with the researchers' home organisations; publications with the publishers; and grants and awards with funders. In this way the assertions about affiliation, authorship, and awards can be verified, and updated as well, by each authoritative source.

This feature can be exploited through ORCID's APIs [ORCID-API], which allow the retrieval of ORCID IDs and related records. Specifically, the retrieved information (e.g. affiliation) is accompanied by the source: in the case of self-asserted information, the source points back to the ORCID user; when the information is inserted and verified by the researcher's home organisation, the source points to the home organisation itself. There are, however, some limitations:

- Only the information that has been made public by the user is retrievable (at least using the ORCID public APIs).
- Matching of ORCID identifiers for home organisations with, for example, SAML 2.0 /eduGAIN entityIDs is not straightforward.

The Umbrella Collaboration is currently investigating the possibility of driving ORCID adoption within their partner organisations (14 photon and neutron sources and their aligned partners across Europe) by potentially providing it as an additional attribute (in an eduTEAMS attribute authority) on login. This is currently subject to a GÉANT eduTEAMS pilot [UMBRELLA]. Additional attributes that are useful to Collaboration partners and services using Umbrella ID via eduGAIN are under consideration.

8 Conclusions

AARC has provided a blueprint architecture that can help architects and implementers of international research collaborations to build scalable and interoperable access management solutions for their communities, on top of eduGAIN. The AARC blueprint architecture uses eduGAIN as the solid foundation for scalable identity services in research and education, and allows the integration of social IDs, and other guest identity services, where needed.

With the introduction of the proxy model, the AARC blueprint architecture enables the direct and indirect use of federated access, even for services that do not directly support SAML, such as OIDC services and legacy non-web-based services. Furthermore, the architecture allows the integration of community-operated user management systems as attribute authorities, enabling the communities to implement authorisation policies based on federated identities, augmented by community-specific information. Highlights of the AARC blueprint architecture include:

- Built on top of eduGAIN, with support for social IDs and other guest identity services.
- Enables federated access for non-web-based services.
- Allows the use of multiple protocols and legacy services via the token translation services.
- Accompanied by a set of guidelines and best practices for implementing authorisation at the community level.
- Supports secure and scalable attribute aggregation.
- Supports the use of levels of assurance.

The AARC blueprint architecture has already been adopted by e-infrastructure providers and research infrastructures. Examples include:

- EGI [EGI-AAI].
- ELIXIR [ELIXIR-AAI].
- EUDAT [EUDAT-B2ACCESS].
- GÉANT eduTEAMS [GEANT-eduTEAMS].
- INDIGO [INDIGO-IAM].

Work on the AARC blueprint architecture will continue in AARC2, focusing on:

- Addressing aspects relating to the integration of the blueprint architecture and its components into the existing AAIs.
- Exploring tools and services for interoperable infrastructures and integrating additional technical components into the AAI design to support a wider range of use cases than to date.

- Exploring service provider architectures and authorisation in multi-SP environments.
- Providing models for the evolution of the AAIs for research collaborations, ensuring cross-sector interoperation.
- Providing guidelines for scalable virtual organisation platforms.

References

Note: If the URLs in the [I2-*X*] references result in a "webpage cannot be found" error, replace "%20-%20" in the browser address field with "#".

[AARC] AARC website
https://aarc-project.eu/

[AARC-AT-X] AARC wiki page: Best practices and recommendations for the attribute translation from federated authentication to x.509 credentials
https://wiki.geant.org/display/AARC/Best+practices+and+recommendations+for+the+attribute+transaltion+from+federated+authentication+to+x.509+credentials

[AARC-BPA-Web] AARC Blueprint Architecture website
https://aarc-project.eu/blueprint-architecture/

[AARC-BPA-2016] MJRA1.4: First Draft of the Blueprint Architecture
https://aarc-project.eu/wp-content/uploads/2016/08/MJRA1.4-First-Draft-of-the-Blueprint-Architecture.pdf

[AARC-BPA-2017] AARC Blueprint Architecture
https://aarc-project.eu/wp-content/uploads/2017/04/AARC-BPA-2017.pdf

[AARC-CILogon] AARC wiki page: CILogon TTS pilot
https://wiki.geant.org/display/AARC/CIlogon+TTS+pilot

[AARC-CO-SSH] AARC wiki page: COmanage OpenConext ssh access
https://wiki.geant.org/display/AARC/COmanage+OpenConext+ssh+access

[AARC-DNA3.1] Deliverable DNA3.1: Differentiated LoA recommendations for policy and practices of identity and attribute providers
https://aarc-project.eu/wp-content/uploads/2017/04/DNA3.1-Differentiated-Assurance.pdf

[AARC-DJRA1.1] DJRA1.1: Analysis of user community and service provider requirements
https://aarc-project.eu/wp-content/uploads/2015/10/AARC-DJRA1.1.pdf

[AARC-JRA1.4A] Guidelines on expressing group membership and role information
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4A.pdf

[AARC-JRA1.4B] Guidelines on attribute aggregation
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4B.pdf

[AARC-JRA1.4C] Guidelines on token translation services
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4C.pdf

[AARC-JRA1.4D] Guidelines for credential delegation
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4D.pdf

[AARC-JRA1.4E] Best practices for managing authorisation
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4E.pdf

[AARC-JRA1.4F] Guidelines on non-browser access
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4F.pdf

[AARC-JRA1.4G] Guidelines for implementing SAML authentication proxies for social media identity providers
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4G.pdf

[AARC-JRA1.4H] Account linking and LoA elevation use cases and common practices for international research collaboration
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4H.pdf

[AARC-JRA1.4I] Best practices and recommendations for attribute translation from federated authentication to X.509 credentials
https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4I.pdf

[AARC-MJRA1.2] MJRA1.2: Design for Deploying Solutions for "Guest Identities"
https://aarc-project.eu/wp-content/uploads/2016/06/MJRA1.2-Design-for-Deploying-Solutions-for-Guest-Identities.pdf

[AARC-SA1-AMP] Attribute Management Pilot wiki
https://wiki.geant.org/display/AARC/AttributeManagementPilot

[AARC-SA1-SCP] AARC wiki page: SocialIDCockpitPanel
https://wiki.geant.org/display/AARC/SocialIDCockpitPanel

[AARC-WTS-SSH] AARC wiki page: WaTTS SSH plugin – SSH access using OIDC login
https://wiki.geant.org/display/AARC/WaTTS+SSH+plugin+-+SSH+access+using+OIDC+login

[APPLE-1] Apple: Using app-specific passwords
https://support.apple.com/en-gb/HT204397

[CIGETCERT] Description of cigetcert tool
https://cdcvs.fnal.gov/redmine/projects/fermitools/wiki/cigetcert

[CILogonECP] Description of CILogon ECP profile
http://www.cilogon.org/ecp

[DYNAF] Dynafed – The Dynamic Federation Project web page
http://lcgdm.web.cern.ch/dynafed-dynamic-federation-project

[eduGAIN] eduGAIN website
https://www.geant.org/Services/Trust_identity_and_security/eduGAIN

[EGI-AAI] EGI AAI wiki page
https://wiki.egi.eu/wiki/AAI

[EGI-LOA] EGI wiki page: AAI guide for SPs – Level of Assurance
https://wiki.egi.eu/wiki/AAI_guide_for_SPs#Level_of_Assurance

[ELIXIR-AAI] ELIXIR AAI documentation web page
https://www.elixir-europe.org/services/compute/aai

[EUDAT-Attr] EUDAT Attribute metadata - background
http://doi.org/10.23728/b2share.20c1c0c8ba254e768fbcb67724918936

[EUDAT-B2ACCESS] EUDAT B2ACCESS web page
https://www.eudat.eu/services/b2access

[GDPR] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)
http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679

[GEANT-eduTEAMS] GÉANT eduTEAMS web site
https://www.geant.org/Innovation/eduteams

[GITHUB-1] GitHub web interface: Creating a personal access token for the command line
https://help.github.com/articles/creating-an-access-token-for-command-line-use/

[GOOGLE-1] Google: Sign in using App Passwords
https://support.google.com/accounts/answer/185833?hl=en

[I2-EP] eduPerson Object Class Specification
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html

[I2-EPA] eduPersonAffiliation description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonAffiliation

[I2-EPAs] eduPersonAssurance description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonAssurance

[I2-EPE] eduPersonEntitlement description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonEntitlement

[I2-EPPN] **e**duPersonPrincipalName description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonPrincipalName

[I2-EPO] eduPersonOrcid description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html#eduPersonOrcid

[I2-EPSA] eduPersonScopedAffiliation description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonScopedAffiliation

[I2-EPUI] eduPersonUniqueId description
http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201310.html#eduPersonUniqueId

[INDIGO-IAM] INDIGO Identity and Access Management web page
https://www.indigo-datacloud.eu/identity-and-access-management

[IRODS] iRODS Docs – Tickets (Guest Access)
https://docs.irods.org/4.2.0/system_overview/users_and_permissions/#tickets-guest-access/

[MACAR] A. Birgisson et al., Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud, Network and Distributed System Security Symposium, 2014
https://www.internetsociety.org/doc/macaroons-cookies-contextual-caveats-decentralized-authorization-cloud/

[MACE] MACE website
https://www.internet2.edu/communities-groups/middleware/middleware-architecture-committee-education-mace/

[MACE-SR] "Information for organisations requesting a delegated namespace"
https://www.internet2.edu/products-services/trust-identity/mace-registries/#service-registries

[NISTIR-8112] Attribute Metadata: A Propose Schema for Evaluating Federated Attributes, NIST Internal Report 8112
https://pages.nist.gov/NISTIR-8112/NISTIR-8112.html

[OIDC-GUIDE] OpenID Connect Basic Client Implementer's Guide 1.0 – draft 37
http://openid.net/specs/openid-connect-basic-1_0.html

[OIDC-Iss] Issuer Identifier description
http://openid.net/specs/openid-connect-core-1_0.html#IssuerIdentifier

[OIDC-Sub] Subject Identifier Types description
http://openid.net/specs/openid-connect-core-1_0.html#SubjectIDTypes

[OIDCre] REFEDS OpenID Connect for Research and Education Working Group
https://wiki.refeds.org/display/GROUPS/OIDCre

[OIDCre-SAML-OIDC] REFEDS wiki page Mapping SAML attributes to OIDC Claims
https://wiki.refeds.org/display/GROUPS/Mapping+SAML+attributes+to+OIDC+Claims

[ORCID] ORCID website
https://orcid.org/

[ORCID-API] ORCID web page: The ORCID API
https://orcid.org/organizations/integrators/API/

[ORCID-CC] ORCID web page: Collect & Connect
https://orcid.org/content/collect-connect/

[RCauth] RCauth website
http://rcauth.eu

[REFEDS-AWG]	REFEDS wiki page: Assurance Working Group https://wiki.refeds.org/display/GROUPS/Assurance+Working+Group
[REFEDS-RS]	REFEDS web page: Research and Scholarship Entity Category https://refeds.org/category/research-and-scholarship
[REFEDS-RS-1]	REFEDS wiki page: "Are SPs allowed to request attributes other than R&S attributes?" https://wiki.refeds.org/display/ENT/Research+and+Scholarship+FAQ#ResearchandScholarshipFAQ-AreSPsallowedtorequestattributesotherthanR&Sattributes?
[SAML-ECP-1]	Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS, March 2005 http://docs.oasis-open.org/security/saml/v2.0/
[SIRTFI]	Sirtfi website https://refeds.org/sirtfi
[SIRTFI-1]	T. Barton et al., A Security Incident Response Trust Framework for Federated Identity (Sirtfi) https://refeds.org/wp-content/uploads/2016/01/Sirtfi-1.0.pdf
[SURF-LOA]	SURFnet wiki page: Connecting your SP to SURFconext Strong Authentication https://wiki.surfnet.nl/display/surfconextdev/Connecting+your+SP+to+SURFconext+Strong+Authentication
[SWITCH-IMO]	isMemberOf description https://www.switch.ch/aai/support/documents/attributes/ismemberof/index.html
[UMBRELLA]	GÉANT eduTEAMS Umbrella pilot wiki page https://wiki.geant.org/display/gn42jra3/Umbrella
[VOMS-PROC]	OGF VOMS attribute PROCessing Working Group https://redmine.ogf.org/projects/voms-proc-wg

Glossary

AA	Attribute Authority
AAI	Authentication and Authorisation Infrastructure
AARC	Authentication and Authorisation for Research and Collaboration
API	Application Programming Interface
AS	Authorisation Server
AUP	Acceptable Use Policy
AuthN	Authentication
AuthZ	Authorisation
BPA	Blueprint Architecture
CA	Certification Authority
DN	Distinguished Name
ECP	Enhanced Client Protocol
EGI	European Grid Infrastructure
EI	e-Infrastructures
eP	eduPerson
ePA	eduPersonAffiliation
ePE	eduPersonEntitlement
ePO	eduPersonOrcid
ePPN	eduPersonPrincipalName
ePSA	eduPersonScopedAffiliation
ePUID	eduPersonUniqueId
FIM	Federated Identity Management
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GDPR	General Data Protection Regulation
GSI	Grid Security Infrastructure
GSS-API	Generic Security Service Application Program Interface
GUI	Graphical User Interface
HO	Home Organisation
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
Iaas	Infrastructure as a Service
IAM	Identity Access Management
IdP	Identity Provider
IR	Incident Response

iRODS	Integrated Rule-Oriented Data System
JRA1	Joint Research Activity 1, Architectures for an integrated and interoperable AAI
JSON	JavaScript Object Notation
PKI	Public Key Infrastructure
LCMAPS	Local Credential Mapping Service
LDAP	Lightweight Directory Access Protocol
LIGO	Laser Interferometer Gravitational-Wave Observatory
LoA	Level of Assurance
MACE	Middleware Architecture Committee for Education
NREN	National Research and Education Network
OAuth2	The industry-standard protocol for authorisation
OIDC	OpenID Connect
OIDCre	OpenID Connect for Research and Education Working Group
OP	OpenID Connect Provider
OS	Operational Security
PAM	Pluggable Authentication Modules
POSIX	Portable Operating System Interface
PR	Participant Responsibilities
RBAC	Role-Based Access Control
RC	Research Collaboration
RCauth	The white-label Research and Collaboration Authentication CA Service for Europe
RI	Research Infrastructures
R&E	Research and Education
R&S	Research and Scholarship
RP	Relying Party
SA1	Service Activity 1 Pilots
SAML	Security Assertion Markup Language
SCIM	System for Cross-domain Identity Management
SFTP	Secure File Transport Protocol
SHA-256	Secure Hash Algorithm 2, with 256 bits hash value
Sirtfi	Security Incident Response Trust Framework for Federated Identity
SP	Service Provider
SSH	Secure Shell protocol
SSO	Single Sign-On
SSOT	Single Source of Truth
SVN	Subversion
TLS	Transport Layer Security
TTS	Token Translation Service
UC	User Community

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VLE	Virtual Learning Environment
VM	Virtual Machine
VO	Virtual Organisation
VOMS	Virtual Organisation Membership Services
VOOT	Protocol for dynamic exchange of group and authorisation data
X.509	X.509 Public Key Infrastructure Standard
XML	eXtensible Markup Language