31-12-2015

Milestone MJRA1.1: Existing AAI and available technologies for federated access.

Milestone MJRA1.1

Contractual Date:	01-01-2016
Actual Date:	31-12-2015
Grant Agreement No.:	653965
Work Package:	JRA1
Task Item:	1
Lead Partner:	EGI.eu
Document Code:	MJRA1.1
Authors:	P. Solagna (EGI.eu), Christos Kannelopoulos (GRNET), N. Liampotis (GRNET), M. Hardt (KIT), M. Salle (Nikhef), S. Paetow (Jisc), M. Malavolti (GARR), N. Van Dijk (SURFnet), J Jensen (STFC), I. Liabotis (GRNET), M. Jankowski (PSNC), S. Memon (Juelich), M. Prochazka (CESNET), B. Oshrin (SURFnet), B. Monticini (GARR), H. Short (CERN), U. Stevanovich (KIT)

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 summarises the technologies and solutions available to implement AAI, focusing on the software most common in the research and education (R&E) environment, which features are more likely to fulfil the use cases of the R&E communities.

Table of Contents

Table of Tables

Executive Summary

The AARC project activities have consisted of research of technology and interviews with the stakeholders to understand their requirements and their solutions. In addition to the information gathered in the first part of the project, the AARC consortium already had a large knowledgebase based on previous experiences at the beginning of the project.

This document summarises the technologies and solutions available to implement AAI, focusing on the software most common in the research and education (R&E) environment, which features are more likely to fulfil the use cases of the R&E communities.

Both standards and software implementing the standards are individually analysed, and at the end of the document tools and software are compared in tables to make easier for a potential user to choose which one best fits their use case.

The readers who are reading the document to choose one or few tools to implement their use case are invited to first check the comparison tables, and then use the information in section 3 to get more information about the software that suits their requirements.

The milestone does not select preferred solutions, since depending on the use case users or communities may choose different tools to implement their AAI capabilities. This document will provide an overview, hopefully covering most of the interesting aspects that can be considered to choose a software solution for AAI use cases, to facilitate the architectural design of AARC, and the development of AAI capabilities by infrastructures and user communities.

1 Introduction

This milestone is a summary of the available technologies to support AAI use cases that are used or that can be used by the research and education community. The content is structured in the following sections.

The “Relevant Standards” section provides an overview of the standards relevant for the AAI use cases analysed by AARC[1]. The protocols and standards described in the section cover all the technologies used by the research infrastructures and e-infrastructures active in the international research and education ecosystem.

The “Authentication and authorization technologies and tools” section collects a summary description of the software and tools that are used by the communities, or that directly address the use cases of research and education. Most of the services have been mentioned in the surveys and the interviews carried out among the AARC stakeholders in the first part of the project, or are widely used tools that are relevant for the communities. The section is not structured by use case, since many tools have functionalities that cover more than one use case. To avoid repetitions the section has been kept with a flat structure.

[1] AARC project stakeholders include representatives from: e-infrastructures, NRENs, research and higher education institutions and libraries.

The “Comparison of authentication and authorization technologies” section is structured in sub-sections for every use case or feature, and the tools supporting the use cases are compared in comparison tables. Here the comparison is made versus relevant features both common and specific for the use case. The goal of the tables is to provide a quick overview how every tool implements specific functionalities or requirements, or how the tools implement the most relevant features of the use cases.

Every section describing the individual standard or the technology will conclude with an overview of the requirements identified in the DJRA1.1 deliverable[2]. Among the many requirements captured by the previous document, the following are the most relevant for AAI technologies and standards:

- User and Service Provider friendliness
- User-managed identity information
- Different Levels of Assurance
- Community-based authorisation
- Attribute aggregation / Account linking
- User groups and roles
- Step-up authentication
- Browser & non-browser based federated access
- Delegation
- Federation solutions based on open and standards-based technologies
- Social media identities
- Integration with e-Government infrastructures

Other requirements, for example policy related ones, are not directly implemented by the services, and therefore could not be applied to the content of this milestone document.

This document is not a fully comprehensive summary of all the standards and technologies that can enable authentication and authorisation. The focus of this milestone is to provide information and to have a review of the standards and technologies that are most relevant for the research & education community.

In the first part of the project, AARC gathered surveys about the AAI status and requirements of several communities and e-infrastructures, as well as performed interviews with representatives from these e-infrastructures and research communities. The technical solutions and the technologies described in this document are the solutions of choice that the AARC stakeholders are considering using to implement their AAI, plus the tools and software that can support the R&E requirements based on the experience of the authors contributing to the document.

[2] Summary of DJRA1.1 requirements https://wiki.geant.org/display/AARC/Collected+Requirements

2 Relevant standards

2.1 SAML2

The Security Assertion Markup Language 2.0 (SAML2) is an open standard and one of the key technologies for federated identity. It enables single sign-on (SSO), which is used to decouple the authentication and authorization process from an application. It means that a user can use a single credential to access multiple applications.

User credentials are not stored in these applications; they are stored in trusted Identity Providers, which handle authentication and authorization processes by themselves. SAML2 is used to exchange this authentication and authorization data, commonly called an assertion. Besides being a technical means of transporting the users' data, SAML also provides a technical mechanism for building trust between entities. Identity Providers and Service Providers can establish relations by leveraging digitally signed metadata published by a trusted third party. This mechanism scales well, as is shown for example in eduGAIN[3], the global confederation for Research and Education.

2.1.1 Entities involved in the standard

Identity Provider (IdP)
Attribute Authority (AA)
Service Provider (SP)

2.1.2 What does it imply for the user

- Users can access mainly web-based services with their institutional credentials, a username and password.
- Between SAML-enabled Service Providers, users can get single sign-on; once user is logged in, no further login is required for authenticating at subsequent Service Providers.

2.1.3 What does it imply for the service provider

Assertions are in XML format. One assertion represents a set of information about an identity, made by a SAML entity. Assertions are exchanged between an identity provider, the entity that is able to verify user's credentials and a service provider, an entity that needs the identity provider to verify user's credentials.

According to the request-reply model of SAML, there are 3 kinds of assertions: the authentication assertion, the attribute assertion and the authorization assertion:

[3] http://edugain.org

- The authentication assertion asserts that the identity was authenticated by an authentication mechanism at a certain time.
- The attribute assertion asserts that the identity was associated with the specified attributes (name, surname, etc.).
- The authorization assertion contains proof that the identity has been authorized to access a specific resource with specific rights.

Authorization by Services is done based on group and attribute information that can be carried by SAML2 in two ways:

- As part of authentication: SAML attributes transported as part of the Authentication statement. Many attributes in the commonly used eduPerson schema actually represent roles and authorisations:
  - eduPerson{Scoped}Affiliation provides a fixed naming scheme for classing people into groups like student, faculty, member, etc.
  - eduPersonEntitlement is used to express roles and rights and may represent groups of people.
  - eduMembers', IsMemberOf is commonly used to express group memberships
  - Additionally, SAML allows arbitrary attributes to be used to express group membership.
- SAML Attribute Query: This protocol provides a back channel for querying attribute and thus also group information from a SAML Attribute Authority. The authorisation management between the SAML Attribute Authority and the requesting Services is based on the same mechanisms as between Identity Providers and Service Providers (SAML metadata). This mechanism is rather coarse, and may therefore not serve all use-cases.

It should also be noted that SAML supports a variety of security mechanisms at transport- and message-level, namely SSL 3.0 or TLS 1.0 for transport-level security and XML Signature and XML Encryption for message-level security.

2.1.4 Relevant RFCs and documents

For more information about SAML2, please see https://www.oasis-open.org/standards#samlv2.0.

SAML2 is widely used in Research & Education national federations.

2.1.5 Supported requirements

- User and Service Provider friendliness[4]
- Different Levels of Assurance
- Community-based authorisation
- User groups and roles
- Step-up authentication
- Federation solutions based on open and standards-based technologies

[4] From a user point of view it is user friendly as login is as simple as providing username/password. For service providers it can be more complicated than other standards

2.2 X.509 / X.509 proxies

2.2.1 Entities involved in the standard

X.509 is an ITU-T standard for a public key infrastructure (PKI), also known as PKIX (PKI X509). The standard defines formats for:

- Public key certificate format
  Digital document that proves the ownership of the duplet: public key - private key. The public key certificate is signed by a Certification Authority (CA), which certifies that the information in the public key certificate is correct. Usually this means that the owner of the public-private key duplet is the one reported in the certificate information.
- Certificates revocation list (CRL)
  Used by services to reject invalid certificates. These lists are usually published by the CAs and automatically downloaded by the service providers that are using the PKIX. Status of the certificates can also be retrieved through the OCSP[5] protocol by querying the CAs that support the protocol.
- Attribute certificate
  It is a certificate carrying the attributes used for the authorization. The certificate is signed with the user's private key to confirm user's identity, and signed by an attribute authority to confirm the validity of the authorization attributes.
- Certificate path validation algorithm
  To verify that the public key certificate is valid in a X.509 infrastructure.

Other entities are:

- The Certification Authority (CA), who creates and signs the certificates
- The Registration Authority (RA), who verifies the user identity and approves the user request submitted to the CA
- Attribute authority, which signs the attribute certificate

By protecting the private key of the certificate with a password, certificates can be considered an example of two-factor authentication.

In an X.509 based authentication, the user is uniquely identified by the certificate subject DN (Distinguished Name), which does not change when a new certificate is renewed. Within the set of CAs included in the IGTF (International Grid Trust Federation) distributions, the subject DNs are guaranteed to be unique across CAs.

For delegating user authentication, a special type of certificate has been introduced, a so-called proxy certificate, standardised in RFC3820, where the certificate is signed by the (private key belonging to the) user's end-entity certificate (EEC) or another proxy certificate, instead of the CA. These proxy certificates are typically short-lived to reduce the impact in case of theft and thereby are not required to have CRLs. By forming chains of proxy certificates the user can delegate credentials to remote hosts. In practice proxy certificates are often combined with attribute certificates (signed by the private key belonging to the previous certificate level in the chain), in particular as used by VOMS.

[5] RFC6960, https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol

2.2.2 What does it imply for the user

The user, to obtain and use credentials from a PKIX, needs to contact a CA to get the certificate, often this requires an identity validation done by a Registration Authority (RA). While the CA also needs to perform technical activities, the RA is a mostly administrative activity. This allows many geographically distributed RAs to be associated with a single CA, which makes the PKIX more scalable. The user also needs to know which attribute authorities are supposed to sign the attribute certificate.

The user needs to safely store the private key belonging to the personal certificate, protecting it with password and proper file access permissions in case of a shared machine. Usually a personal certificate has a lifetime of 1 year. Before the end of the validity period a user can request a certificate renewal by signing the request with the private key belonging to the existing certificate.

By importing the certificate / private key into the browser, the user can also use the certificate as a form of single sign on.

For command line / non-web access to resources the user would typically create a proxy certificate for which a number of tools are available.

2.2.3 What does it imply for the service provider

The service provider, to access a PKIX, needs to install in the services the CA root certificates (to verify the CA signature in the public key certificates), configure the attribute authorities by identifying them with their host certificate DN, or other similar information, to verify the signature of these attribute certificates, and periodically (usually every 6 hours) download the CRLs from all the CAs the service supports.

X.509 certificates can be used both to authenticate users and services, in fact X.509 certificates are the most common authentication mechanism for hosts and services, for example for the HTTPS protocol, and form the basis for TLS.

In the services accessed with X.509 credentials, the service provider receives all the information provided by the PKIX with the attribute certificate submitted by the user and can verify this locally, given that the service is querying the CA for certification validity information (e.g. for certificate revocation lists). Therefore, service providers do not need to interact directly with any third party (CA or Attribute authority).

X.509 is widely used by e-infrastructures where command-line access to resources is critical.

2.2.4 Relevant RFCs and documents

- RFC 5280
- RFC 3820

2.2.5 Supported requirements

- Third party attribute management
- Flexible and scalable attribute release policies
- Browser & non-browser based federated access

- Delegation
- Up-to-date identity information

2.3 OpenID Connect/OAuth 2.0

OAuth 2.0 (referred also as OAuth in this document) is an open standard for authorisation. OpenID Connect (OIDC) is a standard for single sign-on and identity provision, which adds an identity layer on top of the OAuth 2.0 protocol.

2.3.1 Entities involved in the standard

OAuth enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf. Instead of requiring the resource owner to share their credentials with the third party, OAuth allows issuing a different set of credentials than those of the resource owner when a client requests access to resources controlled by the resource owner and hosted by the resource server. More specifically, the client obtains an access token, i.e. a string denoting a specific scope, lifetime, and other access attributes, which can then be used to access the protected resources hosted by the resource server.

OpenID Connect allows clients to verify the identity of an end-user based on the authentication performed by an authorisation server, as well as to obtain profile information about the end-user. Compared to other popular federation approaches, such as SAML and OpenID 1.0/2.0, its main strengths include usability and simplicity. In OpenID Connect, client applications receive the user's identity encoded in a secure JSON Web Token (JWT), called an ID token.

Apart from being portable, such ID tokens support a wide range of signature and encryption algorithms. In this context, the ID token resembles the concept of an identity card, in a standard JWT format, which is signed by the OpenID Connect Provider (OP). To obtain one, the client needs to send the user to their OP with an authentication request. The returned token asserts the identity of the user, called the subject in OpenID Connect (sub). Each token specifies both the issuing authority (iss) and the particular audience, i.e. client (aud), for which it was generated. It may specify when (auth_time) and how, in terms of strength (acr), the user was authenticated. It may include additional requested details about the subject, such as their name and email address. Being digitally signed, it can be verified by the intended recipients. It may optionally be encrypted for confidentiality. The ID token statements, or claims, are packaged in simple JSON objects, thus supporting web applications, as well as native / mobile apps.

2.3.2 What does it imply for the user

OAuth separates the role of the client from that of the resource owner, i.e. the end-user, and provides the following advantages over the traditional client-server authentication/authorisation model:

- Users have the ability to restrict duration of access and/or provide access to only a limited subset of resources.
- Users can revoke access to an individual third party without revoking access to all third parties.

OAuth is thus commonly used to allow users to sign into third party websites using their Google, Facebook or Twitter accounts without exposing their password.

2.3.3 What does it imply for the service provider

OAuth provides the following benefits for service providers:

- Third-party applications are not required to store the resource owner's credentials, typically their password in clear-text format.
- Servers are not required to support password authentication

It should be noted that the version 2.0 specification replaces, and is not backward compatible with, the original OAuth 1.0 protocol described in RFC 5849. OAuth 2.0 does not support native encryption capabilities; thus it relies on the SSL/TLS protocols to provide encryption of the sensitive data being exchanged between parties.
While OpenID Connect is most commonly known for its adoption by Social Media service providers for sign-in purposes, it is also gaining traction in enterprise-targeted services, such as Windows Azure Active Directory (WAAD), Ping Federate and PingAccess. OpenID Connect can be integrated with provisioning protocols such as System for Cross-domain Identity Management (SCIM). In addition, it can provide ISO/IEC 29115[6] Level of Assurance 1 to 4, through the use of the optional Authentication Context Class (acr) claim. Finally, it is worth mentioning that while OpenID Connect has many architectural similarities to OpenID 2.0, the identifier format is different and thus service providers need to migrate those user identifiers to continue serving these users.

2.3.4 Relevant RFCs and documents

- Hammer-Lahav, E., “The OAuth 1.0 Protocol”, RFC 5849: https://tools.ietf.org/html/rfc5849
- Hardt, D., “The OAuth 2.0 Authorization Framework”, RFC 6749: https://tools.ietf.org/html/rfc6749
- Sakimura, N., Bradley, J, Jones, M., de Medeiros, B., Mortimore, C., “OpenID Connect Core 1.0”, http://openid.net/specs/openid-connect-core-1_0.htm
- Sakimura, N., Bradley, J., Agarwal, N., Jay, E., "OpenID 2.0 to OpenID Connect Migration 1.0", http://openid.net/specs/openid-connect-migration-1_0.html

2.3.5 Supported requirements

The following are the supported requirements:

- User and Service Provider friendliness: Users are able to control access to their resources without revealing password. Service Providers are not required to store the resource owner's credentials.
- Different Levels of Assurance: OpenID Connect can provide ISO/IEC 29115 Level of Assurance 1 to 4, through Authentication Context Class claims
- Community-based authorisation: A community can operate an OAuth Provider for authorisation purposes (see also R6)
- Attribute aggregation / Account linking: OAuth can be used to enable the SP to establish a session with an Attribute Authority. The SP thus becomes an OAuth Consumer and the Attribute Authority an OAuth Provider.

[6] International Organization for Standardization, “ISO/IEC 29115:2013 - Information technology - Security techniques - Entity authentication assurance framework,” ISO/IEC 29115, March 2013.

- User groups and roles: OpenID Connect can carry group membership info through the use of additional scope values (e.g. “memberOf” scope).
- Step-up authentication
- Browser & non-browser based federated access: OAuth/OpenID supports web-based applications, as well as native apps and mobile applications
- Delegation: OAuth defines a delegation protocol that can be used to convey authorisation decisions across a network of web-enabled applications and APIs
- Federation solutions based on open and standards-based technologies
- Social media identities: OAuth/OpenID are widely used by Social media identity providers

2.4 XACML

The eXtensible Access Control Markup Language (XACML) is an OASIS standard that defines a declarative access control policy language expressed in XML and a processing model describing how to evaluate access requests according to the rules defined in policies. XACML is an Attribute Based Access Control system (ABAC), whereby attributes associated with a user, action or resource are used for deciding whether a given user may access a given resource in a particular way. Role-based access control (RBAC), being a specialisation of ABAC, can also be implemented with XACML.

2.4.1 Entities involved in the standard

The standard proposes a reference architecture describing the various entities involved and their interactions.

As per the architecture, when an access request is sent from the subject to perform a specific action on the object, it is intercepted by the **Policy Enforcement Point (PEP)**, which transforms the business/application request to a XACML request.

The PEP sends the XACML request to the **Policy Decision Point (PDP)**, which uses the information provided in the XACML request and the rules set in the policy to decide whether the request should be allowed or not.

The PDP uses the **Policy Information Point (PIP)** to look up attributes that are referenced in the XACML policy and hence needed to make a decision on the XACML request. In theory, PIPs can be any source of attribute information, including, but not limited to, LDAP directories, SQL databases or even CSV files. For example, if a particular XACML policy uses the subject attribute of “role” in it, but the XACML request created by the PEP is only able to provide the subject attribute “username”, an LDAP PIP may be used to look up the role associated with the username of the subject in question and then use that value to evaluate the request against the XACML policy.

The **Policy Administration Point (PAP)**, as the name suggests, is the architectural entity that is used to manage policies the PDP later evaluates. More specifically, it allows authoring, deployment, change management etc. of XACML policies.

It should be noted that the entities described above are not XACML-specific. PDP, PEP, and PIP are all defined in RFC 2904: AAA Authorization Framework.

2.4.2 What does it imply for the user

XACML bears no implications for end-users. Access requests for resources (either web- or non-web-based) must be intercepted by the service provider's PEP, which transforms the business/application request to a XACML request before sending it to the PDP (see below).

2.4.3 What does it imply for the service provider

The service provider must implement a PEP. For instance, a PEP may be part of a remote-access gateway, part of a web server or part of an email user-agent, etc. It is unrealistic to expect that all PEPs issue decision requests to a PDP in a common format. Nevertheless, a particular policy may have to be enforced by multiple PEPs. It would be inefficient to force a policy author to write the same policy in several different ways in order to accommodate the format requirements of each PEP. Similarly, attributes may be contained in various envelope types (e.g. X.509 attribute certificates, SAML attribute assertions). Therefore, there is a need for a canonical form of the request and response handled by an XACML PDP. This canonical form is called the XACML context. Its syntax is defined in XML schema.

Naturally, XACML-conformant PEPs may issue requests and receive responses in the form of an XACML context. But, where this is not the case, an intermediate step is required to convert between the request/response format understood by the PEP and the XACML context format understood by the PDP.

The benefit of this approach is that policies may be written and analysed independently of the specific environment in which they are to be enforced.

In the case where the native request/response format is specified in XML Schema (e.g. a SAML-conformant PEP), the transformation between the native format and the XACML context may be specified in the form of an Extensible Stylesheet Language Transformation (XSLT).

2.4.4 Relevant RFCs and documents

- XACML Version 1.0, OASIS Standard, February 2003, http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf.
- XACML Version 1.1, OASIS Committee Specification, August 2003, http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf.
- XACML Version 2.0, OASIS Standard, February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.
- XACML Version 3.0, OASIS Standard, January 2013, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.
- SAML 2.0 Profile of XACML, Version 2.0, OASIS Committee Specification 01, August 2010, http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cs-01-en.pdf.
- REST Profile of XACML v3.0 Version 1.0, OASIS Committee Specification 01, April 2013, http://docs.oasis-open.org/xacml/xacml-rest/v1.0/xacml-rest-v1.0.pdf.
- RFC 7061: eXtensible Access Control Markup Language (XACML) XML Media Type
- RFC 2904: AAA Authorization Framework.

2.4.5 Supported requirements

- Different Levels of Assurance
- Community-based authorisation: PDPs/PAPs can be operated by communities
- Attribute aggregation / Account linking
- User groups and roles
- Browser & non-browser based federated access: A PEP can transform the business/application request for either web- or non-web-based resources into a XACML request before sending it to the PDP
- Delegation
- Federation solutions based on open and standards-based technologies

2.5 ABFAB (Moonshot)

Application Bridging for Federated Access Beyond Web (ABFAB) is an architecture for providing federated access management to applications using GSS-API and SASL. RFC7055 describes the core mechanism for federated authentication using a modified RADIUS infrastructure, while RFC7056 describes the GSS-API attributes for using the naming extensions to access information bound in a SAML message provided by the core mechanism. Several RFC drafts refer to the overall architecture (ABFAB-ARCH), the usability constraints and considerations for ABFAB clients (ABFAB-USABILITY-UI-CONSIDERATIONS) and the use cases for ABFAB (ABFAB-USECASES).

Due to the conglomeration of several Internet technologies whose terms for various entities involved in this standard, each entity will specifically refer to the most commonly used terms in AAI.

2.5.1 Entities involved in the standard

The Client is the actual client device that initiates the ABFAB AA request. This may be a laptop or a console system, a mobile device or a service. The client requires the ABFAB GSS-API mechanism to be installed, and in the case where the operating system does not have intrinsic support for client credential storage, UI client credential management software. In SAML terms, the client is the Subject.

The Service is the actual service that the client attempts to access. This may be any service that supports the GSS-API. The service also requires the ABFAB GSS-API mechanism to be installed, and it must have a configured connection to an RP Proxy (see next). In SAML terms, this, together with the RP Proxy, constitutes the Service Provider.

The RP Proxy is the gateway service to the ABFAB RADIUS infrastructure. This service accepts ABFAB RADIUS requests over RadSec (preferred) or UDP and either processes them locally (in the case of a combined RP Proxy/IdP), or requests information from the Trust Router (see next) to be able to contact the Identity Provider (see next) to complete the AA request. It also processes the AA response based on locally established rules before returning it to the service. The RP Proxy requires RADIUS server software to support dynamic realm look-up by querying the Trust Router. Currently the only server software with this support is FreeRADIUS. It also requires the ABFAB temporary identity service software to be installed.

The Identity Provider (IdP) is the service that provides AA services for its users. The Identity Provider requires RADIUS server software with support for dynamic realm lookup. Currently the only server software with this support is FreeRADIUS. It also requires the ABFAB temporary identity service software to be installed. This

service may also be connected to a SAML attribute authority or identity source (an Issuer) to be able to provide an AA response that includes an attribute statement for use by the RP Proxy or the Service.

The Trust Router is the trust service that underpins the wider ABFAB architecture. This service is trusted by the organisations participating in the ABFAB infrastructure, and it maintains the registry of trust relationships in a similar fashion as the metadata in a SAML federation. In SAML terms, the trust service acts as the Federation. The trust service is usually maintained by an NREN or large infrastructure (such as a research infrastructure). The only trust router service available currently is the Jisc Assent service, with GÉANT likely to be the provider of the second service once pilots are complete.

A Community of Interest (COI) is a concept maintained at the Trust Router that allows groups of service provider and identity providers to build communities with specific requirements (such as policy, LoA, etc.). By specifying at the RP Proxy which community of interest the Service is part of, the Trust Router is able to constrain identity AA requests further, while the identity providers may provide different information depending on the COI that is passed along with the AA request. Identity providers can belong to any number of COIs, but currently, RP Proxies can only be part of one COI at any time.

2.5.2 What does it imply for the user

The user must install the ABFAB mechanism and credential management software (collectively called the Moonshot client) onto their client device. The user is issued with an ABFAB credential by their organisation (it may be provisioned for them onto their client device). Ideally this credential is the same as the user's existing SAML federation (eduGAIN or otherwise) credential.

The user should, to be able to take advantage of ABFAB services, use software that fully complies with the GSS-API standards (not just Kerberos-style single-trip authentication) and that has been verified for use with Moonshot. A non-exhaustive list of compatible client software is available from the Moonshot wiki at https://wiki.moonshot.ja.net/.

2.5.3 What does it imply for the service provider

The service provider must install the Moonshot client and Moonshot-compliant server software to be able to take advantage of ABFAB. The service provider must also install an RP Proxy and apply to Jisc (or in the future, GÉANT) to join the trust router infrastructure. The service provider should also join a community of interest where appropriate, and process received AA responses according to its local policy requirements and the community of interest requirements it is part of.

2.5.4 What does it imply for the identity provider

The identity provider must install the Moonshot client and Moonshot-compliant FreeRADIUS software to be able to take advantage of ABFAB. The identity provider must also apply to Jisc (or in the future, GÉANT) to join the trust router infrastructure. Where appropriate/applicable, the identity provider must configure FreeRADIUS to return AA responses appropriate for the communities of interest it is part of.

2.5.5 Relevant RFCs and documents

- RFC7055
- RFC7056
- draft-ietf-abfab-arch-13
- draft-ietf-abfab-usability-ui-considerations-03
- draft-ietf-abfab-usecases-05

2.5.6 Supported requirements

- Community-based authorisation
- Federation solutions based on open and standards-based technologies
- Browser & non-browser based federated access

2.6 SCIM

System for Cross-domain Identity Management[7] (SCIM) is an IETF standard for account management and synchronisation. It describes parties that manage accounts in remote services (such as those provided in clouds) as well as the synchronisation of accounts between providers. SCIM defines use cases (RFC7642), a schema (RFC7643) and a means for entities to exchange information based on the schema (RFC7644) over HTTP. The schema consists of a *core* and allows for extensions, i.e. based on the type of resource which is being managed, and the schema by default is rendered in JSON (RFC7159). Schemas defined by the group (core, user, enterprise user) are defined in a namespace of urn:ietf:params:scim:schemas but do not have OIDs. The protocol (using HTTP) allows for CRUD of user accounts and their attributes; the latter include one for 'password' (flagged as `writeOnly' for security reasons), thus allowing for passwords to be set or reset over SCIM.

2.6.1 Entities involved in the standard

- Cloud Service Provider: A resource in which identities/accounts are registered and managed
- Enterprise Cloud Subscriber: The actual customer of the CSP, which registers user accounts with the remote cloud service
- Cloud Service User: End user of the services provided in the cloud; it is the CSU whose account details are the subject of SCIM
- Community of Interest: Essentially a diverse and complex infrastructure across which users have common use cases and workflows.

[7] There have been several versions of the acronym; the group started its life late 2011 as Simplified Cloud Identity Management

2.6.2 What does it imply for the user

Users should have consistent account management across all systems. However, the implication is that data about user accounts should be maintained across multiple identity domains.

2.6.3 What does it imply for the service provider

Services must query SCIM endpoints using a REST interface.

2.6.4 Relevant RFCs and documents

- K Li, P Hunt, B Khasnabish, A Nadalin, Z Zeltsan: *System for Cross-domain Identity Management: Definitions, Overview, Concepts, and Requirements*, RFC7642, doi:10.17487/RFC7642
- P Hunt, K Grizzle, E Wahlström, C Mortimore: *System for Cross-domain Identity Management: Core Schema*, RFC7643, doi:10.17487/RFC7643
- P Hunt, K Grizzle, M Ansari, E Wahlström, C Mortimore: *System for Cross-domain Identity Management: Protocol*, RFC7644, doi:10.17487/RFC7644

2.6.5 Supported requirements

- Attribute aggregation/account linking
- User groups and roles

2.7 Kerberos

Kerberos is an authentication protocol using a trusted central authentication service - Key Distribution Center (KDC). Each user and service shares a secret key with the KDC. The KDC issues *tickets* asserting the identity of their bearers, which can be verified by relying parties. A ticket has a limited lifetime and unlike public-key certificates for example, it can be only used for a particular end-service, which is specified in the ticket. Apart from supporting mutual authentication of the peers, the Kerberos protocol also provides means for message encryption or integrity protection. Since a KDC holds a list of all users and services, every authentication among users and services involves contacting the KDC. This feature makes it hard to deploy Kerberos in highly distributed environments since users must be registered with the KDC first. In order to make Kerberos more scalable, the users' space can be divided into administrative groups (realms), served by independent KDCs. The Kerberos cross-realm authentication mechanism allows seamless interoperability among different realms. Kerberos is a widely used mechanism in local security infrastructures operated by many institutions.

2.7.1 Entities involved in the standard

Key Distribution Center (KDC): The authentication and key server for an administrative domain. It has trust relationships established with all client and servers in the domain.
Client: A user or service that initiates authentication exchange. It has to be in possession of a ticket.

Server: A service accepting tickets issued by a trusted KDC

2.7.2 What does it imply for the user

The user has to obtain a ticket before it connects to a service. Usually a Ticket Granting Ticket is obtained, which allows users to obtain subsequent service tickets. The TGT has to be obtained using the user's password or other credentials that are registered with the KDC.

2.7.3 What does it imply for the service provider

Service has to have available a secret shared with the KDC that is used to authenticate tickets presented by the clients. Usually keys are stored on local file systems and have to be securely distributed when a service is being introduced to the domain.

2.7.4 Relevant RFCs and documents

C. Neuman, T. Yu, S. Hartman, K. Raeburn: *The Kerberos Network Authentication Service (V5)*. IETF RFC 4120. 2005.

2.7.5 Supported requirements

- User and Service Provider friendliness
- Access using username/password
- Federation solutions based on open and standards-based technologies

3 Authentication and authorization technologies and tools

3.1 LCAS/LCMAPS (X509)

3.1.1 Overview

The legacy LCAS framework is designed to take an authorization decision based on various credentials as input, e.g. a certificate and/or VOMS credentials, and provide a yes/no decision. LCAS is a framework that can load and run one or more authorization plugins. Most of the plugin functionality has been reproduced by corresponding LCMAPS plugins. LCAS is no longer actively supported except for existing use cases.

The LCMAPS framework is designed to take various credentials as input, e.g. a certificate and/or VOMS credentials, and map them to Unix credentials as output. Unix credentials are the basic POSIX credentials, i.e. User ID, Group ID and Secondary Group IDs. Just like LCAS, LCMAPS is a pluggable framework, but it provides a much more advanced and flexible plugin engine and a wide variety of plugins exist, including plugins to interface with Argus and GUMS.

LCAS/LCMAPS are currently used in many HTC and storage services deployed in the EGI infrastructure, as well as in cloud services federated in EGI. They are also deployed in OSG and are critical components for the WLCG community.

The two frameworks were developed by Nikhef during the EDG, EGEE I-II-III and EMI projects, and are now maintained by Nikhef. They are open source tools, available – among other sources – in the Fedora community repository EPEL, in Debian, Ubuntu and in the EGI UMD distribution.

Via the lcas-lcmaps-gt4-interface library, LCAS and LCMAPS can be used in e.g. gsissh and GridFTP via the gsi-callout mechanism.

gLExec uses LCMAPS for its mapping decision, providing the sudo-like functionality based on X.509 and VOMS credentials.

3.1.2 Features supported by the tool

Support for VOMS credentials
Support for groups and roles within the VOs
Support for X.509 proxies extensions to support robot certificates
Client for Argus and SCAS.
Support for Globus GSI-callout mechanism.
VOMS VO membership allows to group users and hierarchically delegate group membership within the VO

Support for community-based authorisation: through the VOMS extensions, authorization can be based on community attributes (stored on the VOMS server). Using the lcmaps-plugins-vo-ca-ap plugin, it allows for mappings and authorization, which takes into account different levels of assurance.

3.1.3 Supported standards

X.509 (RFC5280, RFC3820), VOMS, SAML2-XACML2 (via plugin).

3.1.4 User interfaces and APIs

Configuration policies and rules are stored in configuration files, to be edited by the system administrators. The tool, being used as a component of a service, does not directly expose interfaces to the user.

LCAS/LCMAPS provides proprietary but stable and open APIs to push credentials and retrieve authorization and mapping decision results.

3.1.5 Support for Virtual Organization (if relevant)

LCAS and LCMAPS enable sites to regulate authorization and user separation based on VO membership and specific roles within these VOs. When a VO is supported by a service, a user which is member can be authorized to access this service, e.g. to submit computational tasks to a cluster or retrieve or store data. All the tasks - where relevant - can be mapped to a pool of local users connected to the VO.

3.1.6 Dependencies with other technologies (libraries, DBs, etc.)

OpenSSL and Globus libraries.
VOMS C API for the VO credentials.

3.1.7 Operational overview (HA, deployment scenarios)

LCAS and LCMAPS are deployed with the services that use the framework for authorization.

3.1.8 Expected level of support

Nikhef supports LCMAPS, its standard plugins and the lcas-lcmaps-gt-interface for bug fixes and relevant new features. LCAS is only supported for security bug fixes.

3.2 Moonshot (RFC7055/7056)

3.2.1 Overview

Moonshot is a standards-based, open source architecture for web and non-web sign-on access within and across organisational boundaries. Moonshot technology touches and uses several other software packages to avoid reinventing the wheel for the sake of it and to make implementation of Moonshot through existing services easier. It also defines a new GSSAPI mechanism, GSS-EAP, which enables RADIUS EAP authentication to be accessed from a GSSAPI-based service.

Moonshot is currently funded by Jisc (JANET) as a project and is released under the BSD licence, although some components, such as the Windows version of the Moonshot GSSAPI mechanism, are closed-source and must be licensed from either Jisc or Painless Security, the primary developers. Users in the educational and research (R&E) space may license the Windows mechanism free of charge.

3.2.2 Features supported by the tool

Moonshot comprises several parts: the client, the service, the RP proxy, the identity provider and the trust router.

The client, comprising the GSSAPI mechanism and the identity manager (on supported platforms), enables the use of Moonshot through existing GSSAPI support. The client must be installed on all parties in a Moonshot deployment: on the client device (a laptop, for example), the service, the RP proxy, the identity provider and the trust router (where applicable), so that all parties can understand the protocol. The API to access the client is the standard GSSAPI.

The service may be any service that supports multi-trip GSSAPI (many modern applications do, but some need help). The service generally does not need modification unless it is to add GSSAPI support where it previously did not exist. A prime example of this may be NFSv4, which may not support GSSAPI unless support has been built in during package creation.

The RP proxy is the gateway service between a GSSAPI-based service and the wider Moonshot network that uses the RadSec protocol for authentication purposes through the FreeRADIUS v3 package. The RP proxy also interacts with the trust router service to identify itself before querying the latter for realm information. Performance will depend almost entirely on the SQLite and FreeRADIUS software, which is designed to be very responsive in high-throughput environments.

The identity provider is a virtually standard FreeRADIUS v3 installation, with a change that allows it to interact with the trust router service and RP proxies that connect to it. As such, the identity provider will support all identity stores that FreeRADIUS will support (e.g. LDAP directories, relational databases, flat files), as well as SASL authentication based on username and password provided by the user. Performance here will depend on the speed and indexing of the identity stores as well as the hardware provided to the identity provider. As with the RP proxy, FreeRADIUS is designed to be responsive in a high-throughput environment. Any additional queries, such as to an attribute authority, will also have an impact on overall performance.

The trust router service provides the support for the trust between entities; it maintains the list of identity realms and their assigned hosts, the list of service realms and the constraints that bind them, as well as the communities of interest, which may be used in a similar fashion to virtual organisations (VOs). The trust router

software has been designed to be available for use in a clustered/multi-instance environment. It is recommended that a proxy is placed in front of the service to handle request management better across multiple instances.

The main AARC requirements supported are:

- Attribute aggregation / Account linking: Attribute aggregation is supported in the sense that both RADIUS attributes and a SAML assertion can be aggregated.
- Community-based authorisation
- Federation solutions based on open and standards-based technologies
- Browser & non-browser based federated access

3.2.3 Supported standards

GSSAPI
SAML2
RadSec
EAP

3.2.3.1 User interfaces and APIs

- On supported platforms with no built-in credential management (such as Linux), a credential manager is provided.
- Any application with MIT Kerberos compliant GSSAPI implementation can use Moonshot

3.2.4 Support for Virtual Organization (if relevant)

- Support for communities of interest on a RP Proxy level
- Support for attribute authorities that can provide further VO support exists to a degree (but has not been tested in anger)
- Account linking is encouraged on the RP Proxy/organisational level. Current use cases indicate that this is generally a preference, but that this may change in the future.

3.2.5 Dependencies with other technologies (libraries, DBs, etc.)

- OpenSAML libraries for internal SAML support
- Shibboleth2 Service Provider on the service (optional)
- FreeRADIUS v3, built with dynamic realm and trust router support (available from the Moonshot repositories)
- SQLite v3 (as non-volatile storage of keys received by either RP Proxy or IdP)

3.2.6 Operational overview

- The client is currently supported on RHEL 6- and Debian-based Linuxes, and Windows 7 and higher. Mac OS X is currently not supported, but support is currently in development. OpenSUSE support is also being explored.
- The RP Proxy and Identity Providers are available for Linux only. Mac OS X support may be considered once the client functions on the platform. Windows will not be supported due to lack of support for the platform by FreeRADIUS. The RP Proxy and Identity Providers can be virtualised. Docker containers have not been tried.
- The trust router is currently supported on Linux only and can also be virtualised. Docker containers have not been tried.

3.2.7 Expected level of support

Basic support is provided by the user community at large and by Jisc (to its connected customers). Documentation for the project is updated where and when necessary on the project wiki. The wiki is editable by members of the community as well.

3.3 SAFESHARE

3.3.1 Overview

SAFESHARE or SAFE SHARE is a Jisc-funded project (www.jisc.ac.uk) running November 2014-June 2016 to develop secure interconnect "solutions" for bioinformatics and biomedicine, particularly for researchers working with sensitive data (not to be confused with SAFE, discussed in 3.20). The project includes an AAI part with a focus on getting a sufficiently high *level of assurance* (LoA) using *two factor authentication* while retaining the *usability* required to get bioinformaticians to use it correctly. Additionally, the project includes facilities for securely interconnecting the participant institutions, the so-called Higher Assurance Network (HAN). While the work is mainly a pilot, we include it in this deliverable because it addresses the need for higher LoA for sensitive work, and for its support for the "last mile" where a trusted network needs to be connected through to the research institute and finally to the individual researcher.

Its principal stakeholders are the Farr Institute (www.farrinstitute.org), the Medical Research Council (www.mrc.ac.uk) and the Francis Crick institute (www.crick.ac.uk). As one of its core use cases, SAFESHARE will link the eMedLab (www.emedlab.ac.uk) sites, namely University College London, the Francis Crick Institute, Queen Mary, King's College London, EMBL-EBI, the Wellcome Trust's Sanger Institute, and the London School of Hygiene and Tropical Medicine. A more recent collaboration is with the Administrative Research Data Network (adrn.ac.uk), which facilitates the use of sensitive data in research.

3.3.2 Features supported by the tool

- Two factor authentication
- Secure private networking linking a large number of sites
- AAI for handling sensitive medical data

- Support for deployment offering end-to-end security

The AARC requirements supported by the tool are:
- User and Service Provider friendliness
- Different Levels of Assurance
- Step-up authentication
- Browser & non-browser based federated access
- Federation solutions based on open and standards-based technologies

3.3.3 Support for Virtual Organization (if relevant)

Not applicable.

3.3.4 Operational overview (High Availability, deployment scenarios)

SAFESHARE is being deployed along with Assent (the Moonshot federation in the UK) as one of its authentication factors. It is also possible to use Google Authenticator along with Yubikey as two-factor authentication.

3.3.5 Expected level of support

While SAFESHARE is developed by a project, it is expected that a sustainability model will be developed, ensuring that on-going support can be provided, probably by Jisc.

3.4 Shibboleth

3.4.1 Overview

Shibboleth is a standards based, open source software package for web single sign-on across or within organizational boundaries. Shibboleth began as an Internet2 project and is currently developed as open source software, released under the Apache Software License[8]. The Shibboleth software package consists of many components, some of which are described in other sections of this document. This section focuses on the components that implement the SAML Identity and Service provider, namely:
- Shibboleth Identity Provider
- Shibboleth Service Provider

[8] http://www.apache.org/licenses/LICENSE-2.0

3.4.2 Features supported by the tool

The Shibboleth Identity Provider provides Single Sign-On services through authentication of users and securely providing appropriate data to requesting services. In addition to a simple yes/no response to an authentication request, the Shibboleth Identity Provider provides a rich set of user-related data to the Service Provider. The main features of the Shibboleth Identity Provider can be summarized as following:

- Out-of-the-box support for LDAP, Kerberos, web server- and Servlet Container-based authentication systems.
- Out-of-the-box support for reading user data from LDAP directories and relational databases (no special schemas required) and performing simple or complex transformations on the acquired data.
- Support for releasing only selected data and making sure it gets there securely.
- Excellent scaling - a single instance can handle millions of authentication requests per day and can communicate with thousands of service providers.
- Works with all other known SAML implementations.
- Documented APIs to allow the software to be extended to support custom services.

The Shibboleth Service Provider SSO-enables and Federation-enables web applications written with any programming language or framework, integrating natively with popular web servers such as Apache and IIS.

The key features of the Shibboleth Service Provider are summarized below:

- Support for Apache and IIS web servers and FastCGI authorizers on a wide range of platforms, including Windows, Linux, OS X, and Solaris.
- Excellent scalability in both user load and management of Identity Providers.
- Support for virtualization of web servers and applications.
- Works with all compliant SAML implementations.
- A variety of authorization and policy-oriented features.

The AARC requirements supported by the tool are:

- User and Service Provider friendliness
- Attribute aggregation / Account linking
- User groups and roles
- Step-up authentication
- Browser & non-browser based federated access
- Federation solutions based on open and standards-based technologies

3.4.3 Supported standards

SAML 2.0
X509
Kerberos
LDAP
SQL

3.4.4 User interfaces and APIs

Web
SAML endpoints

3.4.5 Support for Virtual Organization (if relevant)

Group memberships can be retrieved by issuing SAML 2.0 AttributeQueries to an Attribute Authority configured to retrieve additional attributes from its database and releasing them inside the User Session together with other user attributes.

3.4.6 Dependencies with other technologies (libraries, DBs, etc.)

JAVA JRE
OpenSAML-JAVA
Java Application Server/Container (Tomcat, Jetty)
Apache mod_shib
LDAP

3.4.7 Operational overview (High Availability, deployment scenarios)

The Shibboleth Identity Provider (IdP) is a Java application that runs on a Java web application server (e.g. Apache Tomcat, Jetty).

The Shibboleth Service Provider consists of a daemon running on all major operating systems and a web server module, mod_shib, which is natively supported in the Apache HTTP server and IIS.

3.4.8 Expected level of support

Shibboleth is funded by the Shibboleth consortium. There is no indication that this situation will change in the mid-term future. Support is provided by the user community via the mailing lists, and the project is very well documented in the project's wiki page.

3.5 mod_auth_mellon

3.5.1 Overview

mod_auth_mellon is an authentication and authorisation module for the Apache HTTP server. More specifically, it is used to authenticate the user against a SAML2 IdP, and grant access to directories served through the Apache HTTP server based on attributes received from the IdP.

- Ownership: UNINETT
- Licence: GPL 2

3.5.2 Features supported by the tool

mod_auth_mellon turns an Apache HTTP web server into a SAML2 service provider. The required SAML2 SP metadata can be either configured statically by the system administrator or generated automatically by mod_auth_mellon.

The AARC requirements supported by the tools are:
- Browser & non-browser based federated access: Web-based resources
- Federation solutions based on open and standards-based technologies: SAML2-compliant

3.5.3 Supported standards

SAML2
SAML2 ECP

3.5.4 User interfaces and APIs

mod_auth_mellon configuration options can be set in Apache HTTP server's global and virtual host configuration files.

3.5.5 Support for Virtual Organization (if relevant)

Not relevant

3.5.6 Dependencies with other technologies (libraries, DBs, etc.)

- Apache HTTP server
- OpenSSL
- lasso

3.5.7 Operational overview (High Availability, deployment scenarios)

Being an Apache HTTP server module, mod_auth_mellon is deployed on the service provider's web server.

3.5.8 Expected level of support

Not specified. Project supported by UNINETT and the open-source user community.

3.6 pyFF (Discovery Service)

3.6.1 Overview

pyFF is a SAML metadata aggregator written in Python. It is based on the model for metadata exchange Metadata Interchange V3[9]. It is released with a BSD license.

3.6.2 Features supported by the tool

PyFF is a Python SAML metadata aggregator. It features pluggable "pipelines" for processing SAML metadata, signature validation and creation, support for using PKCS#11 tokens for signing, certificate expiration checking and reporting and fast parallel fetching of multiple streams.

3.6.3 Supported standards

SAML metadata
PKCS#11 tokens

3.6.4 User interfaces and APIs

The tool implements a user UI for the selection of the IdP.

[9] http://iay.org.uk/blog/2008/10/metadata_interc.html

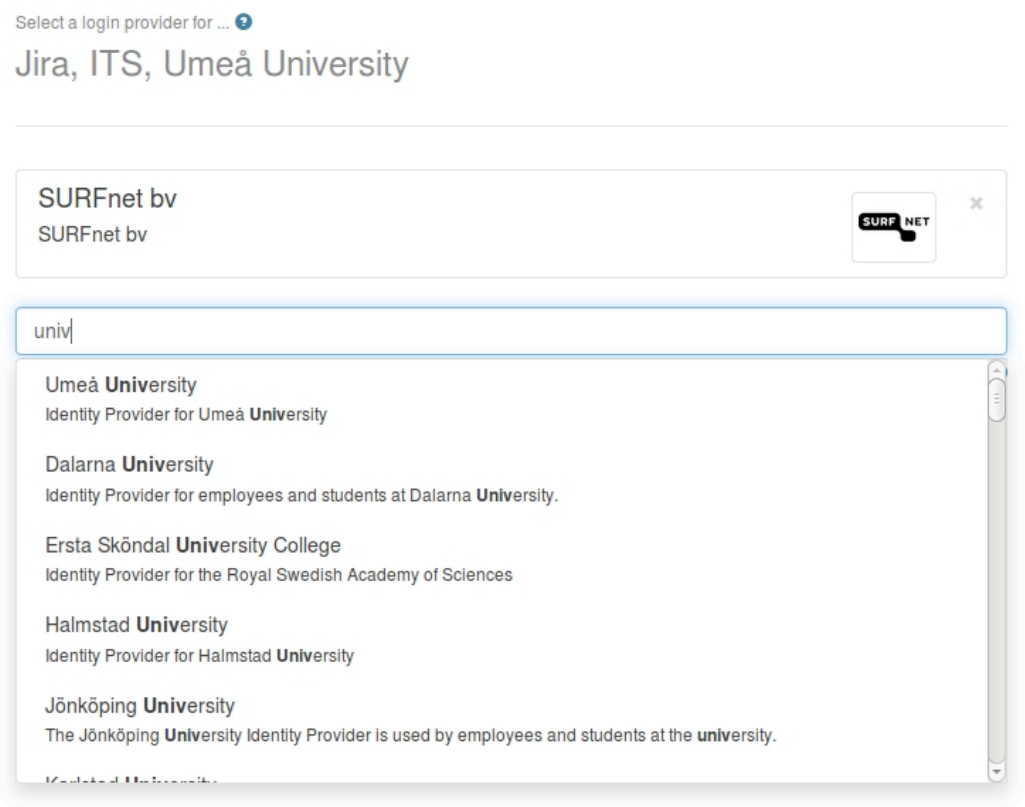

Figure 1: pyFF's IdP selection UI.

In addition, it can represent the results on the metadata handling as a Where Are You From (WAYF) screen. pyFF is used in production in various services and federations including SWAMID, SURFconext and others.

3.6.5 Support for Virtual Organization (if relevant)

Not relevant

3.6.6 Dependencies with other technologies (libraries, DBs, etc.)

Mostly Python libraries such as pyXMLSecurity.

3.6.7 Operational overview (High Availability, deployment scenarios)

pyFF is deployed by the service provider; it can be deployed in Apache or with a stand-alone web-service.

3.6.8 Expected level of support

PyFF is open source under a BSD license: https://github.com/leifj/pyFF[10]

3.7 Shibboleth Centralized Discovery Service (DS)

3.7.1 Overview

The Shibboleth Centralized Discovery Service[11] is part of Shibboleth open source project of the Shibboleth Consortium. It is a standalone service, primarily intended for use by identity federations and other large groups wishing to provide a backstop discovery service.

3.7.2 Features supported by the tool

The Shibboleth DS can work with multiple IdPs and SPs.

It automatically handles both the legacy Shibboleth AuthnRequest message (so-called "WAYF mode") and the full Discovery Service Protocol. The service contains the metadata sources of the IdPs that users will select from. In case of using SAML2 or other protocols not supported by the old WAYF model, metadata of SPs must also be configured in the service to enable safe interaction.

It can keep track of the IdP that was selected by using the browser cookie specified in the Identity Provider Discovery Profile and providing a hint to the user about his previous choice. Only the fact that an IdP was selected, not whether the user authenticated successfully, is stored.

3.7.3 Supported standards

- Shibboleth AuthnRequest (WAYF)[12]
- SAML2 Discovery Service Protocol[13]

3.7.3.1 User interfaces and APIs

- User interface: web
- SAML2

[10] https://github.com/leifj/pyFF
[11] https://wiki.shibboleth.net/confluence/display/SHIB2/DiscoveryService
https://wiki.shibboleth.net/confluence/display/SHIB2/DSInstall
[12] https://wiki.shibboleth.net/confluence/display/SHIB/AuthnRequest
[13] http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf

3.7.4 Support for Virtual Organization (if relevant)

Not applicable

3.7.5 Dependencies with other technologies (libraries, DBs, etc.)

- Java servlet container (e.g. Tomcat).
- Xerces
- Xalan
- Java

3.7.6 Operational overview

Shibboleth DS is a standalone service deployed in a Java servlet container (e.g. Tomcat). Prior to deploying it on the server, endorsement of Xerces and Xalan is required. The Discovery Service is configured by pointing to the metadata sources.

Shibboleth DS uses SAML2 metadata to find out about the Entities (SPs and IdPs) that it interacts with. For a per-federation discovery service this is usually the same as metadata that the federation publishes. For an SP-specific discovery service, this input metadata would be constrained to only the relevant SP and IdPs. Once deployed, the discovery service monitors the metadata files and will reload them as soon as they change.

3.7.7 Expected level of support

Shibboleth is an open source project founded by the Shibboleth Consortium. The lifetime of the consortium (and support for the software) is not limited by any specific project.

3.8 Shibboleth Embedded Discovery Service (EDS)

3.8.1 Overview

The Shibboleth Embedded Discovery Service[14] is part of the Shibboleth open source project of the Shibboleth Consortium. It allows a Service Provider to run a discovery service embedded within their own site.

[14] https://wiki.shibboleth.net/confluence/display/EDS10/Embedded+Discovery+Service

3.8.2 Features supported by the tool

When embedded in the SP site, Shibboleth EDS can look like any other page on the site and will thus not be as jarring to a user as being redirected to a totally different third-party discovery service site. Contrary to the central solution, the discovery may be restricted to only show those IdPs with which the SP has a relationship and to preferentially present favoured IdPs.

The EDS is a set of JavaScript and CSS files; installing it and using it is straightforward and does not require any additional software. It requires an installed and configured Shibboleth Service Provider. The IdPs are configured in a locally-stored JSON file.

3.8.3 Supported standards

- Shibboleth AuthnRequest (WAYF)[15]
- SAML2 Discovery Service Protocol[16]

3.8.4 User interfaces and APIs

- User interface: web
- Website: JavaScript

Figure 2: Example deployment of Shibboleth EDS.

[15] https://wiki.shibboleth.net/confluence/display/SHIB/AuthnRequest
[16] http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf

3.8.5 Support for Virtual Organization

Not applicable.

3.8.6 Dependencies with other technologies (libraries, DBs, etc.)

- Webserver
- To be embedded in PHP/HTML webpage
- Written in JavaScript and CSS

3.8.7 Operational overview

Shibboleth EDS allows the user to select the IdP during the authentication process. It is embedded in the user's webpage (small piece of HTML to be added in the webpage). The available IdPs are configured by setting up metadata providers in a local config file.

3.8.8 Expected level of support

Shibboleth is an open source project founded by the Shibboleth Consortium. The lifetime of the consortium (and support for the software) is not limited by any specific project.

3.9 DiscoJuice

3.9.1 Overview

DiscoJuice[17] is a flexible User Interface library for implementing an IdP Discovery Service by UNINETT AS, available on GNU Lesser General Public License version 3.0.

3.9.2 Features supported by the tool

- Works with SAML2 metadata
- Local Memory (cookie) and Remote Memory (DiscoReadWrite protocol + IdP Discovery)
- DiscoJuice JSON compact UI-focused Metadata format (MDUI friendly)
- Presents logos, searchable keywords, name, descr, country...

[17] http://discojuice.org

- Automatic discovery of country
- Inline incremental search
- Protocol agnostics, demoed with alternative protocols
- Multi-lingual both provider list (from metadata) and UI is translated into 15 languages

3.9.3 Supported standards

- SAML2 metadata

3.9.4 User interfaces and APIs

- Integration JavaScript API using call-backs
- HTML5 Geo-location API

3.9.5 Support for Virtual Organization

Not applicable

3.9.6 Dependencies with other technologies (libraries, DBs, etc.)

- Webserver
- Written in PHP, JavaScript, CSS

3.9.7 Operational overview

There are several ways to deploy DiscoJuice:

DiscoJuice embedded IdP selector popup in application. The login button can open the embedded DiscoJuice discovery service popup window that allows the user to select the Identity Provider, and then redirects the user to the IdP. If the page needs immediate authentication, it may redirect to an IdP discovery response page with embedded DiscoJuice discovery service.

Service Provider IdP Discovery Service. This option means setting up a preconfigured discovery service (webpage) for local service providers that implements the IdP Discovery Service Protocol. The SPs will redirect authentication to that webpage.

Identity Federation Central IdP Discovery Service. This option means setting up a preconfigured discovery service (webpage) for local federation that implements the IdP Discovery Service Protocol. The SPs will redirect authentication to that webpage.

Global IdP Discovery Service. This means using an existing service run by DiscoJuice.

It can be simply deployed at Service Provider side as a small JavaScript reference in the HTML source of the application.

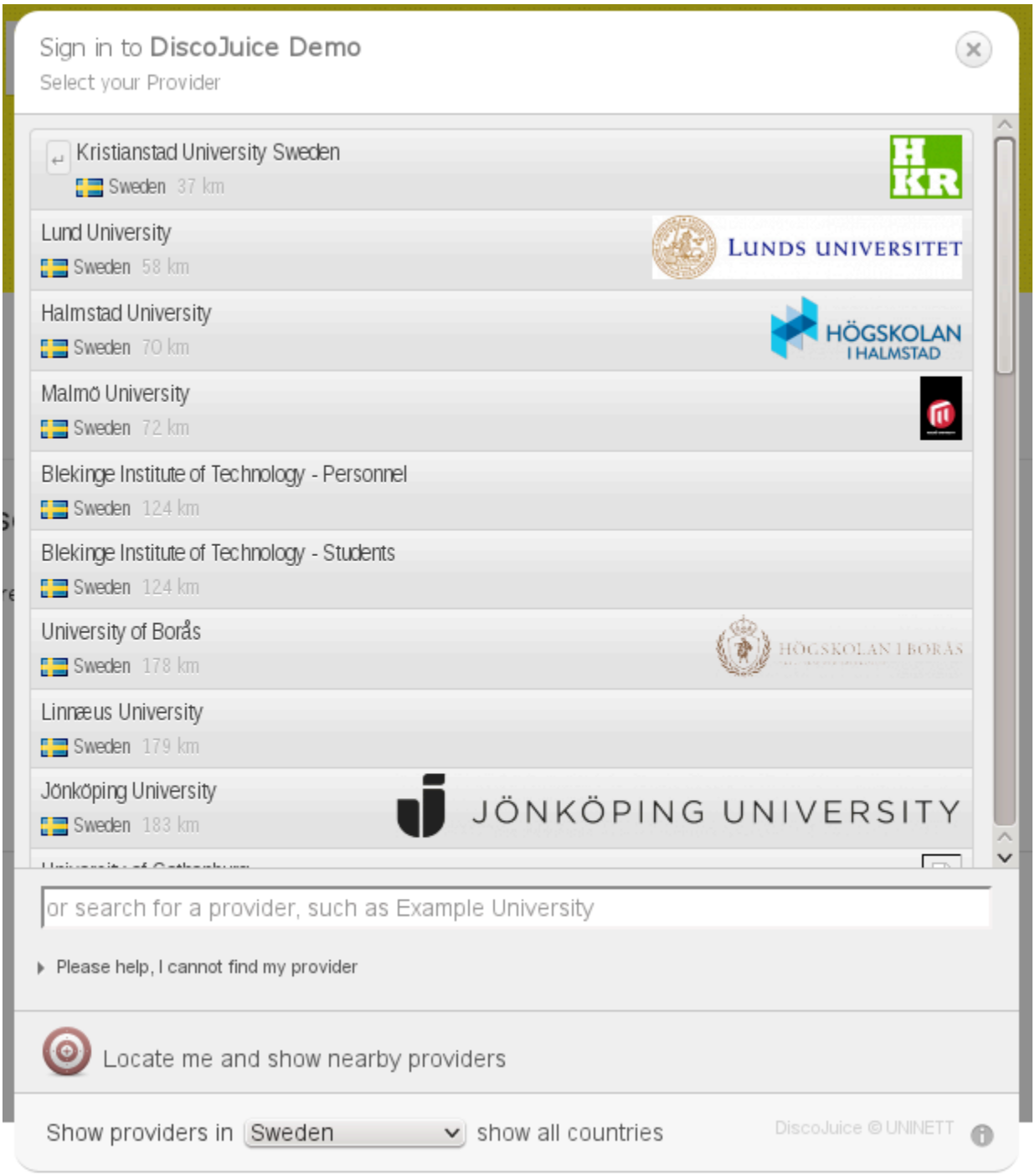

Figure 3: Example deployment of DiscoJuice embedded IdP selector popup.

3.9.8 Expected level of support

As of April 2014 DiscoJuice is not *yet* established in a production ready environment. Latest GitHub check-in was in 2013.

3.10 DiscoPower

3.10.1 Overview

DiscoPower is an open-source IdP Discovery Service implementation that is part of the official simpleSAMLphp release. Although not enabled by default, it extends simpleSAMLphp's built-in Discovery Service in order to scale to a large number of IdPs.

- Ownership: **UNINETT**
- Licence: **LGPL 2.1**
- Source code: https://github.com/simplesamlphp/simplesamlphp/tree/master/modules/discopower

3.10.2 Features supported by the tool

Compared to simpleSAMLphp's built-in Discovery Service, DiscoPower provides more advanced features, including tabbed organisation and live search capabilities. It allows users to filter IdP search results as they type (incremental) and is multilingual based on SAML2 metadata information.

3.10.3 Supported standards

Identity Provider Discovery Protocol[18]
Identity Provider Discovery Profile (SAMLProf)[19]

3.10.4 User interfaces and APIs

After enabling the DiscoPower module of a simpleSAMLphp installation, all configuration options are available through:
`BASE_SIMPLESAMLPHP_DIR/config/module_discopower.php`
Once configured, the DiscoPower discovery service instance can be accessed through the following endpoint:
`http(s)://FQDN/simplesamlphp/module.php/discopower/disco.php`
An example view of the DiscoPower Web UI as currently being used by the TERENA WAYF service has been captured in Figure X.

[18] http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf
[19] http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

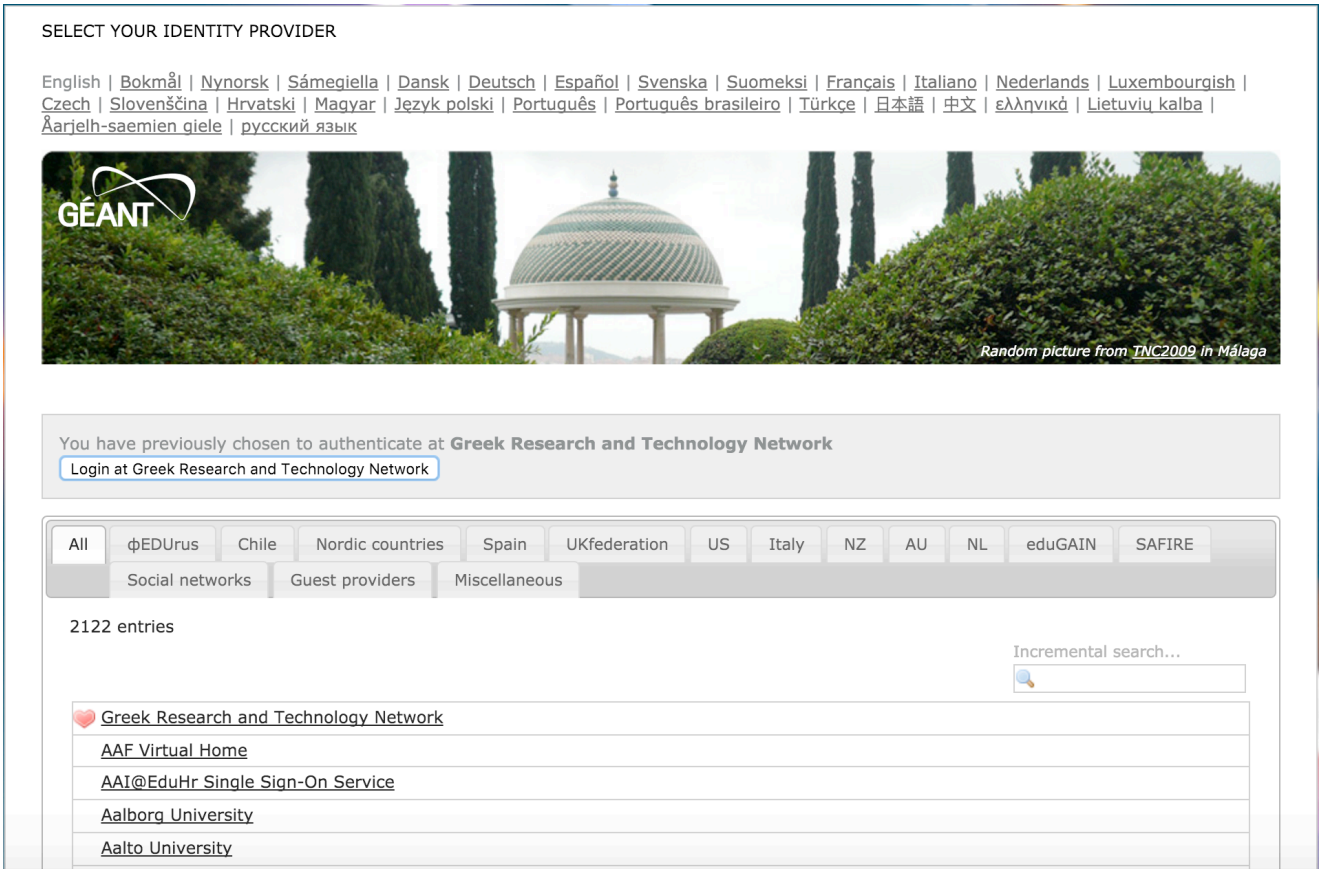

Figure 4: The Web UI of DiscoPower as currently being used by the GEANT WAYF service.

3.10.5 Support for Virtual Organization

Not applicable.

3.10.6 Dependencies with other technologies (libraries, DBs, etc.)

- Core module of simpleSAMLphp (not enabled by default)
- Written in PHP
- Reads SAML IdP metadata expressed in XML
- Accessed through a web server (e.g. Apache)

3.10.7 Operational overview

A DiscoPower-based IdP Discovery Service can be deployed by an Identity Federation as a centralised service available to all SPs in the federation. Alternatively, DiscoPower can be deployed on the server hosting the web-based federated resource.

3.10.8 Expected level of support

Although DiscoPower is not thoroughly documented in simpleSAMLphp's website, support can be provided through the user mailing lists[20].

3.11 SWITCHwayf[21]

The SWITCHwayf is an implementation of the Shibboleth WAYF and SAML2 Discovery Service protocol for use within a Shibboleth architecture.

3.11.1 Overview

The goal of the "Where Are You From" (WAYF) service is to send a user to the Identity Provider of his Home Organization. The WAYF also is referred to as "Discovery Service", which is also the name of a SAML specification implementing the Discovery Service protocol. In the following WAYF and DS are used synonymously although the DS protocol is slightly different as is shown below.

Basically, all the WAYF/DS has to accomplish is to present the user a list of Home Organizations and redirect the user's web browser to the selected Identity Provider (WAYF) or back to the Service Provider (Discovery Service) as this is shown below.

- Role: Discovery Service
- Ownership: SWITCHaai
- Licence: BSD license and provided "as-is".

3.11.2 Features supported by the tool

SWITCHaai reported these features on its main site:

- Lightweight PHP implementation
- Open-Source software
- Multiple languages support
- Category support in drop down list
- Reads SAML2 Metadata
- Automatic redirection to selected Identity Provider in current web browser session
- SAML Domain Cookie compliant
- Various ways of pre selecting an Identity Provider: Kerberos, IP range or IP reverse DNS lookup
- Centralized WAYF and Embedded WAYF feature
- Service Provider can enforce redirect to a Identity Provider
- Logging: General Log (Warning and Error messages) and Audit Log (IP and IdP selected)

[20] https://simplesamlphp.org/lists
[21] https://www.switch.ch/aai/support/tools/wayf/

3.11.3 Supported standards/protocol

- Identity Provider Discovery Service Protocol[22]
- Shibboleth AuthnRequest (WAYF)[23]

3.11.4 User interfaces and APIs

Figure 5: SWITCHwayf UI.

APIs:
Base URL: https://full.qualified.domain.name/WAYF

- **[/{I18N-STRING}][/redirect][/{ENTITYID-HOSTNAME}]**
  Hinted Identity Provider and transparent redirects

[22] http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf
[23] https://wiki.shibboleth.net/confluence/display/SHIB/AuthnRequest

- **[/{I18N-STRING}][/embedded-wayf.js]**
  Embedded WAYF JavaScript code

- **[/embedded-wayf.js/snippet.html]**
  snippet code to put into an unprotected page where host the Embedded WAYF

- **[/IDProviders.json]**
  Identity Providers list JSON format compliant

- **[/IDProviders.php]**
  Identity Providers list SWITCHwayf format compliant

- **[/IDProviders.txt]**
  Identity Providers list TXT format compliant

3.11.5 Support for Virtual Organization

Not applicable.

3.11.6 Dependencies with other technologies (libraries, DBs, etc.)

- PHP
- PHP XML Parser extension is required for parsing SAML2 metadata
- Apache

3.11.7 Operational overview

The service is usually deployed by an Identity Federation (IdP + SP) as Centralized Discovery Service and provided to the federated organisations' users.

The service can also be deployed by a SP as Embedded Discovery Service into the website hosting the federated resource and provided directly to the federated organisations' users.

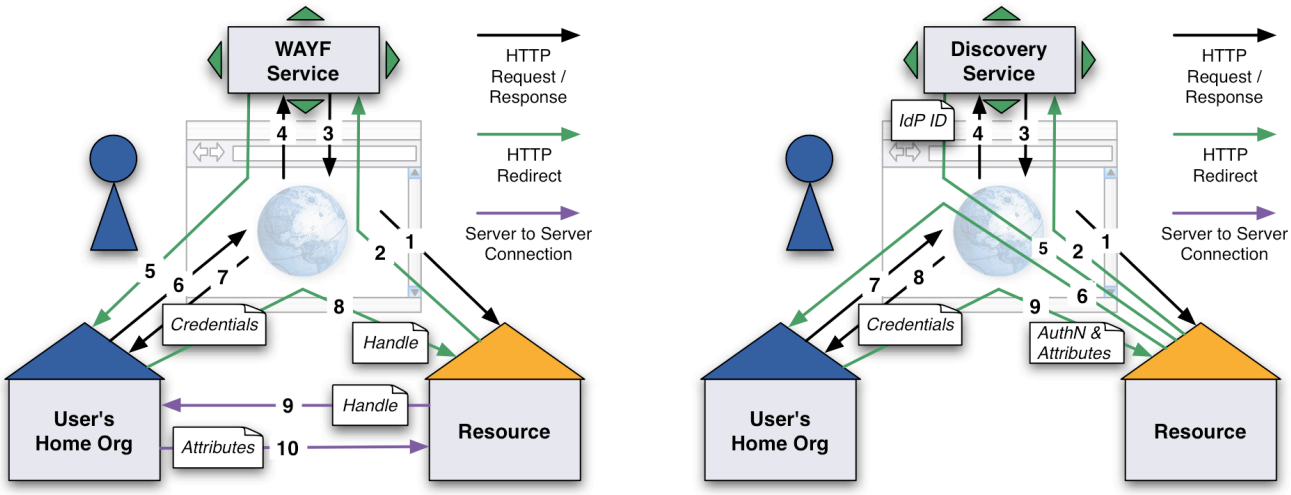

Figure 6: Possible WAYF flows.

3.11.8 Expected level of support

The tool is supported by SWITCH, and long-term support is expected.

3.12 HEXAA

3.12.1 Overview

Higher Education eXternal Attribute Authority (HEXAA) (https://sites.google.com/a/sztaki.hu/hexaa/) is an attribute authority and aggregation service developed and maintained by MTA SZTAKI and NIIFI. It has been developed by an eponymous Open Call project within the GN3plus project and is currently supported by the institutions that developed it.

3.12.2 Features supported by the tool

HEXAA is primarily an external attribute provider, meaning a third party providing attributes on top of the attributes provided by the IdPs.

- Tailored to VO Management
- Storage of non IdP dependent attributes
- Attribute release consent management
- Automations available triggered on changes

HEXAA can act also as an attribute aggregator, providing attributes from external (external from HEXAA) sources (e.g. ORCID).

HEXAA at the moment does not implement full VO membership life cycle management, such as attribute expiration.

3.12.3 Supported standards

HEXAA supports SAML2 protocol for attribute queries and user authentication.

3.12.4 User interfaces and APIs

HEXAA has a user interface focused on improving the VO Management, but it can be used to store every type of attribute.

Besides the standard SAML interfaces, HEXAA also exposes a REST API providing the following pieces of functionality:

1. Retrieving short-time tokens for principals;
2. Retrieving the attributes of a principal associated with a service.

The tokens can be used for accessing API functions. Every token is bound to the requesting user interface; therefore it is possible to restrict access to some API calls.

3.12.5 Support for Virtual Organisations

HEXAA has been designed to support VOs as main use case. HEXAA has no attribute management delegation capabilities, VO managers manage (i.e. configure and approve requests) the whole set of attributes of the VO.

3.12.6 Dependencies on other technologies

PHP, SimpleSAMLphp, MySQL, Apache

3.12.7 Operational overview

HEXAA can be either deployed (available on GitHub with installation instructions) in a local instance, or used as a service in the installation currently operated by SZTAKI.

3.12.8 Expected level of support

The GN3plus project that was originally supporting HEXAA ended in March 2015. Since then, the software has been supported by the SZTAKI and NIIFI institutes.

3.13 UNITY

UNITY is an open-source group, identity, and federation management solution. It is regarded as an authentication service for web or cloud services that enables outsourcing of user authentication to UNITY, using various authentication protocols. It also acts as a hub or proxy between the identity federations and the web or cloud services.

- Developed by ICM Uniwersytet Warszawski
- Developers: https://www.assembla.com/spaces/unity-public
- Users: http://unity-idm.eu/
- Licence: Open source Permissive BSD Licence

3.13.1 Features

- Management of groups and group hierarchies
- Provides internal authorisation to control access within the groups
- Registration and user form management to define forms for enrolment of new users along with email notifications, thus supporting “homeless” users
- Management of attribute consume and release policies (called translation profiles) on Web administrator interface and provides sandbox to “live” test the authentication and attributes release from the IdPs
- Attribute schema management to define new types of attributes
- Supports authentication of users from upstream SAML-, OIDC-, or LDAP-based identity providers as well as native username password and X.509 certificates
- Acts as an OAuth authorisation and resource server to issue access tokens and enables delegated access to user attributes
- Enables bridging of SAML identity federations
- Provides backup and restore functionality of whole server content
- Allows user interface customisation (or branding) for projects or organisations
- Different levels of assurance based on the type of an identity provider (e.g. institutional, social media provider) but no fine-grained attribute level support
- Attribute aggregation / Account linking
- Unique user identities
- User-managed identity information through the provided user registration and account forms.
- Up-to-date identity information (from UNITY v1.8.0)
- Non-web federated access but only when UNITY is not used as proxy IdP, that is only by the native users of UNITY.

3.13.2 Supported standards

- SAML2 (IdP and SP)
  - Web SSO Profile
  - SOAP Attribute Query
  - ECP for non-Browser based clients
- OAuth 2.0 and OIDC
- X.509

3.13.3 User interfaces and APIs

- Separate Web user interfaces for administrators and normal users respectively
- REST API to query user attributes
- Java API

3.13.4 Support for Virtual Organisations

- Hierarchical organization of groups (may be generic enough to be called VOs or virtual communities(?))
- Design and invocation of group specific registration forms

3.13.5 Dependencies on other technologies

- Java runtime environment
- Bundled with an embedded SQL database, but also supports MySQL and PostgreSQL

3.13.6 Operational overview

UNITY distribution can be downloaded and deployed as a standalone service. It can also be deployed in a manner to achieve high availability, however relies on backend SQL database replication functionality.

3.13.7 Expected level of support

- UNITY is being supported by long term PLGrid project and being deployed in the Human Brain Project (HBP), PLGrid, and EUDAT2020
- Developers mailing list: unity-idm-discuss@lists.sourceforge.net

3.14 Perun

Perun[24] is an identity and access management system that covers management of the whole user life cycle. Its key features are virtual organisation management, user and group management, resource management and service management. Perun has been designed to work in distributed and federated environments.

- Licence: FreeBSD licence
- Open source project available at https://github.com/CESNET/Perun
- Developed by CESNET and Masaryk University in Brno, Czech Republic

[24] http://perun.cesnet.cz

3.14.1 Features

- Complete VO and group management
- Identity consolidation (account linking)
- Push mechanism for authorisation data delivery (delivering ACLs, group information to services using push)
- Pull mechanism for authorisation data delivery via LDAP and AA
- Provisioning/de-provisioning of the user rights on services
- Enrolment management (customisable application forms, various enrolment flows)
- Delegation support for VO and group management
- Security teams support (global user banning)
- Import and synchronisation of users/groups with existing identity and group management systems
- Homeless users
- Different Levels of Assurance
- Flexible and scalable attribute release policies
- Persistent and unique user identifiers
- Browser & non-browser based federated access
- Social media identities
- Effective accounting
- Integration with e-Government infrastructures (Ready to be supported)

3.14.2 Supported standards

- VOOT
- SAML2 IdP and AA (via Shibboleth IdP)
- Various authentication protocols, primarily used in enrolment management (via Apache AuthN modules)
- LDAP

3.14.3 User interfaces and APIs

- Web-based GUI
- Command-line interface
- REST-like API
- Libraries: PHP, Perl, JavaScript and Java

3.14.4 Support for Virtual Organisations

- Supports multiple VOs
- Delegated administration of VOs and groups/subgroups
- Does not support hierarchical VOs, but supports VO to VO synchronization
- Support for VO registration (customizable VO application forms)
- Support for management of resources allocated to VOs

3.14.5 Dependencies on other technologies

- Supported DBs: PostgreSQL, Oracle DB
- Requires Java container
- Shibboleth SP and IdP
- OpenLDAP
- Apache

3.14.6 Operational overview

- It can be provided as a service by CESNET
- It is available as a virtual appliance
- Can be deployed locally using source code from GitHub (installation manual is not yet publically available)

3.14.7 Expected level of support

Perun has several production deployments (Czech e-Infrastructure provided by CESNET, Masaryk University, ELIXIR, EGI). Development and support team consists of employees from CESNET and Masaryk University, therefore code base maintenance and future development is ensured.

3.15 OpenConext

OpenConext[25] is an open source collaboration management platform. It provides a SAML2 proxy for identity provider and/or service provider federation, a group proxy for group management and built-in tools for the management of the service registry and of group providers. More specifically, OpenConext provides a set of infrastructure components that enable groups, teams or organisations to bring together federated tools such as wikis, mailing lists, or videoconferencing for use in a collaboration.

3.15.1 Features

Technically, OpenConext comprises two core components:

- 'Engine' is a SAML2 (SAML2Int WebSSO profile) compliant authentication proxy capable of acting as an IdP or SP. Apart from the authentication proxy, it also provides a "Where Are You From" (WAYF) service. Moreover, an interface allowing users to express their consent regarding the release of their identity attributes is available. Finally, the OpenConext Engine includes an interface enabling users to view and manage profile and group membership information.
- API: Serves as the group proxy, also providing a management tool, named Manage. It supports both the Grouper API and VOOT with either OAuth (2.0) or Basic Auth authentication. A central, end-user managed group service, Teams, is available by default.

All other components are provided by third parties, including SPs, IdPs and group providers.

[25] https://www.openconext.org

Support for step-up authentication (second factor) is available by adding the SURFnet-developed Step-up-as-a-service component[26].

OpenConext is actively being developed. Support for using OpenConext as an XACML PEP will be available by the end of 2015. Support for using OpenConext as an OpenID Connect OP will be available beginning of 2016. Formal support for Attribute Aggregation will be available in 2016 as well.

3.15.2 Supported standards

- SAML2 both for the interfaces with the IdPs and the SPs.
- OAuth 2.0 for the REST interfaces
- VOOT1 and 2
- XACML (PEP) (end 2015)
- OpenID Connect OP (mid 2016)

3.15.3 User interfaces and APIs

- Platform management:
  - SAML2 entity registration (SP, IdP AA), including ACL and ARP
  - Group management
- End-users:
  - WAYF
  - Group management
  - Consent management

3.15.4 Support for Virtual Organisations

- Hierarchical organisation, Virtual communities, delegated administration of the groups
- WAYF can be targeted at specific SPs
- Groups and attributes can be used to provide XACML PEP
- Attributes can be mapped

3.15.5 Dependencies on other technologies

OpenConext primarily reuses existing open source technologies, among others:

- Shibboleth
- SimpleSAMLphp
- Janus
- Spring security

[26] https://github.com/SURFnet/Stepup-Deploy

3.15.6 Operational overview

OpenConext is deployed in various scenarios[27] both as a Hub-and-Spoke federation hub as well as an SP Proxy.

3.15.7 Expected level of support

OpenConext is the core platform of SURFnet's SURFconext identity federation, which handles some 1 million authentications per week. OpenConext continues to be actively developed within SURFnet and via contributions to the Open Source software components it uses.

OpenConext has a website, an installable VM for testing and a mailing list for support.

3.16 VOMS

The Virtual Organization Membership Service (VOMS[28]) is an Attribute Authority that asserts attributes for users, both in the form of X.509 Attribute Certificates and SAML Attribute Assertions.

It is actively developed within the Italian Grid community and released[29] under the Apache 2.0 license.

VOMS is used in the Grid environment for authorisation purposes, serving as a central repository for Virtual Organization user authorisation information and providing support for organising users into group hierarchies, keeping track of their roles and other attributes.

The service follows an established client-server architecture and consists of:

- The VOMS core service (vomsd) that accesses a database (e.g. MySQL) shared with the administrative service (voms-admin);
- The VOMS-Admin tool, a Web application used to manage users and their privileges within a VO;
- Client tools and utilities (voms-proxy-init, voms-proxy-info, voms-proxy-destroy etc.) used to request a signed token (an Attribute Certificate compliant with RFC 3281) from a VOMS server, which carries the attributes that a person holds in a certain VO and is usually embedded inside an X.509 Proxy Certificate;
- APIs for attribute-based authorisation available in Java and C/C++ bindings, enabling easy integration of VOMS-based authorisation in existing services.

Figure 7 shows the Architectural Design with VOMS-Admin[30]:

[27] https://www.openconext.org/showcases
[28] http://italiangrid.github.io/voms/documentation/sysadmin-guide/3.0.4/service-ref-card.html
[29] https://github.com/italiangrid/voms
[30] http://www.eu-emi.eu/documents/10147/31168/VOMS.pdf

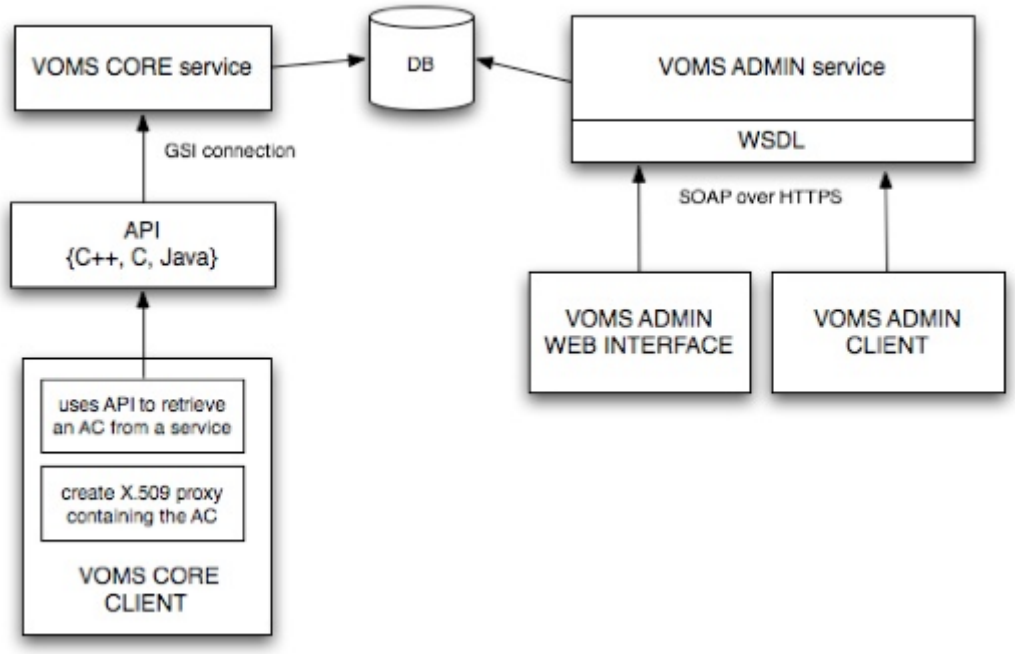

Figure 7: VOMS architectural overview.

3.16.1 Features

VOMS is a tool that allows communities to independently manage their structure and membership. As such, VOMS provides the administrators with the ability to organise users in groups, so called virtual organisations (VO) and designate specific roles and custom attributes to users. VOMS Admin provides a GUI for registration and VO management. Users can request membership via the same GUI. An API is also available, thus VOMS can be used both programmatically and interactively. VOMS outputs can be used in a delegation workflow. VOMS mainly speaks X.509, but the VO membership can be queried through a SAML attribute query as well, although this feature is not used in production.

3.16.2 Supported standards

X.509, proxies (RFC 3820)
VOMS Attribute Certificate (OGF/GFD.182)
SAML2 (OASIS)
SOAP 1.2 (W3C)

3.16.3 User interfaces and APIs

The VOMS-admin tool, a web application, to manage the users and their privileges in the VO. Easy to use and integrated C/C++ and Java libraries to interact with VOMS.

3.16.4 Support for Virtual Organisations

VOMS is the Virtual Organisation standard in the grid community (i.e. WLCG, EGI-based grids, but also available in Unicore-based grids).

3.16.5 Dependencies on other technologies

VOMS depends on an SQL database, per default this is MySQL, and on the Java virtual machine. It also heavily relies on OpenSSL.

3.16.6 Operational overview

A single VOMS service can support many VOs, depending on the number of users and the load on the service, VOs may prefer to deploy their own instances, but is more common among VOs that service providers particularly connected with the community operate a VOMS for many VOs. e-Infrastructures may have central catch-all instances to support communities without the need for the VO to deploy their own.

VOMS can be deployed in high availability configuration, i.e. two instances can be configured to host the same VOs and to replicate users' data, and be transparently used as alternatives.

3.16.7 Expected level of support

Due to the adoption of VOMS by many existing Grid infrastructures, support will continue, in terms of bug fixing and security support. However, support for new features is uncertain.

3.17 COmanage

COmanage[31] is a PHP-based open source (Apache 2 license) person registry designed to manage the identity lifecycle of Virtual Organisation (VO) participants. The project is hosted by Internet2, with primary funding to date via the US National Science Foundation.

3.17.1 Features

The VO participant life cycle typically begins with Enrolment Flows, which can be customised to meet the business processes of the VO. Typical enrolment patterns include invitation, self signup, and administrator-driven signup. As part of enrolment, attributes are collected from authoritative sources such as SAML assertions (e.g. ePPN, name, organisational affiliation) as well as from the Enrolee (user-asserted attributes such as preferred name or mobile phone). VO administrators can customise the attributes collected. Identifiers can be automatically assigned for new VO participants upon enrolment.

[31] http://www.internet2.edu/comanage

In order to facilitate management of larger VOs, COmanage supports delegated administration via a hierarchical model similar to LDAP OUs. Administrators can add roles and manage attributes for participants within their COU ("Collaborative Organizational Unit"). These roles can drive group memberships, which in turn can drive access to services. VO administrators may define Expiration Policies to automatically transition participants out of the VO based on various criteria, allowing for grace periods and other common termination patterns. Both human readable transaction history as well as database level point-in-time audit capabilities are provided.

Application or service integration primarily occurs via a plugin based provisioning infrastructure. Out of the box plugins include support for LDAP, a common application integration pattern, but custom plugins can be written as well. COmanage supports other types of plugins as well to facilitate various types of customisation. A REST API is also available.

COmanage supports various other identity management components. A typical deployment leverages the Shibboleth SP and EDS (though neither are required) for authentication services, and can be easily configured to work with Grouper to provide advanced group management capabilities. A proof-of-concept integration with OpenConext has also been successfully completed.

3.17.2 Supported standards

COmanage itself is standard agnostic. A typical deployment involves a SAML federation, but this is not a requirement and other authentication protocols can be leveraged as well or instead. LDAP is supported for provisioning. Experimental VOOT support has been implemented. Support for the evolving TIER APIs (formerly CIFER APIs) is planned.

3.17.3 User interfaces and APIs

COmanage ships with a fully internationalised, customisable, web-based user interface. A native REST API is also available. Custom functionality can be added by writing PHP-based plugins.

3.17.4 Support for Virtual Organisations

COmanage was designed around VO requirements, with enrolment and hierarchical/delegated administration capabilities to support typical VO models.

3.17.5 Dependencies on other technologies

COmanage is based on the CakePHP framework, and runs via a web server, typically Apache. An RDBMS is required, typically PostgreSQL or MySQL.

3.17.6 Operational overview

High Availability is achieved by making the relevant components (web server, etc.) highly available.

COmanage can be deployed in either single tenant or multi-tenant model.

3.17.7 Expected level of support

On-going maintenance will be funded by the Internet2 TIER initiative for the next several years. New feature development is currently funded from various sources, including grants and deployment specific funding.

The AARC requirements supported by the tool are:
- User and Service Provider friendliness

3.18 Grouper

Grouper[32] is a Java-based open source (Apache 2 license) group registry designed to provide sophisticated group management (and therefore authorisation) capabilities. The project is hosted by Internet2, and has been funded by Internet2, the US National Science Foundation, Jisc, and various universities.

3.18.1 Features

Grouper operates on a hierarchical tree or folder based design, where groups can be nested within other groups. Management of individual groups or entire folders can be delegated, and group memberships can be automatically calculated based on various criteria. Database level point-in-time audit capability is provided.

Groups can be sourced from authoritative sources ("loader" groups), calculated based on rules applied to other groups ("composite" groups), or managed manually ("ad hoc" groups). Group members are based on "subject sources", typically provided by an external person registry. Support for email invitation of external participants is also available. The resultant groups can be published to LDAP or queried via VOOT or web services APIs.

3.18.2 Supported standards

Provisioning and integration via LDAP, SCIM, SQL, VOOT.

3.18.3 User interfaces and APIs

As of v2.2, Grouper ships with a completely new web-based user interface. (The legacy "admin" and "lite" interfaces remain available.) Grouper Web Services offer access to group management capabilities via SOAP and REST-like interfaces.

[32] http://www.internet2.edu/grouper

3.18.4 Support for Virtual Organisations

Grouper offers VOs the ability to represent complex group relationships (hierarchical, set oriented, etc.), and the delegated administration of those memberships.

3.18.5 Dependencies on other technologies

Grouper requires Java and an RDBMS (PostgreSQL, MySQL, Oracle, etc.). In addition, the UI and WS require ant, a servlet container (e.g. Tomcat), and a web server (e.g. Apache).

3.18.6 Operational overview

High Availability is achieved by making the relevant components highly available.

3.18.7 Expected level of support

On-going maintenance will be funded by the Internet2 TIER initiative for the next several years.

3.19 ARGUS

Argus[33] is an authorisation framework developed in EGEE-III and the primary authorisation service used in the EGI infrastructure. It is based on XACML2, consisting of separate PAP, PDP and PEP components. The PEP is split into a separate PEP-server and PEP-client part. The PEP-server and client communicate with each other via a proprietary binary protocol (`Hessian'). The Policy Administration Point (PAP) provides the tools to author authorisation policies, organise them in the local repository and configure policy distribution among remote PAPs. The Policy Decision Point (PDP) implements the authorisation engine, and is responsible for the evaluation of the authorisation requests against the XACML policies it retrieves from the PAP. The Policy Enforcement Point Server (PEP Server) ensures the integrity and consistency of the authorisation requests received from the PEP clients. Lightweight PEP client libraries (Java and C) are also provided to ease the integration and interoperability with other EMI services or components.

- Ownership: maintained by INFN (Java based components) and Nikhef (C-based components)
- Licence: Apache-2.0 licence

3.19.1 Features

The PAP provides fine-grained and hierarchical authorisation decisions. It is currently used with X.509-based credential attributes (such as subject- and issuer-DN) as input, but is adaptable for use with other types of

[33] https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework

attributes. It can be used for community-based authorisation via VOMS attributes. Authorisation decisions based on a specific combination of VO, CA and authentication profile is on the roadmap, in the form of a PIP.
The PEP-server provides a plugin type of framework via PIPs and Obligation Handlers (OHs), such as an obligation handler for mapping to a local Unix account.

3.19.2 Supported standards

SAML2-XACML2 (PAP and PDP only); X.509; VOMS

3.19.3 User interfaces and APIs

Libraries: Java and C libraries exist for communicating with the PEPd (using the Hessian binary web service protocol).
Command line: pap-admin, pepcli
External plugins: LCMAPS plugin (PEP client), gsi-callout library (for use in e.g. gsissh or GridFTP).

3.19.4 Support for Virtual Organisations

Hierarchical organisation, Virtual communities, delegated administration of the groups

3.19.5 Dependencies on other technologies

voms-java, canl-java, numerous Java libraries such as opensaml.

3.19.6 Operational overview

Sites typically install one Argus service consisting of a PAP, PDP and PEPd on one host. The PAPs can be deployed in a hierarchical way: site, NGI (National Grid Initiative) and central (e.g. Europe). The site's Argus PAP instance can import policies from an NGI Argus instance, which in turn can import policies from a central Argus-PAP. Together with CERN, EGI runs such a European central PAP that is used for centrally suspending users.

3.19.7 Expected level of support

Argus Java-based components (e.g. PAP, PDP and PEP server) are currently maintained by INFN; Argus C-based (PEP C-based client library, gsi-callout plugin, pepcli) components are currently maintained by Nikhef.

3.20 SAFE[34]

SAFE is a toolkit for building user account management, quota management, and/or CPU accounting for High Performance Computing (HPC) infrastructures. It was chosen in this deliverable because it fulfils a practical need of user account and quota management in production HPC infrastructures yet is a sufficiently flexible toolkit to enable authorisation decisions. Developed by EPCC at the University of Edinburgh (www.epcc.ed.ac.uk), it is used across HPC nodes such as ARCHER (www.archer.ac.uk), DiRAC (www.dirac.ac.uk), and the Hartree supercomputing centre, as well as managing accounting data for PRACE.

The accounting part of SAFE is open source and freely available on SourceForge[35].

3.20.1 Features

The main features are user account management, CPU usage accounting and quotas, and authorisation. A deployment of SAFE need not use all features.

If the service supports users from multiple communities including local users, commercial users, academics from a wide variety of backgrounds etc. It is difficult to exclusively rely on external frameworks for AAAI services. However, for the sub-set of users who do have access to such frameworks it is important to integrate with the frameworks so those users do not lose the advantages of single-sign-on etc.

3.20.2 Supported standards

SAFE has been integrated with many of the standard authentication mechanisms available to web servers via external authentication, such as Shibboleth, Cosign, and X.509 (client) certificates. Additional mechanisms are usually not hard as long as there is an appropriate web server plug-in.

3.20.3 User interfaces and APIs

The SAFE tool gives HPC infrastructure administrators a means of managing accounts and quotas, and to get an overview over resources used.

The typical workflow is as follows:

- A user will register with a HPC site, providing also an SSH key with which they wish to authenticate themselves. They also typically apply for membership of a "project," a group with a common purpose
- An administrator will review the application and assign a quota to the project
- The user will access the resource(s) in question
- The resources will send accounting data back to the SAFE system
- The administrator will review the resource usage

[34] This entry written with contributions from Stephen Booth from EPCC, lead developer of SAFE <s.booth@ed.ac.uk>
[35] http://gridsafe.sourceforge.net/

As discussed before, any mechanism that can provide certificate proxies to web-portals (Sarongs or Globus-online OAuth) could easily become an external authentication mechanism for the SAFE that automatically captures the certificate identity. These external authentication mechanisms can be configured as either the primary or secondary methods. The primary method is required to register with the site. Secondary methods are alternative authentication methods that users can optionally bind to their accounts (after normal registration) allowing them to use external SSO systems. For example, the Archer and DIRAC SAFE systems allow UK Access Management Federation IDs to be used as a secondary login but internal password based authentication is the primary.

3.20.4 Support for Virtual Organisations

The “project” is the SAFE equivalent to a VO; it denotes a group of users to whom the administrator has allocated resources, and the users then share and consume resources as they see fit (however, the administrator can still see the individual user’s consumption of resources)

3.20.5 Dependencies on other technologies

Note that SAFE needs to be interfaced to the resource accounting of the batch system on the HPC cluster in question.

3.20.6 Expected level of support

While originally developed by one person, SAFE is now supported by a team of 4-5 people at EPCC.

3.21 SimpleSAMLphp

SimpleSAMLphp is an open-source implementation for federated AAI based on SAML.

- Ownership: **UNINETT**
- Documentation: https://simplesamlphp.org/docs/stable/
- Licence: **LGPL 2.1**
- Source code: https://github.com/simplesamlphp/simplesamlphp

3.21.1 Features

SimpleSAMLphp primarily focuses on providing support for SAML2 SPs and IdPs.

At the same time, SimpleSAMLphp supports other identity protocols and frameworks, such as Shibboleth 1.3, A-Select, CAS, OpenID, ADFS, WS-Federation or OAuth. It also supports popular social media identity providers, such as Facebook, LinkedIn, MySpace, Twitter and Windows Live. SimpleSAMLphp is easily extendable due to its modular architecture. Some of the most important extension points of SimpleSAMLphp include:

- Authentication Modules: For implementing custom authentication methods, such as PKI-based, or using proprietary user data sources.
- Authentication Processing Filters: To allow any kind of processing right after authentication has taken place.
- Themes: To customise the look of any page served by SimpleSAMLphp by modifying the CSS, headers, and footers.
- Modules: For extending SimpleSAMLphp with new identity protocols, pages, registry systems etc.

SimpleSAMLphp comes with a number of built-in modules, authentication modules and processing filters that may be used as is, or modified to fit specific needs. It also provides:

- an abstract datastore API, allowing alternative ways of storing data
- an abstraction layer of metadata handling, allowing alternative implementations of metadata consumption
- multiple session handlers, e.g. PHP built-in session handling or Memcache
- step-up authentication based on different Levels of Assurance (see example[36])

Apart from the modules that ship by default with SimpleSAMLphp, a number of extra modules have been made available by third-party developers covering specific features. A non-exhaustive list of such modules follows:

- Attribute Authority: Provides back-end SAML Attribute Authority functionality.
- Attribute Aggregator: Supports attribute aggregation as an Authentication Processing Filter.
- Content Simple Admin: Implements a very simple user interface for managing user consent.
- Kerberos: Enables Kerberos 5 authentication.
- Metadata aggregator2: Aggregates a set of SAML entities into SAML2 metadata documents.
- Metaedit: Allows basic editing of metadata, as well as manually registering metadata for service providers.
- OAuth 2.0: Adds support for the OAuth 2.0 protocol.
- OpenID Consumer: A module adding support for the OpenID protocol as a Consumer.
- OpenID Provider: A module adding support for the OpenID protocol as an Identity Provider.
- Selfregister: Allows registration of users accounts.
- VOOT Groups: Allows retrieving group memberships from an API service protected with OAuth 2.0 using the VOOT protocol and adds this information to the list of attributes received from the IdP.
- Attribute Aggregator module (developed by NIIF): Issues SAML2 AttributeQuery to an Attribute Authority that supports SAML2 SOAP binding
- Attribute-from-rest-api module (developed by NIIF): Requests attributes from REST API in JSON format

3.21.2 Supported standards

SAML2/1.1, OpenID, OAuth 2.0, Kerberos, VOOT, SQL, LDAP, RADIUS

3.21.3 User interfaces and APIs

- Web-based UIs
- SAML SP/IdP metadata in XML exposed through HTTP

[36] https://wiki.surfnet.nl/display/SUAAS/Configuring+a+simpleSAMLphp+SP+for+step-up+authentication

- PHP-formatted configuration files

3.21.4 Support for Virtual Organisations

Using the third-party VOOT Groups module, SimpleSAMLphp can retrieve group memberships from an API service protected with OAuth 2.0 using the VOOT protocol and add this information to the list of attributes received from the IdP.

3.21.5 Dependencies on other technologies

- Written in PHP
- Runs on a wide variety of web servers (Apache, nginx and IIS among others)
- Can utilise a user repository based on a SQL database (e.g. MySQL or PostgreSQL), an LDAP directory (OpenLDAP) or a RADIUS interface (OpenRADIUS).
- Can maintain session information in memcached servers for improving performance/high availability

3.21.6 Operational overview

As simpleSAMLphp is written in PHP, integrating Web-based PHP applications into a federation is very simple. However, simpleSAMLphp also supports non-PHP environments by adding a special cookie in Memcache suitable for the Apache “Auth Memcookie” module. This approach allows passing authentication information in HTTP header variables and enables authorisation via the Apache server configuration. Memcache can also be used to allow an arbitrary number of SimpleSAMLphp web front-ends to work with a back-end matrix of Memcache servers in support of both replication (fail-over) and load-balancing capabilities.

To connect the same SP to multiple IdPs, simpleSAMLphp offers two built-in SAML2 IdP Discovery Services: a basic (enabled by default) and a more advanced one providing scalable search capabilities (please refer to Section 3.10). To act as an IdP, simpleSAMLphp can be configured to utilise a user repository based on a SQL database (e.g. MySQL or PostgreSQL), an LDAP directory or a RADIUS interface.

3.21.7 Expected level of support

SimpleSAMLphp has a large user base, a helpful user community and a number of external contributors[37].

The AARC requirements supported by the tool are:

- Federation solutions based on open and standards-based technologies

[37] https://github.com/simplesamlphp/simplesamlphp/graphs/contributors

3.22 CILogon / OAuth for MyProxy

CILogon[38] provides a federated X.509 certification authority for secure access to cyber infrastructure. CILogon relies on federated authentication for determining user identities when issuing certificates. Federated authentication enables CILogon to serve a national-scale user community without requiring a large network of registration authorities performing manual user identification.

- Ownership: National Center for Supercomputing Applications, University of Illinois.
- Licence: MyProxy: http://grid.ncsa.illinois.edu/myproxy/license.html, NCSA and BSD and ASL 2.0

3.22.1 Features

The CILogon Service[39] allows users to authenticate with their home organization and obtain a grid certificate. The CILogon Service is implemented by a web application, with a back-end MyProxy CA[40] that uses InCommon (SAML) for authentication. Users authenticate to CILogon via the SAML protocol using their campus credentials. The InCommon federation publishes public keys for identity providers (i.e. campuses) and service providers (i.e. CILogon) so they can trust each other. CILogon takes the user information (name, email, unique ID) from the SAML assertion issued by the campus, asks the MyProxy CA to issue a certificate containing that information, and delivers the certificate to the user.

The service consists of a number of separate building blocks, one of which is either an OAuth1 or OpenID Connect server serving as a frontend for a MyProxy server. The CILogon service acts as either an OAuth1 or OpenID-Connect client. The MyProxy server is running typically in CA mode, where the CA is based on OpenSSL, which via suitable engines can also use an HSM as backend.

3.22.2 Supported standards

X.509, OAuth1, OpenID Connect, SAML, MyProxy

3.22.3 User interfaces and APIs

The CILogon service provides a few different interfaces for issuing certificates: web browser, command-line, and OAuth or OpenID Connect (the latter not yet used in production in the US). The web browser interface allows the user to download a PKCS12 file containing a user end-entity certificate with corresponding private key, which can be easily imported in the web browser. The command-line API is based on SAML-ECP[41]. The OpenID Connect API allows, using the OpenID/OAuth for MyProxy protocol, for 'portal delegation[42]' where a portal can obtain and use a user's end-entity certificate.

[38] http://dx.doi.org/10.1002/cpe.3265
[39] https://cilogon.org
[40] http://grid.ncsa.illinois.edu/myproxy/ca/
[41] http://www.cilogon.org/ecp
[42] http://www.cilogon.org/portal-delegation

MyProxy servers, which form the backend for CILogon, can run in multiple modes. They can act as online CAs, via OpenSSL, and can thus support full HSM-based CAs. They can also act as secure credential stores for storing full end-entity certificates or proxy certificates. MyProxy uses the myproxy[43] protocol for which there also exist command line tools.

3.22.4 Support for Virtual Organisations

The CILogon service is primarily focussed on the provisioning of end-entity certificates based on institutional logins. On the other hand, the MyProxy server in credential store mode can also directly interface with VOMS servers and return VOMSified proxy certificates. Integration with e.g. COmanage and support for campus LDAPs is on the roadmap for CILogon 2.0.

3.22.5 Dependencies on other technologies

OpenSSL, Globus toolkit (for MyProxy server), Apache Tomcat (for CILogon OpenID Connect server)

3.22.6 Operational overview

Portal delegation scenario: CILogon OpenID Connect 'delegation server' in front of a MyProxy online CA. Custom(izable) OpenID Connect client can retrieve a EEC on behalf of the user.

Web browser scenario: web portal in front of a MyProxy online CA. User can directly download a PKCS12 file.

Command line scenario: a command line tool (e.g. Perl script), using SAML-ECP, and run by the user, retrieves a certificate to be used for non-web access (e.g. grid compute and storage).

3.22.7 Expected level of support

CILogon is production software widely used in the US. It is actively maintained and further developed.

3.23 TCS

Since 2005, GÉANT Association's Amsterdam office (formerly TERENA) has coordinated a joint procurement on behalf of European NRENs to provide Transport Layer Security (TLS) certificates to their constituencies. Initially, this service focused on the issuance of server certificates, but in recent years it has been expanded to include personal certificates and code signing certificates.

[43] http://grid.ncsa.illinois.edu/myproxy/protocol/

The Trusted Certificate Service (TCS [44]) takes advantage of a bulk purchasing arrangement whereby participating national research and education networking organisations (NRENs) may issue close to unlimited numbers of certificates provided by a commercial CA at a significantly reduced price.

At the time of writing the selected Certificate Authority (CA) provider is DigiCert.

3.23.1 Features

The five main types of certificates available under TCS are:

- SSL certificates – for authenticating servers and establishing secure sessions with end clients.
- Grid certificates – for authenticating Grid hosts and services (IGTF compliant).
- Client certificates – for identifying individual users and securing email communications.
- Code signing certificates – for authenticating software distributed over the Internet.
- Document signing certificates – for authenticating documents from Adobe PDF, Microsoft Office, OpenOffice, and LibreOffice.

Certificates issued under TCS service support the SHA-2 algorithm as hash function.

The issuance of Personal and Personal Grid certificates is provided by a web portal that requires SAML-based federated access. It means that an NREN who joined TCS is required to run/operate/mange a Federation and has to join the eduGAIN confederation in order to provide SSO login to the personal certificate portal to its users.

3.23.2 Supported standards

Standards involved in TCS and relevant for our scope are:

- X.509
- SAML

3.23.3 User interfaces and APIs

For the TCS service started on the 1st of July, DigiCert developed a new Web portal.
Each user of the portal can create their own account credentials themselves after an initial invitation provided by one of the already established Administrators.

The web portal login process can easily be enriched to become more secure by adding the two-factor authentication feature. The second factor can be OTP or a personal certificate. Only Administrators can make these choices on the web portal.

DigiCert also provides a set of APIs to interact with the portal.

[44] http://www.geant.org/Services/Trust_identity_and_security/Pages/TCS.aspx

3.23.4 Support for Virtual Organisations

Hierarchical organisation, Virtual communities, delegated administration of the groups.

3.23.5 Dependencies on other technologies

For the issuance of personal certificates, the web portal requires the NREN to have joined the eduGAIN confederation.

3.23.6 Operational overview

Due to the fact that the web portal is operated by DigiCert, NRENs don't need to run any service/server to request certificates. For the issuance of personal certificate, an NREN must operate a federation included in eduGAIN.

3.23.7 Expected level of support

NRENs who joined TCS can count on a mailing list: tcs@lists.geant.org whose recipients list consists on the other NREN Admins and a support email from DigiCert: support@digicert.com

GÉANT has created a wiki to collect all information related to the service:
https://wiki.geant.org/display/TCSNT/Trusted+Certificate+Service+%28new+TCS%29+Home

Running period: July 2009 - July 2017.

3.24 STS

The Security Token Service (STS) is a service used for transforming an existing security token into another format. A security token is defined as a collection of claims that can be attached to a web service message. STS can receive both UsernameTokens (for verification against LDAP) as well as SAML assertions. From these, STS issues an X.509 certificate and proxy certificate containing the user's VO attributes. This service was developed by EMI for use by WLCG and other organisations utilising EGI.

STS is currently deployed as part of a WLCG pilot for federated access. It has been implemented with Java.

STS: https://twiki.cern.ch/twiki/bin/view/EMI/EMISTSDocumentation

STS Code: https://gitlab.cern.ch/sts/sts-server

EMI: https://twiki.cern.ch/twiki/bin/view/EMI/WebHome

3.24.1 Features

The tool's key purpose is to translate Username and SAML tokens into X.509 and proxy certificates for use within EGI.

3.24.2 Supported standards

- X.509
- SAML

3.24.3 User interfaces and APIs

The interfaces are explained at the following link: https://twiki.cern.ch/twiki/pub/EMI/EMISTSDocumentation/sts-service_interface_description-1.0.pdf

STS uses a Web Services Trust interoperability profile that attempts to strike a balance between the extensibility offered by the WS-Trust specification and the need to scope that functionality into a manageable set.

The end point URLs are listed at the following link: https://twiki.cern.ch/twiki/bin/view/EMI/STSOperation

This service contains the following endpoint URLs:

- https://sts.example.org:8443/sts/wstrust - This endpoint is the recipient of the WS-Trust request messages
- https://sts.example.org:8443/sts/ecp - This endpoint is the recipient of SAML2 ECP profile initiation requests

3.24.4 Support for Virtual Organisations

STS interacts directly with VOMS, which governs authorisation for proxy certificates.

3.24.5 Dependencies on other technologies

For the certificates generated by STS to be considered valid, the CA must be IGTF approved.

3.24.6 Operational overview

For full information see the following link: **http://www.eu-emi.eu/security-token**

STS is required to enable federated access via SAML to infrastructures such as WLCG where X.509 certificates are used. The ability to dynamically create certificates based on home organisation authentication and VOMS authorisation facilitates collaboration and remote access to service providers.

3.24.7 Expected level of support

Since the closure of the EMI project, no active support is provided. There is an admin guide at the following link: https://twiki.cern.ch/twiki/pub/EMI/EMISTSDocumentation/sts-sys_admin_guide-1.0.pdf

3.25 ADFS

While the title of this section is "ADFS", the Active Directory Federation Service, it discusses more generally building identity federations on/with web services using technologies such as WS-Federation[45]. Indeed the core concept of a Secure Token Service (STS) has been implemented by others; the previous section describes an implementation for EMI, and IBM support similar technologies for their WebSphere product[46].

ADFS itself provides a means of linking sites that internally use Active Directory (AD) together in a federation using SAML and WS-* standards, to enable users to use home identities outside their institute without exposing their credentials. It is of interest to this deliverable (a) because of the prevalence of Active Directory deployments, (b) for using open standards to federate access to shared resources, particularly for (SOAP-based) web services, and (c) because the technology can be used for IdMaaS, cf. SA1.1.2 by deploying the STS in the cloud.

The AD approach specifically is becoming increasingly important as research institutes and companies use cloud resources, and ADFS provides single sign-on access to the cloud[47]. It is not the only means of doing so as Microsoft now use "Azure AD Connect" for linking a site AD to Azure; however, in this deliverable we are interested not so much in linking a site to the cloud but in the more general technology for building federations.

3.25.1 Features

While the services can be used as a proxy in the sense of MJRA1.4 *Blueprint Architecture*, a full deployment can be used to federate between different *security realms*.

3.25.2 Supported standards

WS-Federation is an OASIS standard. Building on a slew of other WS-* standards, it provides a means of building web services (by default SOAP) in a federation.

[45] http://www.oasis-open.org/committees/wsfed
[46] http://www.ibm.com/developerworks/library/ws-SAMLWAS/ (accessed Dec '15)
[47] Notably to Azure, of course: https://azure.microsoft.com/en-us/documentation/services/active-directory/ (accessed Dec '15) but there is nothing preventing anybody else from setting it up and it need not be deployed by the CSP. Conversely, Azure also supports SSO via other technologies such as SAML.

3.25.3 User interfaces and APIs

Primarily, the technology was designed for SaaS, where a SOAP API for example could be exposed to users from other security domains. Web interfaces (with browsers as clients) are also supported via WS-PassiveFederation.

3.25.4 Expected level of support

The protocols described in this section are standard protocols.

Microsoft's currently preferred option for providing federated access to resources in Azure is "Azure AD" which in turn encompasses support for SAML2, OpenID Connect, as well as WS-Federation, but the scenario is evolving. IBM continues to support WebSphere but could also find other ways to integrate federated identity management. PingIdentity had a WS-Federation implementation[48]. The authors tested the Apache module for the SP some years ago, but found it somewhat buggy (however, source code was available).

3.26 LDAP Facade

The LDAP Facade is an open source project, licensed under the GNU Public License (GPL). Its purpose is to allow users with federated identities, by using SAML, to authenticate to typical Unix services via LDAP.
It is maintained by the Steinbuch Centre for Computing at Karlsruhe Institute of Technology (KIT).

3.26.1 Features

By combining SAML with LDAP you create a bridge from SAML directly to any service that supports authentication to LDAP. One example is the PAM LDAP plugin, which enables any service using PAM to directly support the LDAP Facade, such as SSH. This means one can login at a Linux machine that uses PAM-LDAP using federated identity.
The LDAP Facade has the ability to inspect the password field, as this is needed for authentication. But it is possible to do even more than pure authentication with user/password. The LDAP Facade uses the password field to perform smart processing of tokens, which are shipped as passwords, such as bearer-tokens or SAML assertions.
The LDAP Facade has implemented multiple production-tested smart strategies to overcome the general problem of deprovisioning.

3.26.2 Supported standards

The current release of the LDAP Facade was built with support for:

- SAML/Web SSO (SAML2)
- SAML/ECP (SAML2)
    - Enhanced proxy: Password traverses through LDAP-Facade

[48] https://technet.microsoft.com/en-us/library/adfs2-federation-with-ping-identity-ping-federate(v=ws.10).aspx

    - Enhanced client: Password does not traverse through LDAP-Facade
- LDAP

As the LDAP Facade is under active development, the support for the following tokens is planned to be supported soon:

- OpenID Connect RelyingParty (OpenID Connect)
- X.509 (RFC 3820)

3.26.3 User interfaces and APIs

The LDAP Facade is accessible via a web application for management and self-registration. It also provides an LDAP interface for authentication. This interface includes smart retrieval of needed pieces of information to perform the authentication step.

3.26.4 Support for Virtual Organisations

The support for Virtual Organizations is done via group management as well as attribute queries or aggregation (for third-party group management).

The option to run the LDAP Facade as an attribute authority is under development.

3.26.5 Dependencies on other technologies

As the LDAP Facade is a Java Enterprise Application it depends on the Java Virtual Machine (VM). It uses the OpenSAML libraries for interaction with the SAML Identity providers. For the LDAP interface an interceptor in the Apache DS is used.

For the web-interface JBoss and Dragonfly are supported.

3.26.6 Operational overview

A typical LDAP Facade deployment is one instance per site. The operation is similar to a SAML SP plus an LDAP server.

At KIT, currently the service in production runs on 5 VMs.

3.26.7 Expected level of support

The Karlsruhe Institute of Technology has committed to support the tool for 37 member organizations from previous projects. KIT also works on extending the LDAP Facade within INDIGO and use it as a pilot in AARC.

The LDAP Facade is under active development. The milestones for the coming years are:
12/2015

- Public prototype for DFN + eduGAIN

- SAML-Token support (solve the password exposure issue)

03/2016
- Reduced-Attribute requirements
  - For simplified support of additional IdPs
- Account Linking
- Support additional IdPs / federations
  - e.g. Umbrella, B2ACCESS, ...

06/2016
- OpenID Connect support
- Integration with Globus grid-security-infrastructure
  - i.e. GridFTP to use LDAP-Facade for (UID, [GID])

12/2016
- Support for 3rd party group membership (e.g. via Attribute Authorities or SCIM)
  - e.g. Unity (B2ACCESS), VOMS-SAML, ...

3.27 IdProxy

IdProxy is an open-source implementation for an OAuth 2.0/OpenID Connect provider and IdP proxy against SAML IdP.

- Source code and documentation: https://github.com/its-dirg/IdProxy
- Licence: GPL-3.0

3.27.1 Features

IdProxy can be used to:
- Add authentication methods, such as YubiKey, to an underlying IdP (multi-factor authentication);
- Add an OpenID Connect frontend to a SAML IdP;
- Provide the same SSO token for different underlying IdPs. Thus, all clients (SPs) will only be aware of the SSO token from the proxy;
- Provide SAML and OpenID Connect interface to CAS;
- Act as a standalone OAuth 2.0/OpenID Connect Provider;
- Act as a standalone SAML IdP;
- Transport group membership information through eduPersonAffiliation/eduPersonScopedAffiliation attributes

3.27.2 Supported standards

SAML, OAuth 2.0, OpenID Connect, LDAP

3.27.3 User interfaces and API

No UIs or APIs provided. Initial settings and runtime parameters can be configured by editing a set of configuration files.

3.27.4 Support for Virtual Organisations

No support.

3.27.5 Dependencies on other technologies

- Written in Python
- Depends on pysaml2[49] (SAML2 library implementation in Python)
- Depends on pyoidc[50] (OpenID Connect library implementation in Python)
- Can acquire user information from an LDAP directory
- Can perform authorisation against a CAS server

3.27.6 Operational overview

IdProxy can be deployed as:

- An OpenID Connect frontend to a SAML IdP;
- A SAML IdP proxy;
- A SAML and OpenID Connect interface to a CAS server/LDAP user directory;
- A standalone OAuth 2.0 / OpenID Connect Provider;
- A standalone SAML IdP.

IdProxy does not support any high availability deployment options.

3.27.7 Expected level of support

IdProxy is supported by the open-source community. However, it should be noted that the code base has not been updated since June 2014.

[49] https://github.com/its-dirg/pysaml2
[50] https://github.com/its-dirg/pyoidc

4 Commercial solutions

This section presents some examples of commercial solutions for identity provisioning or identity management that may be relevant to R&E communities.

4.1 Auth0

Auth0 allows public and private entities to expand their identity infrastructure out to third parties by providing single sign-on services through the abstraction of different login and identity services into a single API. It is included in this document as it has been mentioned in the responses of the survey conducted during the requirements gathering activities within task JRA1.1. Auth0 can serve as an IdP supporting several authentication mechanisms, and also an IdP proxy for external IdPs that need to be integrated. Auth0 supports groups and roles for the users. A GUI is provided for account management, as well as an SDK for the integration of apps with the platform. Supported standards include SAML, OpenID Connect, JSON Web Token, OAuth 2.0 / 1.0a, OpenID and WS-Federation. The core Auth0 APIs are:

- **Authentication API:** For enterprise authentication or OpenID operations
- **Management API:** For administrative tasks such as creating accounts or changing settings

Auth0 supports four deployment models:

1. Multi-tenant cloud service running on Auth0's cloud.
2. Dedicated cloud service running on Auth0's cloud.
3. Dedicated cloud service running on Customer's cloud infrastructure.
4. On-premises virtual appliance running on Customer's data centres

Pricing increases with the number of users, however there are free plans offered for developers. The expected level of support depends on the customer's plan.

4.2 Facebook Login

Facebook Login[51] is a set of authentication APIs from Facebook that developers can use to allow users to log into third-party websites, applications and devices with their Facebook account. When people choose to log in with Facebook, they can share their real identity through their public profile, which includes a person's real name, a profile picture, their gender and their locale. Facebook Login can be implemented with the provided SDKs for JavaScript, iOS, and Android. To enable browser-based access without using these SDKs, the login flow involves HTTP redirects to an endpoint that will display the Facebook login dialog and, based on the user's response, return an appropriate access token. An access token is an opaque string that identifies a user and can be used by a third party to make Facebook Graph API calls in order to read, modify or write a specific person's Facebook data on their behalf.

[51] https://developers.facebook.com/docs/facebook-login

Facebook Login supports over 30 permissions[52] in order to determine which information Facebook users will share with the connected system. Facebook Login can be used to complement an existing user account system. In this case, it is up to the developers to support merging the account information created by their application with the information from that person's Facebook account.

4.3 Google Apps federated login

Reference: https://developers.google.com/google-apps/sso/openid_reference_implementation?hl=en
Google Apps offers an OpenID API that allows end users to securely sign in to third party web sites using their Google Apps user account.

Google Apps API supports the OpenID 2.0 Directed Identity protocol, allowing any hosted domain to provide authentication support as an OpenID provider. On request from a third-party site, Google authenticates users who are signing in with an existing Google Apps account, and returns to the third-party site an identifier that the site can use to recognize the user. This identifier is consistent, enabling the third-party site to recognise the user across multiple sessions.

Notably the OpenID API supports also the extensions: OpenID Attribute Exchange and the OpenID+OAuth Hybrid protocol. The second extension allows combining an OpenID request with an OAuth request. In the end Google Apps federated login service can act as an IdP for some use cases, or as an authenticator when connected with another identity. Google Apps provides also a Federated Login API, which can be used to integrate several OpenID Connect IdPs in a service provider.

While as a consumer using Google Apps credentials is free, the federated login API is pay-per-use.

This login is widely used as a catch-all login for users who do not have the ability to login with their institutional login.

5 Comparison of authentication and authorisation technologies

The tools and technologies described above are compared in this chapter in terms of their respective feature sets. More specifically, we have defined feature sets pertaining to five distinct use cases, namely authentication, attribute management, discovery services, credential translation and attribute aggregation. For each use case, there is a table that includes the features used for the comparison in the rows, and one column for every tool or technology. The purpose of this comparison is not to suggest a preferred tool for every use case, since different communities have different requirements and there may not be one-size-fits-all solution.

It should be noted that some of the tools in this document satisfy several use cases; for example a piece of software can be used as an attribute authority but also as an attribute aggregator. For this reason, we avoided grouping the tools in the previous section based on their use cases and features, since there would have been either duplicate or partial descriptions for some of the tools. However, the grouping is performed in this chapter

[52] https://developers.facebook.com/docs/facebook-login/permissions

for comparison purposes, and tools may therefore appear in more than one table, based on the use cases they support.

5.1 Authentication

Authentication technologies are the software and libraries that can be used to allow users to authenticate, and optionally if allowed by the identity provider and requested by the service provider, providing the identity information to another service.

The features used for the comparison are the following:

- Authentication workflow
  - How do users authenticate to a service using the authentication technology? For example, is it username and password authentication or is it multi-factor authentication?
- Information on the LoA
  - Can the information about the IdP level of assurance be embedded in the authentication workflow, and can they be consumed by the service provider.
- Hierarchical organisation of the user community (groups and roles for the users)
  - User communities, or Virtual Organization, do not have a flat structure but are often organized with roles and groups to identify users with different capabilities and different access levels. Can the authentication mechanism add information about the VO organization besides the user identity?
- Import/export of user data
  - How easy is to migrate from the authentication technology to another authentication technology? Can the user data be exported?
- Possibility of high availability (HA) deployment
  - From an operational standpoint, can authentication services be deployed in a high availability configuration?
- Ownership (institution/project)
  - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research & education)
  - Under which licence is the tool released
- Dependencies on other technologies
  - Is the tool dependent on other tools or technologies? This can be important in case other technologies are proprietary.
- Expected level of support
  - To what extent, at the moment of writing, is the support assured by the owning institution?

Table 1, Authentication technologies comparison table

	Shibboleth	LCMAPS	Kerberos	Moonshot	simpleSAMLphp	UNITY
Authentication workflow (username/pass, multifactor)	Password, RemoteUser, RemoteUserInternal, X509, X509Internal, SPNEGO/Kerberos, IPAddress, External	X.509 proxy certificate	Username/password, OTP, Kerberos ticket	Username/password (any RADIUS EAP-supported mechanism)	Username/password from user repository (SQL/LDAP/ RADIUS), X509 authentication through userCertificate LDAP attribute, multifactor (e.g., Yubikey), Social Media identity (Facebook, LinkedIn, MySpace, Twitter, Windows Live), ADFS	Username/Password, Client Certificate, LDAP, Social Media identity (Facebook, Google, GitHub)
Supported standards	SAML 1.1/2.0, X509, Kerberos, LDAP, SQL	X.509 (RFC5280 and RFC3820), VOMS	RFC 4121, RFC 4120	RFC3748, RFC5247, RFC7055	SAML 1.1/2.0, X509, OpenID, OAuth 2.0, Kerberos, VOOT, SQL, LDAP, RADIUS	SAML 1.1/2.0, X.509, OIDC, LDAP
LoA support	It is possible to support it by extending the Shibboleth	Yes, via the VO-CA-AP plugin.		Yes, in the sense that it can transport LoA information through SAML	Yes, in the sense that it can transport LoA information through SAML	Yes, only an attribute containing the LoA (not attribute level)

	configuration, but it is not supported natively.			attributes	attributes	
VO hierarchical organisation	Group memberships can be retrieved by issuing SAML 2.0 AttributeQueries to an Attribute Authority configured to retrieve additional attributes from its database and releasing them inside the User Session together with other attributes.	Yes, it supports VO	User community can be split into the realms	VO-like support through so-called communities of interest (COI) that are defined at the trust router level (i.e. not inherent to GSS-EAP mechanism). COI information is defined in vendor-specific RADIUS attributes.	Group memberships can be retrieved from an API service protected with OAuth 2.0 using the VOOT protocol and added to the list of attributes received from the IdP. Alternatively, issuing SAML2 AttributeQuery to an Attribute Authority can be supported	Yes
Import and export user data	Yes, importing/exporting underlying user repository	Not relevant	Only to other Kerberos DB	User data not relevant to technology	Yes, importing/exporting underlying user repository	Yes, importing/exporting underlying user repository
HA deployment	Yes	Deployed in the service	Yes	RADIUS service can be run in HA environments	Yes, through multiple memcached service instances	Relying on Database layer
Dependencies on other technologies	JAVA JRE, OpenSAML Java Application Server/Container (Tomcat, Jetty), Apache httpd mod-shib2	OpenSSL, VOMS C library		RADIUS (FreeRADIUS) for transport, SAML for attribute encapsulation.	PHP, Web servers (Apache, nginx, IIS among others), user repository (SQL database, e.g. MySQL or	SQL Databases (MySQL, PostgreSQL), Java

					PostgreSQL, , LDAP directory (OpenLDAP) or a RADIUS interface (OpenRADIUS), memcached for user session management	
Licence	Open Source	Open Source (Apache-2.0)	Open Source	Open Source	Open Source (LGPL 2.1)	Open Source, Permissive OpenBSD Licence
Expected support level	Supported/funded by Shibboleth consortium, dedicated dev team.	Supported by NIKHEF	Fully supported by Linux OS distributions	Supported by Jisc, user communities	Large user community (dev team external contributors, mailing lists)	Supported by ICM, JSC and Funded by PLGrid

5.2 Attribute management

The attribute management services store information associated with a user credential, or more user credentials. Identity providers usually provide information, attributes, to describe the user identity, where attribute providers are used by third parties to associate other, community-specific, information with a user. Typical examples of attributes are membership to a research group, access rights to a service or a dataset or specific roles within the collaboration.

The features used for the comparison are the following:

- Input standards supported
  - To provide users' attributes the service has to consume information about user identity, in other words user needs to authenticate on the service. How can the user authenticate?
- Output standards supported
  - In which format, and standards, the attributes can be communicated to another service by the attribute authority.
- Need to explicitly configure IdPs in the service
  - Has every IdP to be configured in the service?
- Need to explicitly configure SPs in the service
  - Is a configuration in the attribute authority required for the services consuming the attributes?
- User membership life cycle management (configurable/automatic expiration of user membership)
  - In case attributes are used for the VO membership, does the tool also support membership expiration and acceptance of policies?
- Hierarchical organization of the user community (groups and roles for the users)
  - Does the tool support groups within the VO?
- Delegated administration of user groups
  - Can management of users groups within the community be delegated to group-managers?
- Import/export of user data
  - Can the information about users can be exported in standard formats, for example XML?
- Possibility of HA deployment
  - From an operational standpoint, can authentication services be deployed in a high availability configuration?
- Ownership (institution/project)
  - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research & education)
  - Under which licence is the tool released
- Dependencies on other technologies
  - Is the tool dependent on other tools or technologies. This can be important in case other technologies are proprietary.
- Expected level of support
  - To what extent, at the moment of writing, is the support assured by the owning institution?

	VOMS	HEXAA	COmanage	Grouper	Perun	UNITY
Input standards	X.509	SAML2	SAML (via Apache)	SQL, LDAP, XML	SAML2, X.509	SAML2, X.509, LDAP, OIDC
Output standards	X.509, SAML	SAML2	VOOT, LDAP, SAML (via Shib IdP)	LDAP, VOOT, SCIM, XML	SAML2, VOOT	OIDC, SAML
IdP configuration	CA configuration	It requires configured acceptable IdPs	No	No	It requires configured IdP	Yes
Handle attribute release consent	Not directly	Yes	Not directly	No	Via the IdP	Yes
SP configuration	Configure authoritative VOMS for the supported VOs	It requires to configure the SPs specific for the VO	Not by default	No	It requires configured SP	Yes
Membership life-cycle management	Yes	No	Yes	No	Yes	No (will be supported)
VO Organization (groups, roles, etc.)	Yes. Groups and Roles	Yes, fine grained VO organization	Yes	Yes	Yes. Groups, Roles, Attributes	Yes, (Hierarchical) Groups, Roles, Attributes
Delegated organization of the	Yes	No	Yes	Yes	Yes	Yes

VO groups						
Import and export of user data	Partially yes	Partially yes, through proprietary API	Yes, through API and pluggable provisioners. 1.1.0 release will introduce significant new import capabilities.	Yes, through Grouper Loader, Provisioning Service Provider, Web Services, and Import/Export tool	Yes, via various protocols. Periodical synchronization supported as well.	Yes
HA deployment	Available	Not available	Yes	Yes	Partially supported	Yes
Dependencies on technologies	TBD	PHP, SimpleSAMLphp, MySQL, Apache	PHP/Apache, RDBMS (PostgreSQL, MySQL)	Java, container, RDBMS	Java container, Apache, Shibboleth	SQL Databases (MySQL, PostgreSQL), Java
Licence	Open Source	Open Source	Open Source (Apache2)	Open Source (Apache2)	Open Source, FreeBSD licence	Open Source Permissive BSD Licence
Expected level of support	Supported by INFN, bug fixes.	Supported by SZTAKI and NIIFI	Supported by Internet2 TIER, various grants, and other sources	Supported by Internet2 TIER, various grants, and other sources	Supported by CESNET and Masaryk University. Maintenance and development.	Supported by ICM, JSC and Funded by PLGrid

5.3 Authorisation

Services can implement authorisation policies based on external information or locally. For distributed infrastructures in particular, it is common for services to use an external policy engine to take authorisation decisions. The purpose for this configuration is to support centralised management of authorisation policies for security reasons, as well as to simplify configuration at service level.

The following features are used in the comparison table:

- Type of input attributes
  - Which type of attributes/information is the service able to consume (e.g. SAML attributes, JSON formatted information)
- Support for policy management
  - Does the service support ingestion of policy rules from external services? Can a more than one service be deployed in a hierarchical configuration?
- Possibility of HA deployment
  - From an operational standpoint, can authentication services be deployed in a high availability configuration?
- Dependencies on other technologies
- Is the tool dependent on other tools or technologies? This can be important in case other technologies are proprietary.
- Ownership (institution/project)
  - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research & education)
  - Under which licence is the tool released
- Expected min support (years)
  - For how long, at the moment of writing, is the support assured by the owning institution?

Table 2. Authorization technologies comparison table

	Argus	LCMAPS	mod_auth_mellon
Type of input attributes	PAP Configurable, PEPd configurable via new plugins (e.g. PIP). Based on SAML2-XACML2 attributes, currently primarily X.509 and VOMS based (DNs and FQANs). Need to be profiled for interaction between clients and servers.	X.509 proxy certificates with VOMS ACs or derived attributes (subject DN, FQANs etc.).	SAML2 attributes
Support for policy management	Yes, PAP can import policies from remote PAPs.	Config file allows complicated flows of plugins, including callouts to remote services (such as Argus).	Basic policies via Apache HTTP server config files
LoA support	In principle yes, in practice needs extra plugins.	Yes, via lcmaps-plugins-vo-ca-ap	Yes, if LoA information available through SAML attributes
HA deployment	Yes, several solutions are possible. E.g. running multiple PEPd-s or PDPs. The LCMAPS Argus plugin has support for multiple endpoints. Also failover setups such as in http://wwwae.ciemat.es/~delgadop/argus/	No, not really applicable (not a service but a library).	Yes, provided an Apache HTTP server HA deployment
Dependencies on other technologies	Numerous Java libraries (shipped along) such as cANL-java, OpenSAML, voms-java.	VOMS C-API, OpenSSL	Apache HTTP server, OpenSSL, lasso

Ownership / maintenance	INFN (Java) / Nikhef (C)	Nikhef	UNINETT
Licence	Apache-2.0	Apache-2.0	GPL-2
Expected support	Supported at least for bug fixes	Supported for security fixes and feature requests, provided such a feature or enhancement is also of interest for NL-NGI, EGI.eu or a community which is supported by the NL-NGI or Nikhef itself	Not specified. Project supported by UNINETT and the open source user community

5.4 Credential translation

The key for interoperating infrastructures is the possibility to translate user credentials from one standard to another standard. Authentication and authorization data is usually exchanged with the service provider through security assertions, or through short-lived credentials.[53] To access services supporting different type of credentials, based on different standards, the goal of the token translation services is to translate the information contained in the input assertion and encode equivalent information in an assertion based on a different standard, or generate a short lived credential for the same user, so users can access services that support different standards.

The comparison will be performed using the following features:

- Input standards supported
  - The formats that the user's credentials can be ingested by the services
- Output standards supported
  - In which standards, the output assertions, or credentials, can be encoded
- Interfaces with users (can user get credentials out, can only services use it?)
  - Has the service a proprietary API, or a GUI?
- Need to explicitly configure IdPs in the service
  - Has every IdP to be configured in the service?
- Need to explicitly configure SPs in the service
  - Is a configuration in the attribute authority required for the services consuming the attributes?
- Possibility of high availability (HA) deployment
  - From an operational standpoint, can authentication services be deployed in a high availability configuration?
- Ownership (institution/project)
  - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research & education)
  - Under which licence is the tool released
- Dependencies on other technologies
  - Is the tool dependent on other tools or technologies? This can be important in case other technologies are proprietary.
- Expected min support (years)
  - For how long, at the moment of writing, is the support assured by the owning institution?

[53] The TCS service is not included in this section, since it is not entirely fulfilling the described use case. In some use cases TCS can be used as a token translation service between SAML (eduGAIN) and X.509 credentials, but in the current deployment it requires users to manage the X.509 credentials once obtained from TCS. Future developments of TCS may provide features more similar to the other token translation services.

Table 3. Credential translation services comparison table

	STS	CILogon	ADFS	LDAP Facade	Unity	IdProxy
Input credential types supported	Username Token, SAML	SAML, OpenID-Connect	Active Directory	SAML, planned: OIDC, X509	SAML2, OpenID, OAuth, local	SAML, LDAP
Output credential types supported	X.509 certificate, Proxy certificate	X.509 certificate, Proxy certificate	SAML	LDAP	SAML, OpenID, OAuth, LDAP, Kerberos	SAML, OAuth 2.0 / OpenID Connect
Interfaces with other services	VOMS, CA, LDAP	MyProxy CA, (VOMS)	WS-*			CAS, LDAP, SAML2 IdP
UI for the users	N/A	Web, command line	Web	Web, any SSH client, including non-web	Web	No UIs or APIs provided. Admin needs to edit a series of configuration files.
HA deployment		TBD	Multiple ADFS servers and IdMaaS endpoints can be deployed		Possible (Relying on backend database)	No
Licence	http://www.apache.org/licenses/LICENSE-2.0	http://grid.ncsa.illinois.edu/myproxy/license.html, NCSA and BSD and ASL 2.0	Various: some proprietary, some proprietary with source available, some open source (see description)	GPL	Open Source (BSD)	Open Source (GPL-3.0)
Expected level of	Community driven.	Production software	Very high for	In production,	In production,	Supported by the

support		in US	Office365 and Azure.	actively developed, installing pilots available	actively developed	open source community. Code base has not been updated since June 2014.

5.5 Discovery services

In a federation, users and services have to discover the relevant IdPs (and other services) that are part of the federation. The goal of the following services and libraries is to help to match users and services with the IdP.

The comparison will be performed using the following features:

- Geographical localisation of IdPs
  - Can IdP be ordered for example to show first the IdP closer to user location (e.g. in the same country)
- UI internationalisation
  - Is the tool supporting multiple languages?
- Search/filter capabilities
  - e.g. free text search
- Preferred user choice
  - Does the tool remember the preferred user's choice?
- Input standards for IdP metadata
  - Which type of data is the service able to ingest, in which format?
- Ownership (institution/project)
  - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research and education use)
  - Under which licence is the tool released
- Expected level of support
  - To what extent, at the moment of writing, is the support assured by the owning institution?

Table 4, discovery services comparison table

	SWITCHwayf	DiscoPower	DiscoJuice	Shibboleth DS	Shibboleth EDS
Geographical filtering of IdPs	Not supported	Partial, through IP geo-location	Yes, through HTML5 geo-location API (needs user consent)	No?	No?
UI internationalization	Yes, manually on a specific language file. Languages en, de, it, fr and some other languages are included.	Yes. Translated UI elements based on JSON-formatted configuration files and localised IdP names/descriptions are read from XML metadata	Yes	Yes	Yes, language detecting on browser settings
Search and filter capabilities	Search-as-you type or selection from a list of organisations	Search-as-you type or selection from a list of organisations that are organised in tabbed views (usually grouped by country)	Filtering by country. Search-as-you type or selection from a list of organisations	Search-as-you type or selection from a list of organisations	Search-as-you type or selection from a list of organisations
Preferred user choice	Yes	Yes	Yes	Yes, by cookie	Yes
Input standards for IdP metadata	XML	SAML V2.0, SAML V2.0 Attribute Extensions V1.0	SAML V2.0	SAML V2.0	SAML V2.0
Ownership/Project	SWITCHaai/SWITCHwayf project	UNINETT AS	UNINETT AS	Shibboleth Consortium	Shibboleth Consortium

Licence	BSD (as-is)	Open Source (LGPL 2.1)	Open Source (LGPL 3.0)	Open Source (Apache Software License 2.0)	Open Source (Apache Software License 2.0)
Expected level of support	Not defined	Unspecified but currently supported by large user community (dev team, external contributors, mailing lists)	Unspecified. Not production ready (August 2014).	Unspecified but currently supported by large user community (dev team, external contributors, mailing lists)	Unspecified but currently supported by large user community (dev team, external contributors, mailing lists)

5.6 Attribute aggregation

Users may have separate accounts on different IdPs and, at the same time, a particular user can often be part of several communities, while every community usually uses their own attribute authority service to register the users' community attributes. In these cases, services have to deal with multiple entities to decide on authorisation. In the typical implementations of federated authentication, the use of a third party attribute provider is not supported, hence the need for aggregating attributes from different sources of information (IdPs and/or attribute authorities) and providing them in a single assertion to a service in order to enable federated authorisation to that service.

The comparison will be performed using the following features:

- Attribute retrieval standards/protocols supported (e.g. SAML2 AttributeQuery, OAuth, VOOT, REST, SOAP, LDAP, SQL)
- Shared (non-targeted) identifier required
- Possibility of high availability (HA) deployment
    - Can the service be deployed in high availability?
- Ownership (institution/project)
    - Who develops and maintains the tool
- Licence (Open Source, Commercial, free for research & education)
    - Under which licence is the tool released
- Expected level of support
    - To what extent, at the moment of writing, is the support assured by the owning institution?

Table 5, Attribute aggregation services

	Unity	OpenConext	Perun	Moonshot	SimpleSAMLphp
Attribute retrieval standards/protocols	SAML2 Attribute Query , OAuth, LDAP	SAML2, OAuth, VOOT1-2, XACML, OpenID Connect OP (future)	SAML2, VOOT, LDAP, SQL, CSV	Attributes are encapsulated in a SAML assertion included in a RADIUS attribute. A mix of RADIUS attributes and one SAML attribute statement (as per standard) are supported.	The VOOT groups module can fetch group memberships from an API service protected with OAuth 2.0 using the VOOT protocol (versions 1 and 2 are supported) and add them to the list of attributes received from the IdP. The Attribute Aggregator module can issue SAML2 attribute queries to Attribute Authorities that support SAML2 SOAP binding. The Attribute-from-rest-api module can request attributes from REST APIs in JSON format.
Shared (non-targeted) identifier required	Yes	Yes	Yes	No, but it is supported	Yes
HA deployment	Possible (Relying on backend database)	TBC	Partially supported	Not yet	See HA deployment options for SimpleSAMLphp above
Ownership	ICM	SURFnet	CESNET/Masaryk University	Jisc/Painless Security	VOOT groups module: OpenConext

					Attribute-from-rest-api module: NIIF Attribute-from-rest-api module: NIIF
Licence	(Open source) BSD	Apache 2.0	Open Source, FreeBSD licence	Open Source	Open Source
Expected level of support	In Production, actively developed	Actively developed	In production, actively developed	In production, actively developed	Active community support

6 Conclusions

The goal of this document was to provide an assessment of the technologies and standards that are relevant for R&E AAIs. To achieve this, the AARC consortium worked collaboratively to produce detailed information on a significant number of standards, software products and services for federated access, based on the partners' extensive knowledge and hands-on experience. The list of technologies and tools included in the analysis is by no means exhaustive, however, the provided information should enable AARC stakeholders to identify the technologies that fulfil their specific use cases, and to understand what they could expect from current state of the art on federated AAI.

To support this, the tools and technologies described here have been compared in terms of their respective feature sets. More specifically, distinct feature sets have been identified pertaining to five distinct use cases, namely: authentication, attribute management, discovery services, credential translation and attribute aggregation. For each of these use cases, a table has been produced to present the features used for the comparison in the rows, and one column for every relevant tool or technology. The purpose of this comparison was not to suggest a preferred tool for the identified use cases, since different communities have different requirements and there may not be one-size-fits-all solution. Finally, it should be noted that this document is expected to serve as valuable input for the work in the pilots work package (SA1).

Glossary

A(A)AI	Authentication, (Accounting) and Authorisation Infrastructure
ABAC	Attribute Based Access Control
ABFAB	Application Bridging for Federated Access Beyond the web
AC	Attribute Certificate
API	Application Program Interface
Argus	EGEE/gLite Authorization Service, widely used in EGI infrastructure
CA	Certification Authority
CRL	Certificate Revocation List
CRUD	Create, Read, Update, Delete
CSV	Comma Separated Values
DN	Distinguished Name
EAP	Extensible Authentication Protocol
EDG	European Datagrid (former project)
EEC	End-Entity Certificate
EGEE	Enabling Grids for E-sciencE (3 projects)
EGI	(formerly) European Grid Infrastructure
EMI	European Middleware Initiative, former Grid middleware project in Europe.
EPEL	Extra Packages for Enterprise Linux, add-on repository for RHEL/CentOS/Scientific Linux
gLExec	sudo-like tool using grid credentials as input
GridFTP	GSI-version of FTP (both protocol and service)
gsissh	GSI-version of SSH
GSSAPI	Generic Security Service API
GUI	Graphical User Interface
GUMS	Grid Identity Mapping Service, authorization service used by OSG
HA	High Availability
HSM	Hardware Security Module
HTC	High Throughput Computing
HTTPS	Secure HTTP
IdP	Identity Provider
IGTF	International Grid Trust Federation
JSON	JavaScript Object Notation
LCAS	Local Centre Authorization Service
LCMAPS	Local Credential Mapping Service
LDAP	Lightweight Directory Access Protocol
LHC	Large Hadron Collider
NREN	National Research and Education Network
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OGF	Open Grid Forum
OH	Obligation Handler
OpenSSL	widely used SSL/TLS/Cryptography libraries written in C
OSG	Open Science Grid
PAP	Policy Authorization Point
PDP	Policy Decision Point,
PEP	Policy Enforcement Point
PIP	Policy Information Point
PKCS12	Public-Key Cryptography Standards number 12
PKI	public key infrastructure
PKIX	PKI using X.509
POSIX	Portable Operating System Interface, a set of standard operating system interfaces based on the Unix operating system.
PRP	Policy Retrieval Point

RA	Registration Authority
RadSec	Protocol for RADIUS over TCP with TLS encryption
RADIUS	Remote Access Dial-In User Service, a protocol widely used for remote access. eduroam uses this protocol
RBAC	Role Based Access Control
REST	REpresentational State Transfer
RP	Resource Provider, a different term for a Service Provider
SAML	Security Assertion Markup Language, an OASIS standard
SAML-ECP	SAML Enhanced Client Protocol / Enhanced Client Proxy
SCAS	Site Central Authorization Service
SOAP	Simple Object Access Protocol
SQL	Structured Query Language (database programming language)
SP	Service Provider
STS	Security Token Service, credential translation service defined in WS-Federation
TIER	Trust and Identity in Education and Research, an Internet2 initiative
TLS	Transport Layer Security
UMD	Unified Middleware Distribution
VOMS	Virtual Organization Membership Service
W3C	World Wide Web Consortium
WLCG	worldwide LHC Compute Grid
WSDL	Web Service Definition Language
XACML	eXtensible Access Control Markup Language, an OASIS standard
XML	eXtensible Markup Language, a W3C standard
XSLT	eXtensible Stylesheet Language Transformation