31-03-2013

Open Call Deliverable OCK-DS1.1
Final Report (HEXAA)

Open Call Deliverable OCK-DS1.1

Grant Agreement No.:	605243
Activity:	NA1
Task Item:	10
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	MTA SZTAKI
Document Code:	GN3PLUS14-1316-84
Authors:	István Tétényi (MTA SZTAKI), Mihály Héder (MTA SZTAKI), Zsuzsanna Magyar (MTA SZTAKI), Kristóf Bajnok (NIIFI)

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7 2007–2013) under Grant Agreement No. 605243 (GN3plus).

Abstract

This document is the final deliverable of the HEXAA Open Call project. HEXAA has addressed successfully the legal and technical challenges by carefully studying the attribute requirements of research communities and the relevant legal constrains of using attributes. Based on these findings the development of new attribute authority software was started. Today, HEXAA is in service at eduID.hu is able to support communities from different identity management federations with SAML based virtual organizations, profiles attributes and roles.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The Higher Education External Attribute Authority (HEXAA) project is one of the Open Call participants of the GN3plus project whose activity is focusing on the Authentication and Authorization Infrastructure area. We are all accustomed to the environment of ubiquitous Internet. One of the biggest challenges ahead of us is harnessing the power of the Internet for research as best we can. The access to the Internet is considered resolved although big differences exist in network capacity and service level parameters. The access to research infrastructures for mainly web based resources needs a consistent, interoperable, trustful and distributed approach. This is being resolved by the steady growth of identity management federations of the national research and education networks (NRENs). eduGAIN is successfully bridging the gaps between national federations in different world regions. Consequently, the end-user's identity can now be trustfully checked within the ever growing eduGAIN federation.

Problems almost immediately come to light when researchers from different federations would like to participate in a joint project accessing resources from a distributed environment by using their home identity. This is manifested in the lack of proper attributes that are required to sufficiently authorize and also organize researchers in accessing the resources. HEXAA is addressing this niche in the environment with the capability of handling attributes for research groups in a standardized, flexible way.

The first section of the document summarizes our findings about the attribute handling in identity federations, and also gives some insight of the results of a user survey, and sums up the technical challenges that one has to overcome in order to support federations with attributes.

Section two focuses on the end-product of HEXAA, a piece of software that implements an attribute authority. There are three standpoints of presenting HEXAA: the developer view, the operational view and finally the user view.

Section three addresses the issue of integrating HEXAA with federations be it national one or eduGAIN and also sums up the most important legal requirements that any attribute authority provider must keep. This section is a real guideline from both technical and legal aspects.

HEXAA project delivered all of the Milestones on time but more importantly it developed a brand new attribute authority that is integrated to the Hungarian NREN federation and soon will be part as a service of eduGAIN. Modules for existing open source software are additional results of the HEXAA project that allow the collaboration of e-science gateways, content management systems and cloud environments (OpenNebula and OpenStack). The legal study is also very important as the eduGAIN community does not have a fresh view on the legally binding operational requirements of attribute authorities.

1 Introduction

The objective of this document is to give an extended, factual summary of the results of the HEXAA open call project. The behind the scene details are presented in the Milestone documents. Our purpose with the final document is to create a concise view of HEXAA in the hope to use it either in other projects or in some cases as guidelines for interested parties to find a common ground of collaboration. Our view is that the use of Attribute Authorities in identity federations is very uneven and this hinders the overall pan-European research community to create sizeable research groups. We truly believe that our work in the HEXAA project helps to overcome this problem.

An attribute authority is a facility that is able to store information about collaboration space elements - virtual organizations, user roles and rights - and is able to provide this information using a secure standard (SAML 2.0). An attribute authority is a special service that provisions attributes to service providers (SP) when an identified user tries to access their service.

Attribute authorities can be deployed within an institution or community to facilitate group and rights management. However they are unavoidable if the management of roles and other user attributes is a task of a virtual organization (VO) or a project body. In this case, authentication is provided by the users' home institution (Identity Provider, IdP), as well as basic identity attributes, while the management of roles and other attributes are performed at an attribute authority. A possible scenario when this attribute authority is operated by a trusted third party, such as an NREN.

Attribute authorities help home institutions and service providers by moving the responsibility of managing the information to more natural and dynamic actors: to virtual organization (project) managers and to the users themselves. This model is sound as it limits the responsibility of the home institution to the identity checking of the person in question and establishes room for authorization information coming from a third party the AA.

The goal of the HEXAA project to prove that an efficient, flexible, easy to use attribute authority can be created that helps the work of researchers in federations like eduGAIN or eduID. The HEXAA project within GEANT3plus Open Calls is a "proof of concept" type of project.

The structure of the final document follows the structure that was proposed and contractually agreed in the HEXAA project. This is in line with the project plan that in the first half of the project the focus was on study documents partly on the legal issues of attributes, partly analyses on the user attributes in federations. Section two is summing up our work on the study area. Section three focuses on the proof of concept realization of an attribute authority. Section four addresses the problem of how an attribute authority can be integrated into identity federations.

We wanted to create a final delivery document with a readable size, instead of a 200+ pages document. Details can be found in the milestones and the authors are ready to answer questions from the readers even after the project finished.

Prunk-Éger Edgár Bana Tibor Bajnok Kristóf Tétényi István Szabó Gyula
Bálint Márton Magyar Zsuzsanna Soltész Balázs Tenczer Szabolcs Héder Mihály

2 Analysis on handling user attributes within federations

The “Analyses on handling user attributes within federations” is a study type of document. In this section we outline the most important findings of our results. The comprehensive background document was delivered as HEXAA MS 2.1.

Our work was carried in parallel with the legal analyses of using external attribute providers[1] (EAP). Although the legal study is not part of this final deliverable, the most important requirements are included in Section four. More details can be found in the HEXAA milestone documents MS 3.1 and MS 3.2 are giving a deep insight of the legal aspects of the user attributes.

2.1 Approach

Our approach used the following elements:

a) checking and analyzing the scientific and research network community activities in the field
b) making several interviews using videoconferencing to deepen our understanding of the problem space and getting feedback from the scientific research groups about their approach
c) planning and carried out a web-based user survey to understand the end-user’s view on the subject
d) assessed the technical challenges
e) developed a categorization of attribute authority architectural relationship to SAML federations

2.2 Results

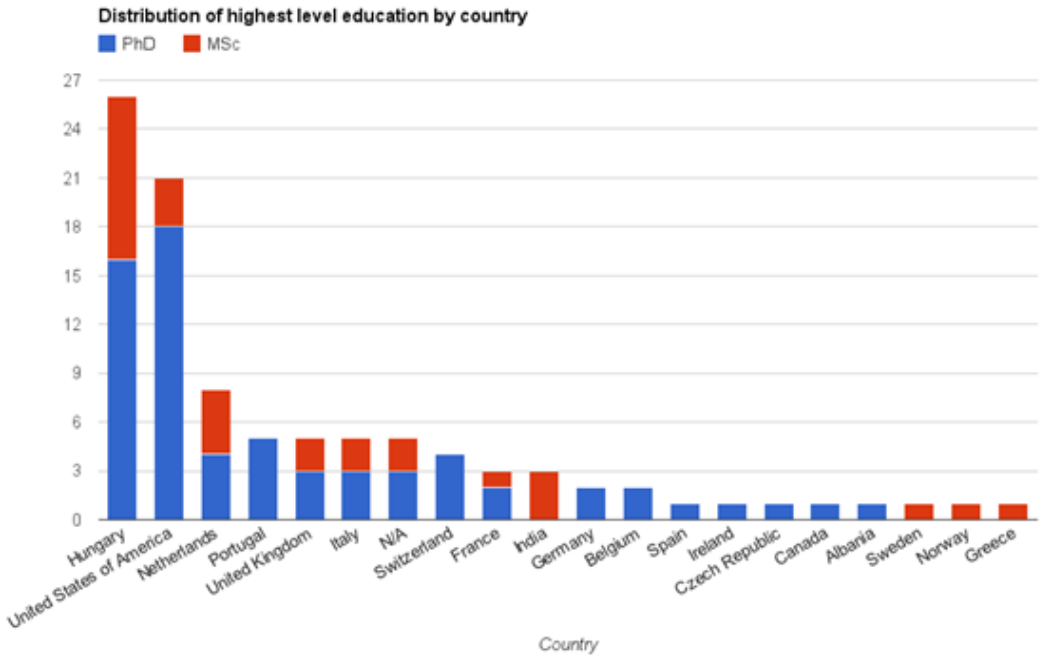
a) Three overlapping analyses areas were identified:
   - the research community around European Grid Infrastructure (EGI) has several years of experience using attribute authority functionality that is based on using VOMS and X.509 certificates
   - the research community that has concentrated its efforts in the Federated Identity Management for Research (FIM4R) framework
   - the community around the working group of Research and Education Federations (REFEDS), that works with international collaborations including eduGAIN, Internet2, etc.

[1] The legal term of External Attribute Authority (EAP) and the technical term Attribute Authority (AA) are both used in this document.

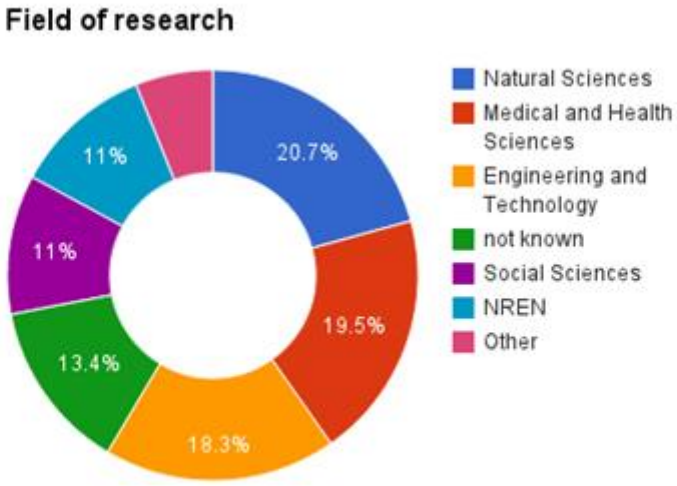
b) The list of interviews is detailed in MS 2.1, Annex A. There is a broad consensus that the federated identity management results so far manifested mainly in the authentication (AuthN). (This is the only requirement to access Eduroam service.) However, a successful authentication is only a precondition to facilitate work in virtual communities. This underlines the basic problem even more: the lack of attributes that are inherently required in group collaboration, and led to authorization difficulties (AuthZ). One of our finding is that *profile attributes* that characterize the end-user and *group attributes* that are defined within a virtual organization are both required and need to be supported consistently. The X.509 type authorization is getting obsolete and or unpopular in the environments where cloud-based services are accessed via Web. It also turned out that most developed NREN federations have introduced their own attribute authority solution sometimes not based on SAML.

c) The HEXAA user survey was a web based questionnaire. HEXAA MS 2.1 Annex B contains the HEXAA survey questions and HEXAA MS 2.1 Annex C is a summary of the answers. The responses arrived from PhD holders in a big proportion, so it is assumed that research and collaboration environments were known by them. Responses arrived from 23 countries from different research fields.

d) The article "*The HEXAA Survey on SAML External Attribute Providers*" was sent to be published in ICIC Express Letters on the 30th of March 2015.



2.1. Figure Distribution of responses by highest degree by country (Q6)



2.2. Figure Distribution of scientific areas

The responses were helpful to estimate the average user knowledge about federations, privacy, their requirements and preferences.

e) Assessed the technical challenges

Attribute Consent

In HEXAA we use the standard SAML Attribute Query for retrieving attributes. The attributes that HEXAA handles relate to entitlements, group information or profile information. The SAML Attribute Query is done via SOAP that the SP initiates and HEXAA answers. Therefore, this communication does not involve the user's browser.

The user data cannot be released without user consent. In the case of an IdP, a request for consent screen is usually presented to the user when the first attribute release happens. Technically this is not an issue because the authentication process happens in the user's browser and this process can be paused for an interim user consent step. In the hub and spoke or proxy architecture an IdP can act as an attribute aggregator. This attribute aggregation happens in the background but it is still connected to the browser session. Therefore a consent screen also can be presented that covers all kinds of attributes from all sources.

In our case however, there is no possibility to request the user interaction as the SP communicates directly with HEXAA via SOAP. As a consequence, no attributes will be released unless the user visits HEXAA web interface first and gives consent there.

Not releasing attributes might break the use case in question, so the user should be notified of the requirement. One possible way to solve this would be to present an appropriate message to the user either by the SP software or by the application itself. However, the SP software are not able to do this at the moment, and delegating this task to the application would break the functional decomposition of the system. Simply, it is not the task of the application to take care of the consent. Moreover, if there are multiple EAP-s in the system, the application does not have a way to know where the user is managed so it won't be able to show direct link to the appropriate EAP web interface.

Therefore, the best way to request for consent is when a user accepts the invitation to a group or signs up to a group using a web interface. This is how we implemented HEXAA.

EAP Discovery

The EAP Discovery problem is similar to the IdP discovery problem in some respects. It is generated by the fact that there could be multiple EAP-s in a federation all of which can in theory contain the user's data. Therefore, either all the EAP services have to be contacted or there has to be a way in which the user selects its own EAP.

However, there are major issues with the EAP Discovery concept. One issue is that while the user always has one IdP in a session, it is entirely possible that the rest of the additional attributes have to be aggregated from many different EAP-s. As a consequence, the choice of EAP is a multiple choice.

Another issue is that an additional interactive discovery step besides the IdP discovery would be highly confusing for the users.

Finally, currently there is no technical solution whatsoever to instruct the SP software which EAP-s should be contacted for a certain user in a certain session.

Technically the SP would be able to present an EAP discovery screen for the user as a part of the authentication process. Naturally, this would require significant modifications of the SP code. While we think this could be done, we do not think that this is the right approach because an additional discovery step would be too confusing for the end users.

This means that currently all the EAP-s have to be contacted upon every session initiation. There might be use cases in which the EAP-s that respond with actual data could be remembered, though. But currently that is not supported either by the SP-s.

During the span of HEXAA project we did not implement any kind of EAP Discovery service because of the problems above. Instead, the SP contacts all EAP-s that are known it. We have conducted experiments on this, that show that this scheme does not scale well beyond 3-5 AA-s. Normally this number is enough; however, in the long run it will be necessary to develop EAP discovery protocols and implementations. This will involve standardization work, and is out of scope of the current HEXAA project.

Attribute Aggregation and Attribute Conflict

In a federated system an SP will know more than one EAP. A technical issue, besides the number of EAP-s (see the previous section) is how the attributes are aggregated and merged. If there is an attribute “A” coming from EAP1 and an attribute “B” from EAP2 then the attribute set provided for the application will contain both. Therefore the attribute set will be a simple union of all the attributes. The origin of the attributes will not be communicated to the application which probably would not process this information anyway. However, if attribute A comes from multiple EAP-s, or one from the IdP and one from an EAP then there is a potential conflict of attributes. If the values of “A” are the same then it is clear that only one instance should be kept. However, if the attribute values are different, then one approach would be creating a multi-valued attribute that contains both values. This means however that the applications should be advised that every attribute is potentially multi-valued, even those that are not naturally multi-valued (e.g. Surname). Another approach would be to drop one value following some priority list.

All this would require additional software development if the SP is a Shibboleth SP, however. In the current HEXAA project we could only document what the default behavior in attribute conflict situation: Shibboleth will always merge the attributes and will provide a multi-valued attribute, while simpleSAMLphp is configurable to either keep one attribute (using a given priority) or merge.

Level of Assurance

The problem of the missing technical solution to communicate the Level of Assurance is not specific to EAP-s. However, the presence of one or more EAP-s besides the IdP-s makes this problem more pressing. The term can refer to the strength of the authentication and also to the assurance of the validity of attribute values. As HEXAA does not authenticate, LoA in this case is a claim about the attribute values.

HEXAA is able to store LoA information for every attribute value, even though the LoA levels are not defined at the moment and can be different in each setup. Also, on the API level this information has to be made available, regardless the fact that the SAML AA component cannot relay this information in its current state because there is no standard way of communicating attribute LoA in SAML. Moreover, in HEXAA it is possible to define LoA levels. LOA levels and corresponding control measures can be defined freely in HEXAA. Default LOA level of attributes in HEXAA is 0, which indicates that no control measures are implemented to ensure the accuracy of attributes (e.g.: attribute is provided or confirmed only by the user);

f) The categorizations of use-cases in which the attributes are used help conceptually and technically address the attribute problems in federations.

Scope – organization related	Target community
Local	university / research institute
NREN federation	eduID /HU/
Inter-federation	eduGAIN*
Scientific federations	EGI science gateways

2.1. Table Hierarchical classification of uses cases with examples

Scope – platform related	Target application area
Content management	Liferay, MediaWiki, Drupal
Cloud	OpenNebula, OpenStack, CloudStack,etc.
All other applications	Icinga, AjaxPlorer, EduJabber, RackTables, etc.

2.2. Table Platform related classification of use cases

This classification clarifies two fundamental questions:

- what is the scope of collaboration (local, national, etc.)?
- what type of applications (services) need to be provided for the collaboration?

Attribute authorities have to be able to support the two faces of requirements with attributes to ease collaboration.

2.3 Observations

It has been observed by many parties that Identity Providers (IdP) are not able or not willing to support environments with attributes that is outside of their management domain. This roots in the flexible demands of the “outside” environment opposing to the pragmatic, procedure driven and well controlled internal rules of organizations. The interpretation of the Code of Conduct is not fully consistent within the European NRENs, debates are quite frequent about definitions of attributes especially covering different organizations or countries. This is why the contribution of attribute authorities is essential.

During the lifetime of the project the Horizon 2020 calls for proposals have been published. One of the long awaited activity addresses the collaboration field especially inter-federations. This is considered a main step-forward that hopefully will help the consolidation of collaboration technologies.

There is a new challenge of integrating non-SAML based authentication methods into federations, including attributes that are coming from additional sources.

2.4 Recommendations

i. Within the research community the TERENA Code of Conduct is a key document concerning the legal and technical requirements. This document needs to be revised to be more applicable if attribute authorities are used in federations.
ii. Users’ knowledge about federations, virtual organizations, attributes, etc. are rather limited. Their privacy concern is well defined. Continuous work is required to help communities to understand their technical options and there is a need for a support framework to help communities to establish virtual organizations for their scientific and research need.
iii. Comprehensive standardization work is required on the field of Level of Assurance/Confidence of attributes in order to be able to use these unambiguously in a wide range of environments.
iv. Interworking of Attribute Authorities is anticipated for inter federation purposes but the way of operation, set of requirements are not defined.
v. Attribute Authority discovery is foreseen requirement but its status is unknown in the research community in spite of the fact that the use of it is unavoidable.

3 Developer, administrator and end-user documentation

This is most detailed documentation of our work in the HEXAA project. The three subsections separately deal with the three aspects of the results in the area of:

- the HEXAA software developer
- the HEXAA administrator
- the HEXAA user

It is unavoidable that the information is very technical due to nature of the project. This also gives opportunity to the reader to skip it or go deeper in the accompanying Milestone documents of MS 4.2.1; MS 4.2.2, MS 4.2.3. Even the three documents refer to external documentation that is available for the API function calls of HEXAA that are generated automatically from the back-end source code.

3.1 HEXAA software developer documentation

3.1.1 Approach

HEXAA is a flexible web-based application that realizes a SAML-based External Attribute Authority. The Software consists of a “Core” part that is implemented in the Symfony PHP framework. The “Core” provides an API that serves both the web GUI that is implemented in AngularJS and the SAML AA endpoint which is implemented by simpleSAMLphp.

HEXAA benefits from in-house experience of previous VO management software implementations. The design of HEXAA does not rely solely on the use cases we were supported in the past. Instead, in the framework of a requirement analysis work package we explored various use technical cases by relying on literature, a survey and interviews as well. Legal considerations were taken seriously, and become part of the requirements.

3.1.2 Results

Architecture

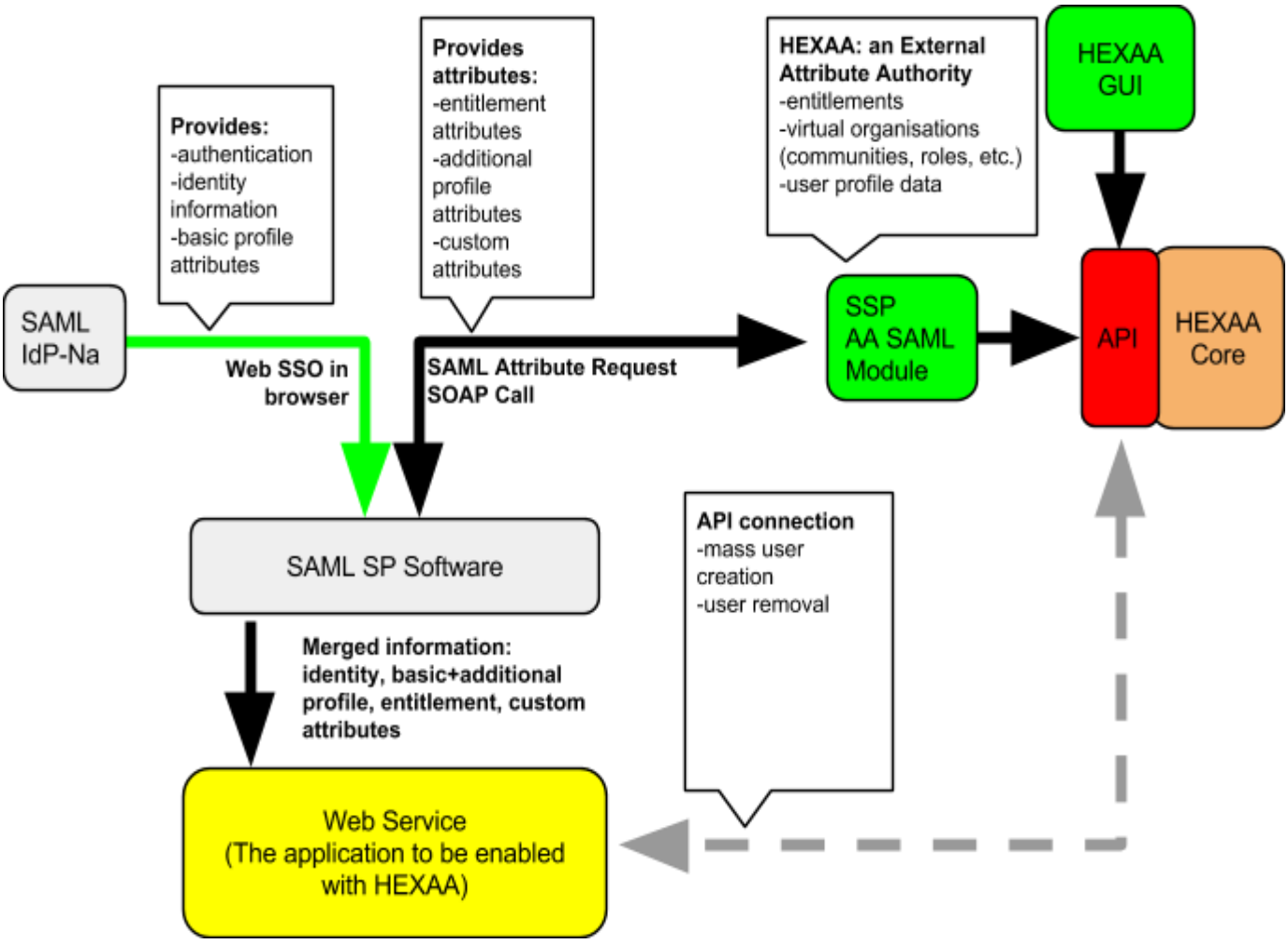
HEXAA's core is a php application written in Symfony2. The HEXAA core provides a REST API to its web interface (written in AngularJS) and for its SAML Attribute Authority module. The architecture is represented by the following diagram.

The HEXAA Core (backend) is a standalone Symfony2 application. The backend was written using the Symfony2 REST edition (https://github.com/gimler/Symfony-rest-edition) to provide a consistent, powerful and scalable REST interface. A key feature of HEXAA is that every action can (and must) be done using its REST API. This makes the backend highly suitable for integration with other systems.

The backend application is divided into two Bundles: ApiBundle, which contains all the controllers and actions and StorageBundle which contains the Entities, Forms and Validators.

This separation is required to keep the REST specific configuration to ApiBundle, while allowing various other tasks to be coded in StorageBundle.

In most of the CRUD actions Symfony2 Forms are used to process, validate and store incoming calls. Various components of the rest-edition are used, that help handling of JSON formatted data. XML is enabled in the HEXAA configuration, but it is experimental as JSON is the recommended format in our system.

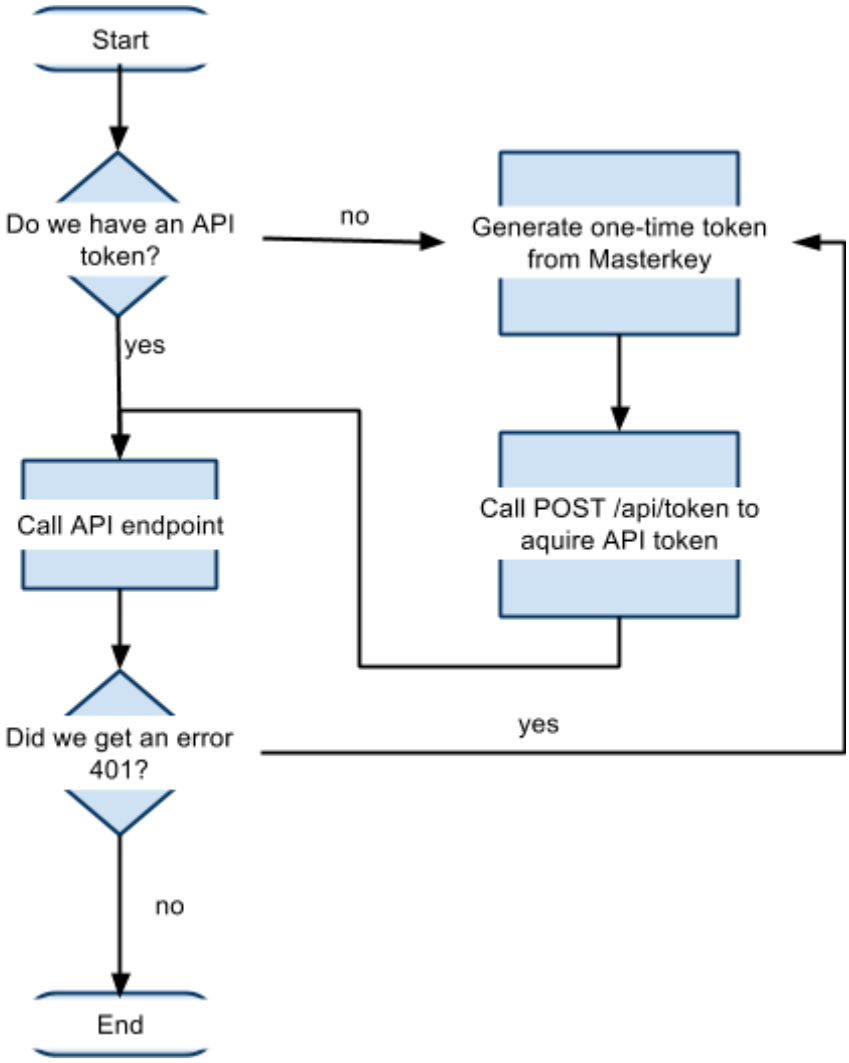


3.1. Figure HEXAA architecture

Authentication in HEXAA

Most endpoints require our custom header attribute (X-HEXAA-AUTH) to be set with a valid token as a value for authentication. This token can be acquired by calling POST /api/token with another one-time token generated from the system *masterkey*.

The auth process is described in the following flow chart:



3.2. Figure Authorization process in HEXAA

Entities

The following table describes the main entities in HEXAA.

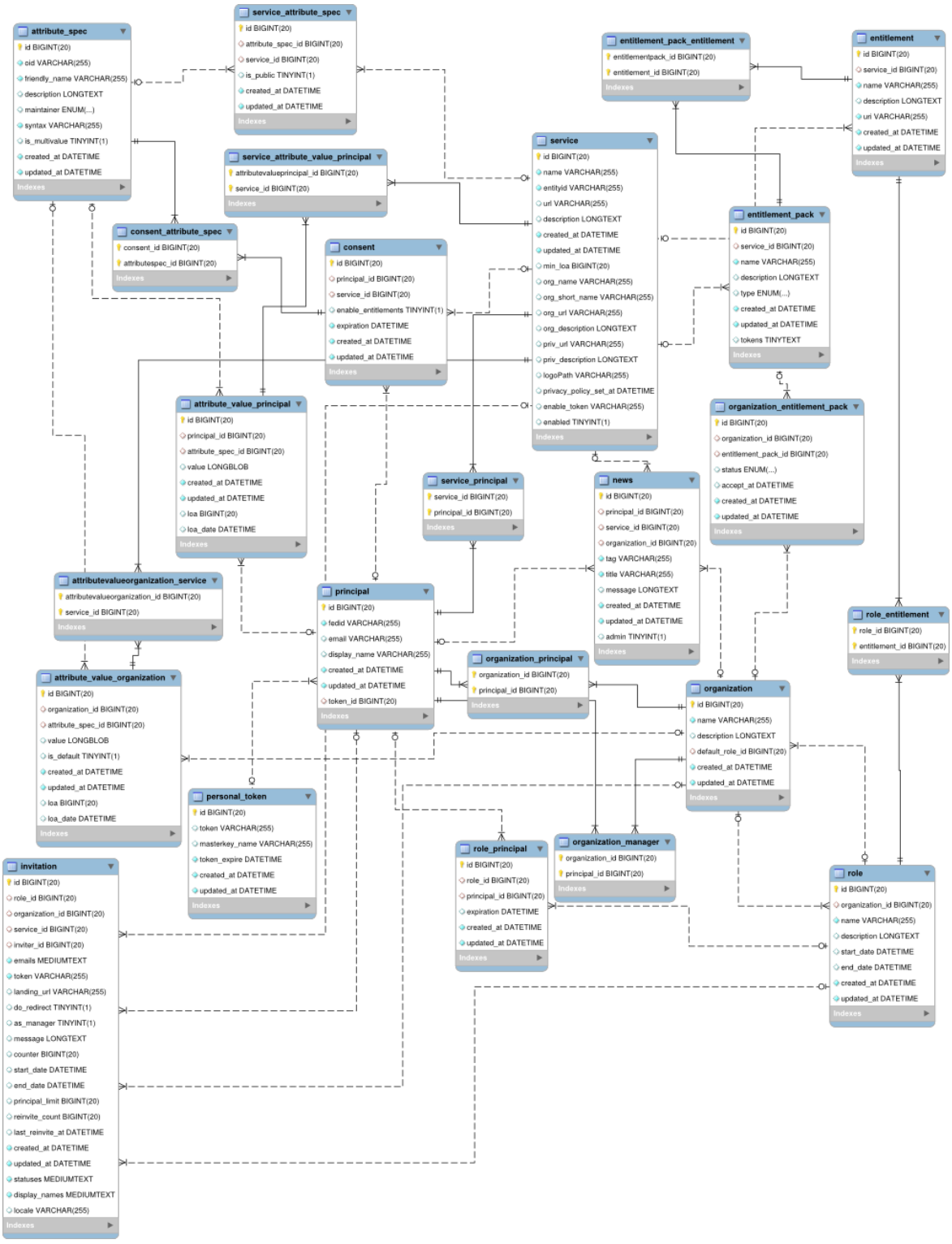
Entity	Description
AttributeSpec	Stores Attribute specifications. Only HEXAA admins may create and modify them, to avoid overflowing of attributes.
AttributeValueOrganization	Represents an attribute value of an organization. The entity stores the AttributeSpec of which the value is created.
AttributeValuePrincipal	Represents an attribute value of a principal. The entity stores the AttributeSpec of which the value is created.
Consent	An instance of the Consent entity stores the enabled AttributeSpec instances of a service-user combination. Attributes from the stored AttributeSpecs will be released to the stored Service.
EntitlementPack	A package of Entitlements. Every EntitlementPack is owned by exactly one Service and may only contain the Entitlements of the owner Service.
Entitlement	Owned a Service, an entitlement is translated into eduPersonEntitlement at attribute release. Read more here: https://wiki.surfnet.nl/display/surfconextdev/Standardized+values+for+eduPersonEntitlement
Invitation	An invitation may be created by any Organization or Service manager. Only a HEXAA admin might invite people into Services or Organizations not managed by him/herself. May contain e-mails, but as tokens are used to access the invitation instances, URL (mass) invitations are possible, too.
News	A News entry is created for every action. These may be linked to a principal, a service or an organization. Various filtering options are available for the queries of News objects.

Entity	Description
OrganizationEntitlementPack	Represents a connection between a Service and an Organization. There are two ways an OrganizationEntitlementPack entity can be created: an Organization requests a public EntitlementPack (which then needs to be approved by the Service manager), or the Service manager shares a one-time token for a private EntitlementPack with the Organization manager, who uses it, to link the EntitlementPack to his/her Organization.
Organization	A virtual Organization (VO). The members of the VO automatically inherit the VO's attributes. Organization managers may sort members into Roles.
PersonalToken	Always assigned to a Principal, a PersonalToken is required to access almost all API actions.
Principal	Represents a user. Has a PersonalToken linked to it.
Role	Owned by exactly one Organization, Roles make Entitlement-Principal assignment possible. The owner Organization's managers may assign Entitlements and Principals to Roles. Supports timed Role (de)activation
RolePrincipal	A connector Entity between Roles and Principals. This Entity stores the expiration of a Role-Principal assignment.
ServiceAttributeSpec	A connector Entity between Services and AttributeSpecs, which stores the type (public/private) of the Service-AttributeSpec assignment. Public AttributeSpecs appear to all users as a possible attribute-type to enter values to, while private linked AttributeSpecs only appear to users, who are members of an Organization linked to the Service.
Service	Represents a HEXAA-user application (a SAML SP in most cases). Stores information not only about the application itself, but about the organization managing the app.

3.1. Table Entities in HEXAA

3.1.2.1 *Database*

HEXAA relies on Symfony2's default, Doctrine ORM, so a variety of database solutions are supported. The structure of the HEXAA database is detailed on the following diagram. Although generated by Doctrine, this diagram helps understanding connections between Entities better.



3.3. Figure HEXAA database schema

A scalable and well readable image is available here: https://hexaa.eduid.hu/landing/hexaadb.svg

3.1.3 Observations

We collected observations about the feasibility and viability of HEXAA in two ways: by relying on a large number of unit and integration tests and by using the software in production.

To ensure the correct operation of HEXAA and its API, a testing environment was created with test cases for each possible call. The environment was written in Java (requires version 1.8 or later), using the Apache HTTP Client. The tests were created in JUnit (requires 4.1 or later).

We are relying on a class called CoverageChecker that lists the untested API calls. This way we were able to keep up with the test cases as the API evolved during development.

In our production environment there are currently hundreds of users, and many dozens of VO-s and applications. The feedback from our users is very positive.

3.1.4 Recommendations

There are a number of recommendations we can make based on our experiences with HEXAA development and operation in production, here only the two most important ones are included.

i. The implemented entitlement solution in HEXAA is flexible enough to cover a very wide variety of use cases where simpler group-based authorization wouldn't be enough.
ii. Our approach to rely on the SAML Attribute Query SOAP calls proved to be very fruitful as both simpleSAMLphp and Shibboleth SP-s are able to make these calls now (the SSP implementation was part of this project). This way the application integration can be very light weight and for the application it does not matter whether the authoritative attributes are coming from the IdP or from a separate External Attribute Provider.

3.2 HEXAA Attribute Authority administrator documentation

3.2.1 Introduction

This section sums up the most important information that is required for the HEXAA administrator.

3.2.1.1 *Overview of HEXAA components*

HEXAA consists of three main components:

- the backend (API)

- the frontend (GUI)
- and the Attribute Authority

3.2.1.2 *Preparing for installation*

HEXAA depends on the presence of the following features on the server:

- webserver (Apache)
- PHP (CLI binaries are required)
- connection to an SQL database (ie. mysql-client)
- Shibboleth SP (>=2.0) for the GUI

This guide will not go into details about how to configure and operate the software above; you must use the corresponding documentation of the tools instead.

In addition, the following tools are used for a normal install:

- *git*
- *curl*
- *composer*

Out of the above, composer should be installed from its upstream:

```
curl -sS https://getcomposer.org/installer |php [--install-dir=/path/to/dir]
```

The installation steps are detailed at the sections describing each HEXAA component.

3.2.1.3 *Source build*

In addition to the normal installation, the following tools must be available for doing a source build of the GUI component. This is the recommended approach however, because it enables you to upgrade the Symfony components independently.

- *nodejs*, including the following utilities:
  - *npm*
  - *bower*
  - *grunt-cli*

Note that the *nodejs* package in debian wheezy does not contain *npm*, therefore the recommended approach on Debian is to install the *nodejs* package from the *nodesource* repositories, which can be set up by the following command:

```
curl -sL https://deb.nodesource.com/setup | sudo bash -
```

3.2.2 HEXAA Backend

The purpose of the Backend is to provide an API to the User Interfaces, particularly to the HEXAA GUI.

3.2.2.1 *Installation*

```
composer create-project --no-dev hexaa/hexaa-backend /opt/hexaa dev-master

composer.phar install --no-dev
```

Note that this guide will assume that your HEXAA root directory is `/opt/hexaa`, which is also your working directory.

Fix any missing requirements that are reported by composer. Add write permissions for the webserver to the `app/cache` and `app/logs` directories and the HEXAA log directory (`/var/log/hexaa` by default).

```
chgrp www-data app/cache app/logs /var/log/hexaa

chmod 775 app/cache app/logs /var/log/hexaa
```

3.2.2.2 *Configuration*

The main HEXAA configuration file is `app/config/parameters.yml`. It is recommended to copy `app/config/parameters.yml.dist` for first time configuration. You should configure the parameters for the database connection first. (The configuration options are self-describing.)

After you have created the database on the database server, create the tables for the application with the following command:

```
php app/console doctrine:schema:update
```

You can configure the mail delivery options with the `mailer_*` parameters. Since the mail handling relies entirely on Symfony, you can find the description of the configuration options on the Symfony website.

Other configuration options from `app/config/parameters.yml`:

- *`locale`*: default user interface language. Currently the available options are `en` and `hu`.
- *`secret`*: the secret salt used for hashing miscellaneous data, such as tokens.
- *`hexaa_ui_url`*: the 'main' HEXAA GUI URL. For some operations like invitation, HEXAA Backend gives the user callback links (such as token verification). This parameter is used to construct these URLs.
- *`hexaa_log_dir`*: the location where HEXAA stores its log files. Note that the webserver must be able write to this directory.

- *hexaa_master_secrets*: this is a list of secretKey -> keyName pairs. It allows different GUIs to use the services of the API with different keys. It is also possible to assign access control rules to different GUIs, see below for details.
- *hexaa_consent_module*: you can globally enable or disable the consent module for HEXAA. If you enable the module, attributes are released to the service provider only if the user agrees on the attribute release. Since attribute exchange is a back-channel operation, the consent must be given before the SP retrieves the available attributes.
- hexaa_entitlement_uri_prefix: a URN prefix that is assigned to this HEXAA instance. The entitlement values are dynamically generated by the software, thus it is very important that the prefix must be properly delegated, otherwise an *eduPersonEntitlement* value could be misinterpreted.

3.2.2.3 *HEXAA Administrator*

HEXAA Administrator has a special role in the system. He/she has unlimited rights to manage every Organization and every Service in the system and can remove any HEXAA accounts. This feature was added to simplify user support.

In addition to managing Organizations and Services, HEXAA administrator can use the GUI for managing attribute specifications, see the next section. The list of the federated identifiers (usually the *eduPersonPrincipalName*-s) of the HEXAA administrators can be specified as a yaml list in `app/config/hexaa_admins.yml`. After modifying this file, the Symfony cache must be cleared:

```
sudo -u www-data php app/console cache:clear --env=prod
```

3.2.3 HEXAA GUI

HEXAA GUI is an AngularJS based frontend for the HEXAA Backend.

3.2.3.1 *Installation*

Download the web application from the following Git repository:

https://github.com/hexaaproject/hexaa-gui.git

For using the pre-compiled scripts, copy the contents of the `dist` directory to your web folder. If you want to build the scripts yourself, go to the root of the HEXAA GUI source and execute the following commands:

```
npm install
bower install
grunt build
```

3.2.3.2 *Configuration*

The configuration of the GUI is in the `config.php` file:

- **$hexaa_base_address** is the root URL of the HEXAA installation
- **$hexaa_api_address** is the URL of HEXAA installation's API
- **$hexaa_cookie_name** is the name of the cookie where the token is stored. This should be unique.
- **$hexaa_master_secret** The master secret of HEXAA. You can find it under `your_hexaa_install_dir/app/config/parameters.yaml`
- **$hexaa_logout_url** HEXAA UI will redirect you to this page on logout. It Should be the Shibboleth SingleLogout Endpoint.
- **$hexaa_env_eppn** Server attribute name of the federal unique ID in alignment with your Shibboleth deployment.
- **$hexaa_env_mail** Server attribute name of the federal unique mail. Compare it with your Shibboleth Installation.
- **$hexaa_dont_check_ssl_certificate**: Set it to true if your installation does not use HTTPS protocol.

3.2.4 Attribute Authority

The Attribute Authority part of HEXAA is implemented by the SimpleSAMLphp AA module, which should be configured with a special HEXAA authentication processing filter.

The AA module and its SP counterpart the Attribute Aggregator module has been accepted by the SimpleSAMLphp maintainers as a standard module (see https://simplesamlphp.org/modules), therefore their documentation is not included here, except for what is specific to a HEXAA deployment.

3.2.4.1 *Authproc filters*

Configure the AA module (`config.php` in your SimpleSAML configuration) to use the HEXAA authentication processing filter to retrieve attributes from the HEXAA Backend:

```
authproc.aa = array(

     ...

     '60' => array(

       'class' => 'hexaa:Hexaa',
       'nameId_attribute_name' =>  'subject_nameid', // look at the aa authsource config

       'hexaa_api_url' => 'https://www.hexaa.example.com/app.php/api',

       'hexaa_master_secret' => 'you_can_get_it_from_the_hexaa_administrator'
```

3.2.4.2 *Apache configuration notice*

The AA authenticates its peer SPs either by the signature of the SAML request or by relying on the TLS channel. The latter is the default with Shibboleth SPs, therefore it is recommended to run the AA in a dedicated port (8443 as an example, don't forget to add it to `ports.conf`!), that can be accessed with X.509 authentication. The webserver on this port is not meant to be accessible for end users.

Note that if you run the HEXAA GUI and the AA on the same host, you most probably want the following Apache directives to be different:

- *ServerName*: due to an undocumented Apache feature, the VirtualHost configuration of more than one SSL-enabled webservers must use different ServerNames. The recommended way is to append the port number to the ServerName.
- *certificate*: user accessible pages should use well-known CAs, on the other hand, for federational entities the use of self-signed certificates is recommended.

An example configuration file snippet:

```
<VirtualHost *:8443>
  ServerName hexaa.example.com:8443
  ServerAdmin admin@example.com
  SSLOptions +StdEnvVars +ExportCertData
  SSLVerifyClient optional_no_ca

  Alias /aa /usr/share/simplesamlphp/www/
```

3.2.5 Managing relying parties in HEXAA

3.2.5.1 *Metadata*

In order to let HEXAA know anything about a Service Provider, the SP's entityID and contact information must be listed in `$HEXAA_BACKEND/app/config/hexaa_entityids.yml` file. You can manage this file by hand, or alternatively you can use the following XSL to generate the YAML file from the SAML2 Metadata (XML) of a federation:

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
           xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
           version="2.0">
<xsl:output method="text"/>

<xsl:template match="/">
   <xsl:apply-templates/>
</xsl:template>

<xsl:template match="/md:EntitiesDescriptor">
   <xsl:apply-templates/>
</xsl:template>
```

```
<xsl:template match="md:EntityDescriptor[md:SPSSODescriptor]">
   <xsl:value-of select="@entityID"/>
   <xsl:text>:&#10;</xsl:text>
   <xsl:apply-templates select="md:ContactPerson"/>
</xsl:template>

<xsl:template match="md:ContactPerson">
   <xsl:text>  - type: </xsl:text>
   <xsl:value-of select="@contactType"/>
   <xsl:text>&#10;</xsl:text>

   <xsl:text>    email: </xsl:text>
   <xsl:value-of select="substring-after(md:EmailAddress,':')"/>
   <xsl:text>&#10;</xsl:text>

   <xsl:text>    surName: </xsl:text>
   <xsl:value-of select="md:SurName"/>
   <xsl:text>&#10;</xsl:text>

</xsl:template>

<xsl:template match="*"/>

<xsl:template match="text()"/>

</xsl:stylesheet>
```

Note that you can apply the XSL by using an XSL processor tool like *xalan*. There are legitimate reasons for which you might want HEXAA to use different SP contact addresses from what is published in the metadata, however, in this case you must maintain the entityID list manually.

3.2.5.2 *Service registration*

In HEXAA every service must have at least one associated administrator account. For registering a service, an administrator must be invited via an e-mail that is sent to one of the contact addresses. The individual who accepts the 'invitation' must be authenticated to HEXAA.

3.3 HEXAA Attribute Authority end user documentation

The end user documentation cannot be discussed separately of Section 3.1, especially the supporting MS 4.2.1 and MS 4.2.3 documents that sums up the use cases from the end user perspective.

3.3.1 Approach

3.3.1.1 *User environment requirements*

The HEXAA user interface uses HTTP protocol to communicate with the end-user browser. The following web browsers were used:

- Internet Explorer of Microsoft Corporation version 11.0
- Firefox version 32.0 or newer
- Chrome 38.0

The operating system we used:

- Microsoft Windows 7 or newer
- Linux versions not older than 2012 for the browsers

The HEXAA user interface is available on smaller screens – like tablets or phones – however the recommended minimum resolution is 1280 x 1024 pixels.

Any user of HEXAA has to have a federated identifier and an IdP that allows her/him to login.

We are not aware of any dependency on plugin tools of the above browsers.

The HEXAA reference installation is available at: http://hexaa.eduid.hu

3.3.2 Results

Four basic levels of authorization are introduced in HEXAA.

1. *Ordinary user*: he is the simple end user who can be members of several virtual organizations (VO)-s. (We will refer to the Ordinary user as “user”.)
2. *Virtual Organization manager*: any *Ordinary user* can be a Virtual Organization manager if he/she creates a VO. (In the future we refer as VO manager this role.) This document mainly focuses on the VO manager.
3. *Service manager*: the person who has special rights over a *Service*, he/she is not necessarily member of any VO. In case the *Service manager* does not have a federated id, his role can be taken over by the *HEXAA administrator*.
4. *HEXAA administrator:* This is the highest level of authorization within HEXAA, this is the “super” admin, who usually has access not only via the user interface with special rights, but also to the operating system that runs HEXAA as a “root” user.

The above categorization is only first dimension of authorization levels. A second dimension is given that allows context dependent authorization of the above four basic categories. This leads to the concept of AA as a Service capability of HEXAA. In practice it means that even the “HEXAA administrator” role is contextualized

resulting in a high level of administration authority in a given context – like activities of entities that linked to an institution. It also allows greater flexibility to connect services for a specific domain.

HEXAA web interface is written in AngularJS a well-known JavaScript framework. As with the all the other software deliverables it is available as a Apache Open Source piece of code from GitHub: https://github.com/hexaaproject

The HEXAA web interface is fully in English and includes a help facility.

The most novel feature of the HEXAA user interface is: the Virtual Organization wizard that helps the inexperienced user to create, populate and refine the requirements a new virtual organization. The Virtual Organization wizard works in offline mode as well.

3.3.3 Observations

The HEXAA graphical user interface (GUI) is a critical element as it has to accommodate requirements from two sides. The GUI builds on function of the API of HEXAA backend. The GUI is written in a high-level scripting language. If layout, functionality and/or requirements of HEXAA change the update of the user interface is inevitable. There can be a pressure from the user side to modify the user-interface in a certain way, should this requirement affect HEXAA backend its implication are bigger.

In an ideal world we could have enough time to develop automatic UI test scripts, unfortunately this was not possible. (The HEXAA backend is systematically tested for errors in an independent environment.)

HEXAA as proof of concept was demonstrated at early November 2014. Since then HEXAA is also in the eduID.hu federation. The code is on GitHub, feedback on the package is expected.

Conceptually, it is not difficult to comprehend the categories, terms and workflows that are implemented within HEXAA, but the guidance of more experienced administrators is anticipated.

3.3.4 Recommendations

The HEXAA Graphical User Interface is written in AngularJs the backend is coded in Symphony. Our recommendation is that as these two toolsets work very well together other GEANT projects should consider it as a serious implementation option.

For planning purposes the GUI specifications, layout designs and requirements need to be complete before the actual implementation starts.

At least two levels of testing of the GUI is recommended one for the basic virtual organization level managers (simple end users) and another one for the end-users solving more complex tasks.

Training courses for HEXAA end-users, a Moodle course for HEXAA and video demonstrations are also needed.

4 Guidelines for integrating Attribute Authorities in federations and eduGAIN

It is clear from the previous sections that two equally important set of requirements need to be fulfilled when federations apply attribute authority solutions. This section sums up the technical and legal recommendations.

4.1 Technical guideline

4.1.1 Introduction

This document contains recommendations for integrating Attribute Authorities (such as HEXAA service) to a national federation or to inter-federations such as eduGAIN. The document focuses on Higher Education and Research use cases, which may not fit for other kinds of attribute authorities operated by governments or for business.

4.1.2 Attribute Authority models

Group or Virtual Organization management platforms, or as they are sometimes called, collaboration platforms can be implemented by using different federation models. Based on the preliminary results of the (yet unpublished) collaboration platform survey (http://bit.ly/aa-overview), external attribute sources are integrated to federations in two different ways:

- as a **trusted third party**, being queried by the service provider and the attributes are aggregated at the SP;
- as a **proxy**, that plays SP role to the users' IdP and an IdP role to the SP. In this case, attribute aggregation happens at the proxy.

Both models have their advantages and disadvantages.

4.1.2.1 *Trusted third party model*

If the AA is a third party, the request flow will be the following:

1. the user makes a request to the SP;
2. the user is sent to her IdP (possibly after IdP discovery);
3. after authentication the user is redirected to the SP with some attributes from the IdP;
4. the SP makes a request to the AA to fetch additional attributes.

Steps 1-3 are a normal federated authentication flow, while Step 4 is usually a back-channel request.

Its biggest advantage is that it does not affect the normal trust relationships, thus the SP trusts the IdP and the attribute authority based on two independent trust decisions. It is also possible to collect attributes from several attribute authorities, although it does not scale well.

Common SAML middleware (Shibboleth and SimpleSAMLphp) come with support of attribute aggregation, however, it is possible to implement Step 4 by using other protocols as well (such as VOOT).

Its main limitations are:

- both the SP and the AA needs to share a common identifier of the user, thus it is not possible to use targeted identifiers only;
- Step 4 happens without interacting with the user, therefore her consent on the attribute exchange must be given by using a different workflow. This also means that it is possible to query the users' attributes at any time, which is undesirable from the privacy point of view although it can be useful for provisioning purposes.

Attribute authorities can be published to eduGAIN independently of its services, therefore one AA might serve service providers from multiple federations.

4.1.2.2 *Proxy model*

Proxy model is similar to how hub and spoke federations work.

1. the SP redirects the user to the Proxy;
2. the user is sent to her IdP (possibly after IdP discovery) by the Proxy;
3. after authentication the user is redirected to the Proxy with some attributes from the IdP;
4. the Proxy adds additional attributes to the response while redirecting to the initiating SP.

The advantage of this model is that the attribute aggregation happens only at the Proxy. It's also possible to interrupt the flow with attribute consent screens and the attributes are only passed to the SP during session initiation (no off-session attribute collection is possible). Hub and scope federations can integrate additional attribute sources, while their IdPs and SPs don't need to be reconfigured.

However, the proxy model changes the normal federated trust relationships between the IdP and the SP because the two entities do not interact with each other directly. The proxy is a `Big Brother' in a way that it processes all of the attributes of all users. The SP can only be integrated to one proxy at a time and the integration fundamentally changes its federated properties (entityID as an example). It is not possible to implement fine-grained attribute policies at the Identity Provider, because the attribute requirements of the Proxy is the union of the requirements of its SPs. End to end security cannot be provided by the middleware layer (or the Proxy must be viewed as a part of the middleware layer).

Currently, proxies are either operated by the national federation or by a group/project. The proxy itself can be published to eduGAIN easily, which enables the services to use inter-federated IdPs, however, the tight relationship between the Proxy and the SP does not allow the integration of SPs from other federations.

HEXAA

Evaluating all the above, the HEXAA project has decided to use the trusted third party model as the preferred way to implement HEXAA services. However, by configuring its underlying middleware, it is also possible to build a proxy service based on HEXAA software.

4.1.3 Metadata considerations

In general attribute authorities consist of two components that are represented in two different SAML entities. The first one is the management user interface, that is used for inviting people to groups, managing attributes and assigning services to groups, VOs or users. The second one is the component used for producing attributes for relying service providers.

4.1.4 User interface

The management interface is usually implemented as a SAML2 Service Provider (SP) entity. The metadata of the SP does not really differ from any other Service Provider, therefore normal recommendations apply, such as:

- Saml2Int Profile;
- EduGAIN Metadata Profile

It is recommended for the SP to apply techniques that facilitate scalable attribute exchange, thus inform the users' Identity Providers about its attribute requirements and privacy policies. However, the privacy policy must cover the ultimate attribute authority use case, that is to supply information to relying parties. Attribute scalability techniques may include but are not limited to:

- use of RequestedAttribute metadata elements;
- use of eduGAIN Data Protection Code of Conduct; however its f) clause requires that the Attribute Authority must only release *the attributes that has been obtained from the Identity* Provider to relying (third) parties if:
    - the relying party is also committed to the Code of Conduct; or
    - if prior consent has been given by the End User.
- use of REFEDs Research and Scholarship Entity Category if applicable. Note that this also requires the use of some metadata attributes that are defined in the Metadata Extensions for Login and Discovery User Interface (MDUI) specification.

4.1.4.1 *Attribute service*

The metadata of the attribute producer interface depends on the model in which the attribute authority works (see the section about Attribute Authority models above).

Assuming that the service is implemented as a SAML2 AttributeAuthority, only the following information needs to be present in the metadata for both national and inter-federated use:

- *entityID*: the management interface and the attribute service might share a single SAML entity, thus the *SPSSODescriptor* and the *AttributeAuthorityDescriptor* might be in the same *EntityDescriptor*. By this it is possible to avoid redundant information in the metadata (*RegistrationInfo*, *Organization* and *ContactPerson* as examples). However, there is very little experience on how applications handle such entities. In the cost of redundancy, the two interfaces can be in two different entities with different entityIDs.
- *credentials*: the certificate used in federation context.
- *AttributeService* URL, the web service to contact for attributes.

4.1.4.2 *Certificates*

There are three certificates associated with the attribute authority service:

- the management user interface must use a certificate that is known to the users browsers;
- the SP protecting the user interface must use a certificate that is allowed by the federation (e.g. a self-signed);
- the attribute service must use a certificate that is known to the SPs.

If the attribute service is authenticated by the relying SPs by using TLS (and not by the signature of the *Response* or the *Assertion*) then the webserver must be configured to use the AA certificate. This kind of configuration is recommended, because Shibboleth and SimpleSAMLphp attribute aggregation uses TLS authentication by default. This requires the attribute service to be located on either a different IP address or a different port.

4.1.4.3 *Interpretation of RequestedAttribute metadata element*

The SAML metadata of Service Providers might contain *RequestedAttribute* elements, by which the SP can signal what attributes it needs. Many Identity Provider deployments rely on this information for constructing their attribute release policies with the underlying assumption that the requirement is justified by the federation or during manual metadata exchange.

During the project it was discussed whether it was possible to use the same element as a basis for constructing attribute release policies in attribute authorities as well. The outcome of the discussion was no because there is no way for the SP to indicate that the attribute is required from the IdP or from the AA, and driving SPs to extend their *RequestedAttribute* usage might have undesired results on IdP attribute release as well.

Therefore HEXAA does not interpret the *RequestedAttribute* element and requires the service provider administrators to manage the attribute requirements of their services in the HEXAA management interface. This is recommended to other third party authorities as well.

4.2 Legal guideline

4.2.1 Policy Recommendations

4.2.1.1 *Need for setting organizational policies for managing attributes with the help of EAPs in HE identity federations*

IdPs are able to help the authentication of users by providing attributes to SPs that are needed for the proper authentication of the users. Additionally, IdPs are also able to provide additional attributes from a technical point of view that are not needed for the proper authentication of users, but needed for the authorization of users' access to different SP resources. However, the experience of the past years shows that the provision of authorization related attributes is not scalable, neither from a technical nor from a privacy and data protection point of view.[2]

Therefore the provision of attributes that are needed for authorization purposes (e.g.: virtual organization membership status) can be managed in the longer run or above a certain number of users or attributes more efficiently only with the help of EAPs.

However, the presence of EAPs in HE identity federations create specific challenges both from technical and personal data protection point of view. Therefore, HE identity federations should develop specific policies for the operation of EAPs within the federation. This is in line with the practice of the grid federations.

4.2.1.2 *Organizational policy recommendations for the introduction and use of EAPs in HE identity federations*

The following recommendations are addressed to HE identity federations from an organizational point of view in order to help the compliancy of EAPs with privacy and data protection requirements.

R1. HE identity federations shall be aware of the possibility of the presence and use of EAPs*.*

R2. HE identity federations shall develop specific application procedure for EAPs*.*

R3. HE identity federations shall develop and publish policy requirements for EAPs *that are harmonized at inter federation level.*

[2] From a privacy and data protection point of view the attribute provision is not scalable at IdPs, because accuracy of attributes can only be maintained in the longer run only at those organizations, where the maintenance of the particular attributes is part of their everyday function. Taking into account of the various needs of SPs, it is unlikely that a single organization that is operating the IdP can ensure the accuracy of all types of attributes without encountering serious organizational or financial difficulties.

4.2.1.3 *Policy recommendations for the operation of EAPs*

- The following recommendations are applicable to all type - trusted third party and proxy model - of EAPs.

R1. EAP *shall develop and maintain controlled attribute management in order to ensure that attributes of users are*

a. collected and stored only in case the data subject unambiguously consented to it;
b. accurate and kept up to date;
c. erased or rectified if data is inaccurate or incomplete;
d. erased in case the processing of the attributes is no longer needed;

I.1 Implementation of this requirement in HEXAA:

- HEXAA provides information to the users about the processing of their attributes;
- HEXAA provides the possibility for users to consent to the collection and storage of their attributes;
- In case attributes are not provided to HEXAA by the data subject HEXAA requests the users to confirm his consent to the processing of attributes;
- HEXAA periodically requests users to confirm the accuracy of the stored attributes;
- HEXAA provides the possibility for users to erase or block the provision of their attributes;
- HEXAA erases the attributes after a preset period unless the user request the continued storage of the attributes;

R2. EAP *shall provide information to SPs or IdPs about the level of assurance (LOA) of the accuracy of the attributes. LOA must indicate the level of control measures that aim to ensure the accuracy of the attributes;*

I.2 Implementation of this requirement in HEXAA:

- HEXAA stores LOA value of the stored attributes;
- LOA levels and corresponding control measures can be defined freely in HEXAA;
- default LOA level of attributes in HEXAA is 0, which indicates that no control measures are implemented to ensure the accuracy of attributes (e.g.: attribute is provided or confirmed only by the user);

R3. EAP *shall release attributes to SPs only in case the data subject unambiguously consented to the release of the attributes.* EAP *shall not release attributes in case the SP does not provide information about the purpose of the processing of the released attributes to the data subject.*

I.3 Implementation of this requirement in HEXAA:

- HEXAA stores references to the privacy policies of SPs that contain the purpose of the processing of the requested attributes;
- HEXAA requires the data subjects to consent to the processing of the attributes by the SPs before the release of the attributes;
- HEXAA stores information about the fact of consenting to the release of the attributes with regard to purposes and SPs or separately;

R4. EAP *shall provide the possibility for the data subject to prohibit the release of his attributes to certain SPs or revoke his previously given consent to the release of his attributes.*

I.4 Implementation of this requirement in HEXAA:

- HEXAA provides to the data subjects the possibility to explore, review and withdraw his previously given consents;
- HEXAA informs SPs upon the withdrawal of the consent to the release of the attributes;

R5. EAP *shall provide information to the user upon the user's request when attributes of the user were released to different SPs.*

I.5 Implementation of this requirement in HEXAA:

- HEXAA stores information about the release of attributes to different SPs;
- HEXAA provides to the user the possibility to explore the information about the release of his attributes;

4.2.2 Personal data protection requirements applicable to EAPs

4.2.2.1 *Overview of issues relating to the processing of personal data in HE identity federations*

EAP and personal data

Attributes provided by EAPs fall into the category of personal data as it is defined by the data protection directive. According to the directive, personal data is any information relating to an identified or identifiable natural person. In identity federation context where attribute provision is closely connected to the identification of users it is not a question whether users are identifiable or not, since IdPs main role is to authenticate the users and EAPs only provide additional attributes (personal data) of the authenticated users, even in case only the user's nickname or other pseudonym is provided, since users even in this case can be identified with the help of IdPs. Therefore in case of IdP authenticated users we find directly identified or indirectly identifiable users, whose personal data is provided by EAPs to SPs. Therefore, the legal requirements of personal data protection always apply to the operation of EAPs within identity federations.

Legal requirements relating to the processing of personal data, EAPs as data controllers

Although the data protection directive is currently under revision[3] and most likely it will be replaced by a new regulation in the near future, it is possible to outline the main requirements relating to processing and use of personal data on the basis of the directive, since basic principles of personal data protection will not change.

The data protection directive defines requirements concerning the processing of personal data and obliges those who process personal data to adhere to these requirements. According to the directive the responsibility for the fulfilment of data processing requirements lies with the data controller, who can be any natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data.

Since EAP collect and store attributes fully or partly independently of SPs, the organizations that operate the EAP are data controllers themselves on their own right or jointly with SPs, because EAP define the purpose of attribute collection and storage independently from or jointly with SPs.

Although, there may be situations where EAP act only and exclusively on the basis of the instructions of SPs and thus not deciding on the purpose of attribute collection and storage, but even in these situations it may be questionable whether EAPs really acts just according to the instructions of SPs, especially in cases the EAPs are serving more than one SP with the same attributes.

Therefore it is more useful to consider EAPs as independent actors and design their operation in line with the data protection requirements that are applicable to data controllers, than prepare just for the fulfilment of those requirements that are applicable to data processors.

Although there are numerous data protection requirements of the data protection directive[4] concerning the processing of attributes by EAPs as data controllers and the national (EU member state level) implementations of the data protection directive also add an additional layer of complexity to these requirements, it is possible to highlight a few basic requirements, without which it is not possible to fulfil data protection requirements at all. These requirements are:

a) ensuring the legal basis of processing of attributes
b) defining the purpose and time of attribute processing
c) maintaining validity and accuracy of attributes
d) collecting and providing information about the processing of attributes and deletion of attributes

[3] eIDAS – Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

[4] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG

Ensuring the legal basis of processing of attributes

The data protection directive states that personal data must be processed lawfully. This means that EAPs as data controllers need legal entitlement for the processing of user attributes. The legal entitlement can be provided by regulation at the level of laws or by the user himself in the form of his consent.

In relation to the consent of the data subject the directive requires that the consent to be a freely given specific and informed indication of the data subject's wishes by which the data subject signifies his agreement to personal data relating to him being processed. This means that EAPs need to provide sufficient information to the users about the processing of their attributes once requesting their consent to the processing of attributes. The consent must be specific, a vague, superficial or broad descriptions of the potential use of attributes cannot be sufficient. As a minimum the expected purpose, period and important conditions of the collection, storage and provision of attributes needs to be described.

Defining the purpose and time of attribute processing

The data protection directive states that personal data must be collected for specified, explicit and legitimate purposes and not further processed in a way incompatible with those original purposes. This means that user attributes cannot be collected, stored and provided to third parties by EAPs without defining the purpose of these activities.

Additionally the purpose must be clearly communicated, in most case in written form to the users. In practice typically privacy policies contain the description of the purpose. However, short summaries (e.g.: short privacy notices on websites) are also very often used to inform the data subjects about the purpose of the processing of personal data, especially in situations where the data controller requests the consent of the data subject to the processing of his data.

Time or period of personal data processing is an inherent element of defining the purpose of personal data processing. In most cases it is impossible to judge the validity or lawfulness of the purpose of data processing without knowing the foreseeable period of personal data processing. E.g.: The lawfulness of storing user's address by an EAP for the purpose of keeping correspondence with him cannot be judged without knowing how long the address will be kept since the user can stop using the service without the knowledge of the EAP that requires the storage of the user's address. However, it is allowed not to set a specific date or period, but in this case it needs to be such a purpose that really justifies the potentially endless storage of personal data. (e.g.: Storing the result of medical treatments in case of illnesses that cannot be cured.) However, it is unlikely that EAPs will be able to justify the endless storage of user attributes. Thus foreseeable timing of attribute storage needs to be defined by EAPs.

Maintaining validity and accuracy of attributes

The data protection directive requires processed data to be accurate and, where necessary, kept up to date. This requirement sounds simple, but compliance with this requirement is quite cumbersome and it is possible to fulfil this requirement only in case the data controller develops organizational procedures, which ensure the validation and regular update of the stored personal data.

In case of EAPs this means that ideally EAPs should develop and maintain organizational procedures and design their systems in a way that supports the validation and update of user attributes. These technical and organizational measures together can constitute a *controlled attribute management*, which ensures the

fulfilment of the simple requirement of keeping data accurate and up to date. It is worth to note that the use of EAPs for attribute provision instead of IdPs is basically triggered by the fact that the validity of non-authentication related user attributes usually cannot be maintained at IdPs neither from a technical nor from an organizational point of view.

Collecting and providing information about the processing of attributes and deletion of attributes

Data subjects have the right to be informed about the processing of their data. According to the data protection directive, data subjects can request confirmation as to whether or not data relating to them are being processed and information about the purposes of the processing, the categories of data concerned, and the recipients or categories of recipients to whom the data have already been disclosed.

This means in case of EAPs that information about the stored attributes and information about the release of these attributes to SPs must be provided to users upon their request. In order to be able to fulfil this requirement EAPs shall collect this type of information. However, the endless collection of information about the release of attributes to SPs might collide with the need for purposeful storage of personal data. In order to create a balance between these two competing requirements it is necessary to define a maximum storage period of the attribute release information.

Data subjects can also request the deletion or blocking of their personal data or withdraw their formerly given consent to the processing of their personal data. In case of such a request, data controllers also have to inform third parties to whom data was released.

In case of EAPs this means that EAPs need to inform SP that the user has requested the deletion or blocking of his data. Ideally, in order to fulfil the above requirement EAPs shall also provide support to SPs concerning user de-provisioning.

4.2.2.2 *Other specific questions relating to the processing of attributes by EAPs*

When and how exactly user consent is required?

In European countries legality of personal data processing can be based on the data subject's consent or on regulatory authorization at the level of laws.

In case of data processing by EAPs it is unlikely that we can find regulatory authorizations therefore, personal data processing (collection, storage, etc., of attributes) by EAPs can only be based on the consent of the data subject unless regulation specifically authorizes this possibility.

In case of data processing by SPs it is more likely that we can find regulatory authorizations in some countries in some specific domains. However, the transfer of attributes to SPs from EAPs cannot be based on these authorizations, because data protection authorizations on the side of SPs do not authorize EAPs to transfer attributes to SPs unless the regulation also specifically authorizes this transfer as well. Therefore, attribute transfer from EAPs to SPs can only be based on the consent of the data subjects unless regulation specifically authorizes the data transfer.

Use of pseudonyms in EAPs

It is possible that EAPs do not transfer the real name of the data subjects to SPs, but the pseudonym of the data subject beside other attributes. In this case the same data protection requirements apply, since all other attributes that are provided by EAPs are still personal data, because EAPs collect, store and provide attributes of identified persons, since EAPs provide attributes of persons who were previously identified by IdPs.

5 HEXAA project self-assessment

In this section the Open Call project HEXAA is self-assessed by first comparing it to the original Open Call text and the wider Open Call scope. The following two subsections contain a copy of requirements from the original documentation of Open Call proposal 15, and that of the HEXAA response to requirements 15.

In section 5.1 we compare HEXAA results and the Open Call planned requirements.

In section 5.2 we compare our response to the requirements of Project 15 and our results.

The HEXAA project team eventually invested much more efforts than it was originally planned and the results have surpassed the anticipated expectations.

5.1 Comparison with the objectives of the Open Call text

Open Call text of "Building Support for External Attribute Authorities in Higher Education Federations" (p37-p38)	HEXAA results and approach in terms of the Open Call request for proposal
15.2 Objectives This topic seeks proposals that analyze possible ways to implement attribute authorities and evaluate their strengths and weaknesses. The following aspects must be considered: • Differing technical implementations, including the underlying technology. • The impact on existing HE Identity Federations: how will the regulatory framework apply to the attributes provided via an attribute authority? And what will be the implications of namespaces on attributes? A proof-of-concept must be built for the preferred model, with an explanation of why this particular model has been selected.	**15.2 HEXAA objectives** In the first phase of the HEXAA project the background was carefully analyzed in order to assess the problems, requirements and solutions that hinder the work of the research community to effectively work in federated environments. A set of interviews were carried out to find the root of the attribute problem. In MS 2.1 the analyses of the results were discussed, including a user survey. The legal environment was carefully analyzed and in MS 3.1 and MS 3.2 the legal requirements were assessed and a proposed attribute authority operation requirement set was proposed. Standardization of the attribute namespace was not in the scope of the project, but the developed technical solution is flexible in handling different attribute namespaces. A proof of concept demo was built and the justification of the selected way of operation is described in MS 2.1. All the objectives of the Open Call proposal were met.

Open Call text of "Building Support for External Attribute Authorities in Higher Education Federations" (p37-p38)	HEXAA results and approach in terms of the Open Call request for proposal
15.3 Expected Impact The results are expected to ease the deployment of HE Identity Federation. There are known use cases, such as e-Science, that would be benefit from this approach. This topic is intended to broaden research horizons in the GN3plus project and seek cooperation with new partners. The project provides an opportunity for external groups to submit a bid and to work to develop a solution in collaboration with existing Identity Federations. This solution, which will be built on open standards, can then be reused by different groups with similar needs. The results of this work will be disseminated to the relevant communities.	**15.3 HEXAA Impact** The HEXAA project has developed an open source software package that is available on https://github.com/hexaaproject. This is extended with a set of tools that ease the most common environments with accessing attributes from HEXAA or from other attribute authority. e-Science integration was developed at the very early stage of the project for a very widespread e-Science gateway the gUSE/WS-Pgrade. HEXAA project has identified a set of problems for research, development and even for the legal framework. There has been a close collaboration with JRA3 T1 (groups and attributes) and REFEDS during the project lifetime and also with other attribute authority provides like PERUN, Unity and OpenConext. Several presentations were held about HEXAA in the last 18 months, see Appendix. The HEXAA software is fully integrated with the Hungarian Identity Federation (eudID) and as the code is open-source it is possible to reuse it in other environments. The expected impacts of the project were met.

Open Call text of "Building Support for External Attribute Authorities in Higher Education Federations" (p37-p38)	HEXAA results and approach in terms of the Open Call request for proposal
15.4 Outputs The outputs are expected to be: 1. A report that includes analysis (including strengths and weaknesses) of possible ways for HE Identity Federations and eduGAIN to support external attribute authorities. Feedback on these findings will be provided by the JRA3 Identity & Trust Technologies for GÉANT Services as well as by the Research and Education Federations (REFEDS) group. 1. Guidelines for HE federations to support third-party managed attributes authorities, based on the selected model(s). These should cover both the technical aspect (namespaces, protocols and so on) and the legal aspects, particularly concerning the data protection issues. 2. One (or more) proof-of-concept for one of the proposed model(s). The proof-of-concept should be demonstrated first with a couple of national HE Identity Federations and later with eduGAIN. The outputs of this work will be used by GN3plus Joint Research Activity 3 Identity & Trust Technologies for GÉANT Services, specifically by Task 1 (Attributes and Groups).	**15.4 HEXAA outputs** MS 2.1 is the technical and organizational analyses document and MS 3.1 and MS 3.2 are the legal study documents. All three studies were made available at the GEANT intranet and also JRA3 T1 circulated the documents for comments, and was presented at the REFEDS meetings (Zürich, Dublin). There are two documents that address how the services of attribute authority can be used MS 4.3 and DS 4.1 A proof of concept demo is available at http://hexaa.eduid.hu/demo including twenty screenshot videos. The HEXAA software is now part of the services of the eduID Hungarian federations. The collaboration with JRA3 is excellent and in the future it will be expanded in GN4 Y1. The required output of the HEXAA project surpasses the expected output formulated in the Open Call.

5.2 Comparison with the original project proposal

Open Call proposal of HEXAA (p6-p7)	HEXAA results and approach in terms of the original Open Call proposal
Project objectives 1. Creating a comprehensive study on the federation-related requirements for Attribute Authorities using a representative set of academic institutions and individuals. 2. Creating a well-founded study on the possible architectures for an Attribute Authority Service. 3. Exploration of the policy and legal aspects of Attribute Authorities and establishing guidelines for federations and individual institutions those want to introduce Attribute Authorities. 4. Implementing good quality, open source software to meet the requirements of the identified and important use cases for Attribute Authorities. 5. Dissemination of the results by papers, talks, targeted communication to Academics.	**HEXAA response** 1. - 2. The MS 2.1 document analyses the current problem space of attribute authorities in identity federations. The results of the interviews with representatives of the research communities were taken into account when the document was prepared. Results of the user survey can also be found of the MS2.1 document. The assessment of the technical issues was also surveyed, and used as a basis for the development. 3. MS 3.1 and MS 3.2 analyses the legal background and formulates a well-defined set of legal requirements for the identity Federations and also for the Attribute Authority operator 4. The HEXAA software code is presented in DS 4.2 and this is an open-source code, using very standard open source development tools and environments that make it possible to deploy in other environments. DS 4.1 document addresses the integration of e-Science gateways and HEXAA. MS 4.3 is the guideline of how to integrate the HEXAA attribute authority to eduGAIN or other federations. 5. There were several dissemination activities. This is detailed in Append A.3. The HEXAA project was carried out strictly along the original project objectives. All the original objectives were met or in some areas it was even surpassed.

Open Call proposal of HEXAA (p6-p7)	HEXAA results and approach in terms of the original Open Call proposal
Progress Beyond the State of the Art Our proposal does not involve the invention of any grand new technology as it will be rather a novel combination of existing technologies. The proposal contains a study on the related state of the art systems, and also the development of a new system that will go beyond the state of the art because: • It will be based on comprehensive requirements survey focusing on e-Science and on the needs of academics. • It will be supplemented by a profound research in the legal and policy context, carried out by legal experts in personal data handling in regulatory and contractual environment. • The developed proof of concept system will have proper and detailed documentation and it will be systematically disseminated to the target group.	**HEXAA Progress Beyond the State of the Art** • The survey got wide publicity and good feedback. The journal paper that is part of the dissemination activities summarize the results in scientific terms, while the MS 2.1 gives a good overview of the survey questions and responses with analyses. • MS 3.1 and MS 3.2 detail our findings in the legal field. Our findings in the legal aspects of the attribute authorities and federations are a really important one and I think that our fresh approach to the subject is a really important contribution to the field. • MS 4.3 is a 100+ pages documentation of the three components of the developed software. The proof of software demo video set is available online. The HEXAA backend documentation is also online. The HEXAA graphical user interface has got an English help system. The proof of the pudding is in the eating. Hungarnet/NIIFI included its service portfolio HEXAA and this secures sustainability. Other NRENs, communities or institutions might also choose HEXAA as an attribute authority and HEXAA is going to be offer services for eduGAIN too.

6 Conclusions

1) The demand for attributes and groups high and recent meetings confirm that in the future this requirement will be with us unless the problem is properly addressed. (Internet Society InterFed+Attributes meeting in September 2-4, 2014, Utrecht, The Netherlands, or FIM4R meeting at GENEVA/CERN: https://indico.cern.ch/event/358127/),
2) HEXAA successfully proved that:
   a) there is a solid requirement of using external attribute authorities from the higher education and research community
   b) a green field approach to provision attributes is more future proof than tinkering obsolete or legacy software to achieve much smaller impact
   c) to reach full state-of-the art compliance of legal requirements for attribute authorities is possible.
3) HEXAA achieved integration with eduID.hu in order to introduce VO as a Service for the Hungarian federation.
4) HEXAA is currently supporting the High-Performance Computing portal of NIIFI/Hungarnet via its API interface, showing a way how legacy applications/environments can be integrated.
5) The graphical user interface of HEXAA allows on-demand formations of virtual organizations and setting authorization for individuals and also keeping the user in control for attribute release.
6) There are numerous plugins that were developed for HEXAA; the OpenStack plugin and the e-Science gateway plugin will have the biggest impact.
7) HEXAA is a project that has lots of potential for the future. There is a planned collaboration in the GN4 JRA3 Groups and Attributes activity. Due to the potentials of the HEXAA project other large scale collaboration opportunities are sought in the H2020 project proposals, including awarded proposals like AARC, MAGIC etc.
8) One key future development that needs high attention in research, development and standardization is the Attribute Authority discovery framework that is a completely missing field.
9) The legal study is a foundation of legal requirements of any Attribute Authority that is to be used within identity federations. The eduGAIN community requires a set of requirements for the AA-s, and this has to be incorporated in the eduGAIN Code of Conduct.
10) HEXAA as a software development result is unique in its functionality, architecture and services. The key areas of impacts are: for simpleSAMLphp the attribute aggregator and attribute authority modules; the HEXAA core support of profile attributes and virtual organization attributes, and the HEXAA API interface that opens up possibilities for supporting legacy systems.
11) The set of plugins that were developed for the HEXAA projects like OpenStack, OpenNebula, Liferay, Drupal, etc. are paving the way for future collaborations.
12) HEXAA complements the work done in GN3plus JRA3 and SA5. The adoption of the project's results will simplify several use cases identified in JRA3 during GN3plus and in GN4.

Appendix A Background references

A.1 Referenced milestone/deliverable documents

Id	Description	Reference
MS 2.1	Analysis on handling user attributes within federations	GN3plus_HEXAA_Milestone_2.1-final
MS 3.1	Legal and Policy Study Document	GN3plus_HEXAA_Milestone_3.1-v1
MS 3.2	Attribute Authorities requirements	GN3plus_HEXAA_Milestone_3.2-v1
MS 3.3	HEXAA personal data protection assessment	
MS 4.1	Proof of Concept	https://hexaa.eduid.hu/
MS 4.2	Developer, administrator, end user software documentation	GN3plus_HEXAA_M4.2.1 Software Developer Documentation GN3plus_HEXAA_M4.2.2 Administrator Documentation GN3plus_HEXAA_M4.2.3 End_User_Documentation
MS 4.3	Guideline for integrating attribute authorities in federations and eduGAIN	GN3plus_Milestone_Hexaa_M4.3_v1
DS 4.1	e-Science application integration and portal engines demos	HEXAA e-Science application integration and portal engines demos
DS 4.2	Final version of software	https://github.com/hexaaproject
MS 5.1	Website established	https://sites.google.com/a/sztaki.hu/hexaa

A.1. Table HEXAA document references

A.2 Source code references

Id	Description	Reference
1	This is the final software deliverable root of HEXAA, the GUI and the backend is available from here.	https://github.com/hexaaproject
2	These are SimpleSamlPHP attribute authority code for HEXAA.	https://github.com/NIIF/simplesamlphp-module-aa https://github.com/NIIF/simplesamlphp-module-hexaa
3	HEXAA plugin for e-Science gateways integration based on Liferay.	https://github.com/mheder/liferay-shibboleth-plugin
4	OpenNebula cloud plugin for HEXAA	https://github.com/burgosz/opennebula-sunstone-shib
5	OpenStack cloud plugin for HEXAA	https://github.com/burgosz/openstack-horizon-shibboleth
6	HEXAA plugin for Drupal	https://www.drupal.org/project/shib_auth
7	HEXAA plugin for pydio	https://github.com/burgosz/pydio-shibboleth

A.2. Table HEXAA source code references

A.3 References to dissemination documents

- EMC2 Presentation 2014 - February 2014 /Kristóf Bajnok, NIIF/
- REFEDS Presentation 2014 - May 2014 /Kristóf Bajnok, NIIF /
- Terena Networking Conference 2014 poster /Mihály Héder, MTA SzTAKI/
- EGI Community forum presentation 2014 - May 2014 /István Tétényi, MTA SzTAKI /
- EGI-GEANT workshop presentation – September 2014 /Mihály Héder, MTA SzTAKI /
- GEANT project symposium – February 2015 - /István Tétényi, MTA SzTAKI /
- Virtuális szervezetek SAML föderációban (in Hungarian) – April 2014 - /Gyula Szabó, NIIF /
- A HEXAA kutatási project (in Hungarian) – April 2014 - /Zsuzsánna Magyar, MTA SzTAKI/
- Hatékony Kollaboráció (in Hungarian) – November 2014 – /Kristóf Bajnok, NIIF /
- HEXAA@eduID (in Hungarian) – November 2014 - /István Tétényi, MTA SzTAKI /
- "The HEXAA Survey on SAML External Attribute Providers" – M.Héder, I.Tétényi /MTA SzTAKI/ to be published at *ICIC Express Letters*

References

Drupal	https://drupal.org/
Entity Category	https://refeds.terena.org/index.php/Entity_Categories
FIM4R	Federated Identity Management for Research framework
FIM4R-study	https://cdsweb.cern.ch/record/1442597/files/CERN-OPEN-2012-006.pdf
gUSE	http://en.wikipedia.org/wiki/GUSE
	http://www.internet2.edu/presentations/jtcolumbus/20040720-piPEfitters-Simar.ppt
Liferay	http://www.liferay.com/
OpenNebula	http://opennebula.org/
OpenStack	http://www.openstack.org
science gateway	https://www.xsede.org/gateways-overview
AngularJS	http://en.wikipedia.org/wiki/AngularJS
eduPersonEntitlement	https://www.internet2.edu/media/medialibrary/2013/09/04/internet2-mace-dir-eduperson-201203.html#eduPersonEntitlement
Jenkins	http://en.wikipedia.org/wiki/Jenkins_(software)
JUnit	http://en.wikipedia.org/wiki/JUnit
SAML	http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
SimpleSAMLphp	https://simplesamlphp.org/
Symfony	http://en.wikipedia.org/wiki/Symfony
VOOT	http://openvoot.org/
VOMS	http://italiangrid.github.io/voms/

Glossary

Apache	Apache HTTP server
API	Application Programming Interface
CRUD	Create, Read, Update and Delete
EAP	External Attribute Provider
GUI	Graphical user interface
HEXAA	Higher Education External Attribute Authority
JSON	JavaScript Object Notation
REST	Representational state transfer
SAML	Security Assertion Markup Language
SP	Service Provider
SSH	Secure Shell
UML	Unified Modelling Language
URI	Uniform resource identifier
VO	Virtual Organization
XML	Extensible Markup Language
IdP	identity provider
LOA	level of assurance
HE	higher education
data protection directive	DIRECTIVE 95/46/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data
GÉANT Code of Conduct	GÉANT Data Protection Code of Conduct, version 1.0, June 2013, http://www.geant.net/uri/dataprotection-code-of-conduct/v1/Pages/default.aspx, http://www.geant.net/uri/dataprotection-code-of-conduct/v1
eduGAIN metadata profile	http://services.geant.net/edugain/Resources/Documents/eduGAIN_metadata_profile_v3.doc
SAML2IntProfile	http://saml2int.org/profile/current
SimpleSAMLphp	https://simplesamlphp.org/