D2.2

Robust SPF: workflow and monitoring

Document information

Title	Robust SPF: workflow and monitoring
ID	CLARINPLUS-D2.2 (CE-2016-0809)
Author(s)	Jozef Mišutka
Responsible WP leader	Dieter Van Uytvanck
Contractual Delivery Date	2016-07-01
Actual Delivery Date	2016-06-30
Distribution	Public
Document status in workplan	Deliverable

Project information

Project name	CLARIN-PLUS
Project number	676529
Call	H2020-INFRADEV-1-2015-1
Duration	2015-09-01 – 2017-08-31
Website	www.clarin.eu
Contact address	contact-clarinplus@clarin.eu

Table of contents

1 Executive Summary

This document describes the first set of tasks and modules that have been implemented in order to make the CLARIN Service Provider Federation (SPF) more robust in terms of quality, reliability and performance.

The first goal of this deliverable was to make the interaction with CLARIN services that require authentication and authorisation more user-friendly and to offer a base line of quality that can be monitored and improved. The second goal was to improve the user and administrator experience of the CLARIN Identity Provider.

The work that has been done to accomplish these goals attracted attention from other projects and groups. One of the provided solutions has already been able to not only identify but also to help solve issues that users from several institutions were faced with when trying to connect to protected CLARIN services.

2 Introduction

CLARIN is the Common Language Resources and Technology Infrastructure project and provides easy and sustainable access for scholars in the Humanities and Social Sciences to digital language data. One of the pillars that make this possible is CLARIN Service Provider Federation (SPF) that connects CLARIN Service Providers to the majority of national federations inside the European Union. Users from institutes that are members of national federation in countries that have joined CLARIN SPF can automatically access CLARIN's protected resources and services in a secure way. Other users can register at the CLARIN Identity Provider and can access the protected resources and services via it after their membership is approved.

The first task was to re-engineer CLARIN Identity Provider exposing new authentication and authorisation mechanisms, offering user-friendly front-end and improved security. The Unity IDM[1] solution was selected because it meets most of CLARIN's requirements. However, an important part was missing and the implementation of this functionality was the main part of this task.

CLARIN Service Provider Federation is not the only inter-federation but it was the first one to systematically address several of the important problems federations are suffering from. In this deliverable, we have addressed the issue when Identity Providers do not release important attributes to a service without which the service cannot fully operate. For instance, if the data is licensed under a restrictive licence the user must be uniquely identifiable over time. However, if the Identity Provider does not release such information, the service cannot let the user download the data.

There are many different services based on many different platforms in CLARIN SPF. Moreover, it happens that the application developers have neither access to the machines where the federation enabling software resides, nor the proper knowledge to handle such situations. Firstly, we need to know that attributes have not been released properly to be able to address it on the project level in a uniform way. The CLARIN Attribute Name Aggregator framework has been developed to collect information about the authentication attempts to participating Service Providers. The aggregator provides a user interface that simplifies the notification of affected entities in a unified way and tracks the progress of on-going issues. From the underlying data, statistics can be compiled about every Service Provider and Identity Providers from federations and the report can be used to improve problematic federations.

To offer a sustainable and high quality federation, we have integrated metrics into the aforementioned Attribute Name Aggregator that clearly show the current state of the member federations and their inclusion in CLARIN SPF.

Furthermore, we have formally strengthened the collaboration with DFN-AAI federation[2] in order to make the first step into a fully automated distribution of metadata in the CLARIN SPF.

The following chapters will describe the work done in this deliverable in more detail. Chapter 3 will cover Clarin Attribute Name Aggregator framework. The re-engineering of CLARIN Identity Provider will be described in Chapter 4. The metrics and statistics used to monitor the quality of CLARIN SPF will be described in Chapter 5.

1 http://www.unity-idm.eu/
2 https://www.aai.dfn.de/

3 Attribute Name Aggregator

The attribute release problem[3] has been mentioned many times in discussions by the academic federation community. By far, the majority of general claims about this topic have been based on the feelings of a particular person rather than on solid statistics.

In this task, we built a platform that can aggregate the names of attributes that Identity Providers released to Service Providers. Together with the metadata about every Service Provider and every Identity Provider, we can decide if the attribute release has been successful or not. We rely on the fact that Service Providers in CLARIN SPF have to meet strict requirements about their metadata and policies enforced by the CLARIN assessment process[4] and by the requirement to implement the Data Protection Code of Conduct[5].

The Attribute Name Aggregator consists of two components. The first one must be installed on the Service Provider itself and the second one is the central component where the statistics are collected and are displayed.

3.1 Attribute Name Aggregator on the Service Provider

There are several software solutions that allow for federating. The most used one is Shibboleth[6]. We developed a script that can be integrated with Shibboleth by using the *sessionHook* feature. The flow of the authentication is redirected to that script before finally getting to the application operated by the service provider. This effectively means that we can read the released **attribute names** in this script and inform the central service in the background.

The workflow is shown in Figure 1.

3 https://www.clarin.eu/content/report-federated-identity-attribute-release
4 http://hdl.handle.net/1839/00-DOCS.CLARIN.EU-78
5 http://hdl.handle.net/11346/GAIU
6 https://www.shibboleth.org

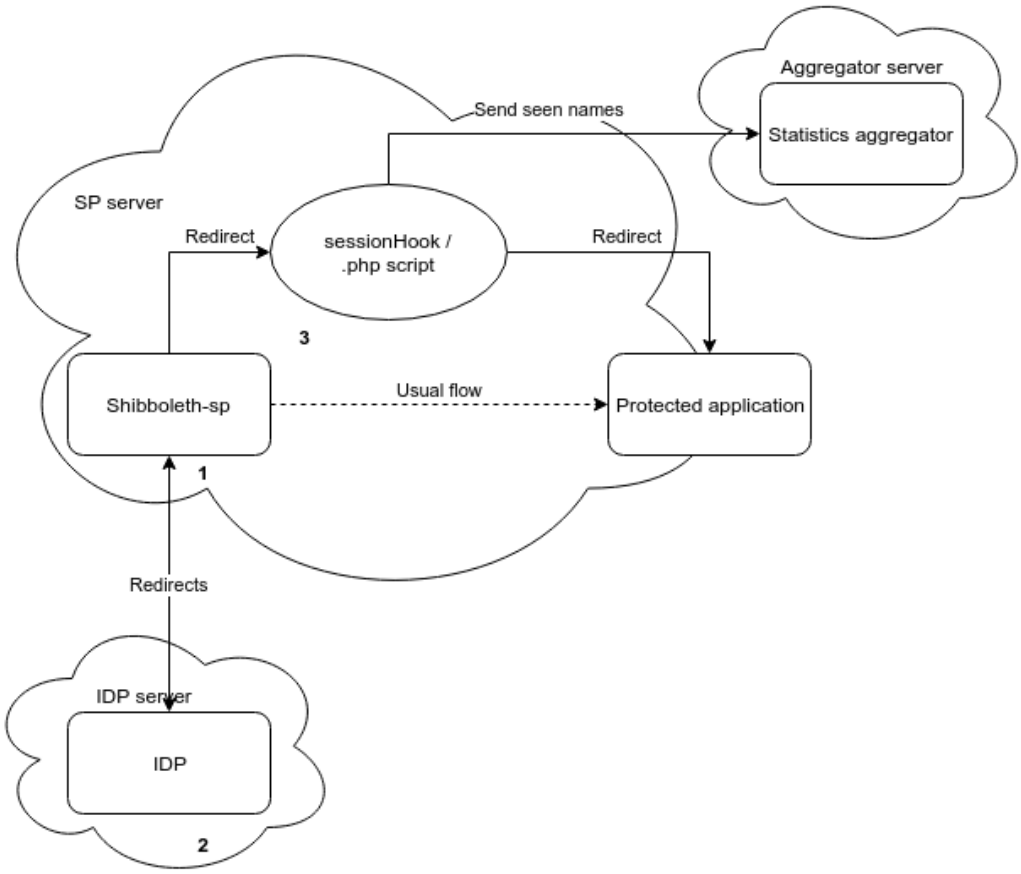


Figure 1: Authentication flow when using sessionHook functionality.

The *sessionHook* functionality[7] can be used to intercept the authentication flow and execute custom code that inspects the session using exposed Shibboleth handlers. We require that the assertion export handler defined via *exportLocation* is available but only locally; otherwise, it would introduce security risks.

The attribute name aggregator script is written in php and is executed on the server side. The script receives two parameters: *target* and *return*. The return parameter contains the location that would be used if no sessionHook was present. The target is the resource destination URL from the authentication point of view.
In order to not break the authentication execution flow, the first action of the script is to redirect the browser to the URL specified in the return parameter. This is done by sending a HTTP Location header[8]. Then, the script reads the *Shib-Assertion-Count* environment variable that contains the number of exported assertions. Then, the script reads environment variables from *Shib-Assertion-NN* (for all NN between 01 and *Shib-Assertion-Count*). The values contain URLs from which we can get the assertions. Only the **name** of the released attribute is stored from each assertion. Finally, an external program is executed to send the **attribute names** to the central service. The script does not wait for the external program to finish, minimising the performance overhead.

7 See the sessionHook description at http://hdl.handle.net/11346/SN4A
8 https://en.wikipedia.org/wiki/HTTP_location

The final step is to inform the central service using the provided REST API. Technically it ends up with a HTTPS GET request similar to

```
GET /aaggreg/v1/got?idp=https://idp.clarin.eu
&sp=https://ufal-point.mff.cuni.cz/shibboleth/eduid/sp
&timestamp=2016-06-14T11:32:21.165Z
&attributes[]=urn%3Aoid%3A2.5.4.10
&attributes[]=urn%3Aoid%3A1.3.6.1.4.1.5923.1.1.1.9
&attributes[]=urn%3Aoid%3A0.9.2342.19200300.100.1.3
&attributes[]=urn%3Aoid%3A2.16.840.1.113730.3.1.241
&attributes[]=urn%3Aoid%3A1.3.6.1.4.1.5923.1.1.1.6
&attributes[]=urn%3Aoid%3A1.3.6.1.4.1.5923.1.1.1.7
&attributes[]=urn%3Aoid%3A2.5.4.3
```

The request can be interpreted that a user tried to authenticate to LINDAT/CLARIN service provider using CLARIN Identity Provider and that several attributes have been released including e.g., urn:oid:0.9.2342.19200300.100.1.3 which is the identifier for mail. Please note, that no private information is being processed, only the **names** of the released attributes.

In case the direct integration of the aggregator is not possible, the fallback plan is to include a javascript script in the web application itself that will send the relevant information to the central service by parsing the default Shibboleth handler. The workflow of this solution is depicted in Figure 2.

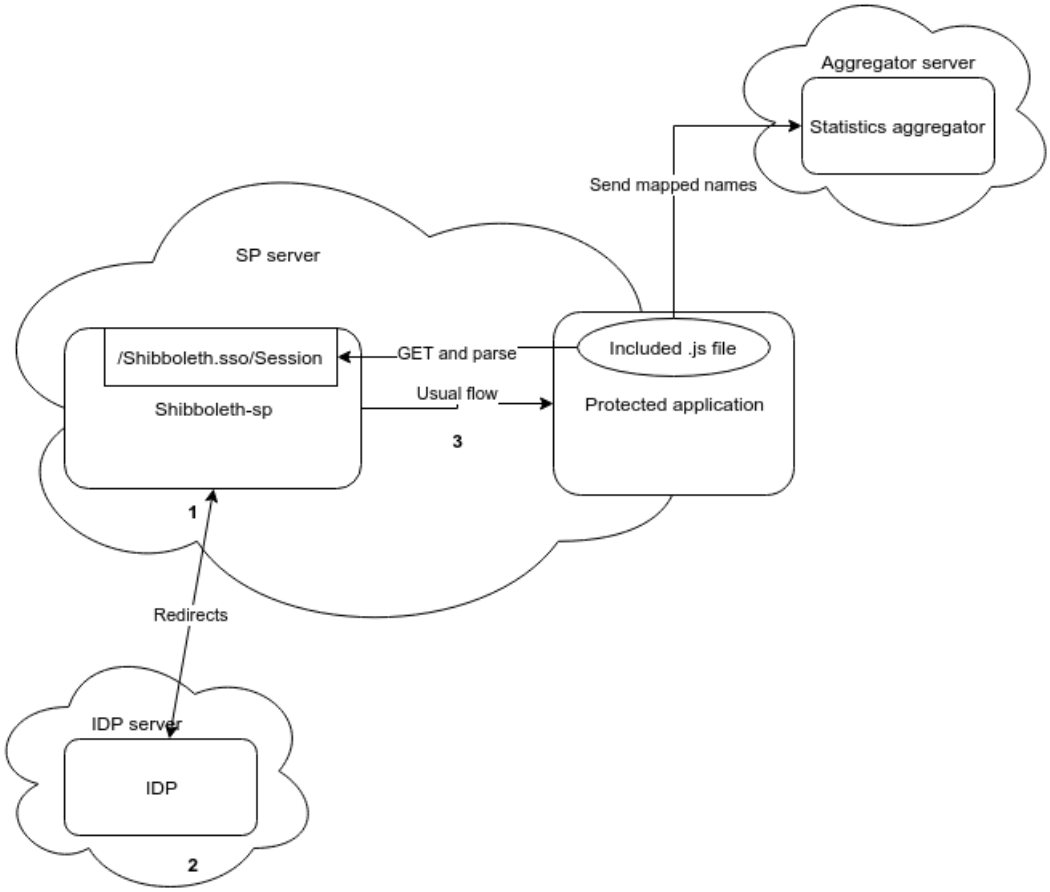


Figure 2: Javascript fallback solution work flow.

3.2 Attribute Name Aggregator as a central service

When the user is in the authenticating process and the aggregator script on the Service Provider side is executed, it sends the information to a central server. The central server is a Node.js application offering REST API to both collect and expose the attribute name aggregated information.

All authentication attempts are stored in Apache Solr search platform[9] and are indexed to enable searching. However, in order to build a usable service we need additional information about the entities. Therefore, a task is being executed regularly that downloads and processes the metadata of the following Identity Providers and Service providers:

- Service Providers from CLARIN SPF;
- Identity Providers from CLARIN SPF;
- Identity Providers from eduGAIN[10] inter-federation;
- CLARIN Identity Provider.

The metadata are parsed and important information is indexed by Apache Solr. With this information available, it is easy to decide and show to the user if the authentication attempt has met all the requirements or not. Firstly, we display the information about

9 http://lucene.apache.org/solr/
10 http://www.edugain.org/

the required attributes by the Service Provider together with the actual released attributes. We also display important entity categories (Research and Scholarship[11], Data Protection Code of Conduct) of both the Service Provider and the Identity Provider that define behaviour when it comes to attribute release.

Additional information is obtained to provide the community with unique statistics on an international scale about more than 120 institutions[12] all over the world.

For users of a SP, the CLARIN Attribute Name Aggregator can be used to transparently improve the quality of academic federating. For SP admins, the aggregator offers a "one-click send email" that sends a description of an authentication problem together with more detailed information to emails parsed from the Identity Provider and Service Provider metadata. This rather simple feature uses information from multiple different sources and makes managing and improving the quality of a feed with more than thousand Identity Providers feasible. Federation operators can use the statistics to understand how their Identity Providers are configured in the context of attribute release.

3.3 Installation and Usage

The software for the Service Provider side of the aggregator project is publicly available at https://github.com/ufal/clarin-sp-aaggregator. The Readme.md file contains a detailed description of how to deploy it to both the Shibboleth federation solution or as a web application dependency.

The central aggregator application is publicly available at https://github.com/ufal/lindat-aai-attribute-aggregator. It contains a definition file for the Vagrant project[13] that calls a setup script which contains the complete step-by-step installation instructions that are used to create a fully working virtual machine with the software installed..

11 https://refeds.org/category/research-and-scholarship
12 Information from May, 2016.
13 https://www.vagrantup.com/

4 Re-engineering CLARIN Identity Provider

In the CLARIN infrastructure, different technologies have been used to meet the AAI requirements. Services that want to utilise the CLARIN user information are forced to use the LDAP protocol. The user database is stored in a LDAP server but Drupal CMS is used for managing individual user records.

With the increasing number of users and new requirements for interoperability, this setup (especially the synchronisation between the Drupal database and the LDAP server) became both difficult to sustain and difficult to integrate new services with. Implementing new features that would make more complex authentication and authorisation workflows possible would mean to support yet another system and possibly also very complex changes to the current technology stack.

The goal of this task was to re-engineer the current state by selecting one solution containing most of the functionality out of the box and updating it to CLARIN's needs. The evaluating criteria were the following:

- user friendly administrator interface;
- user and administrator friendly self-registering functionality;
- consistent and synchronised user information across all authentication and authorisation interfaces;
- health of the project e.g., number of active developers.

Different alternatives have been evaluated (pwm, OpenIDM 3.1.0, Unity IDM 1.5.0 and other solutions that would have to be heavily customised). Unity IDM[14] was chosen because it met most of the requirements above and offered a considerable numbers of both incoming and outgoing interfaces[15]. However, services communicating with the user database using the LDAP protocol could not be easily integrated because the LDAP interface was missing. The following section describes the implementation details.

4.1 LDAP implementation

The current implementation of Unity IDM does not directly support protocols other than based on HTTP. Because LDAP protocol is not based on HTTP, we decided to expose the LDAP endpoint on a different port than the default one Unity IDM listens on. Due to this, a standalone LDAP server was needed. The second issue to address was to connect the LDAP workflow with the workflows in Unity IDM. Although not ideally maintained, the ApacheDS[16] project was used as the standalone LDAP server and its interceptors[17] mechanism was used to inject calls to Unity IDM API. At the moment, interceptors that intercept the following LDAP methods are used:

- lookup;
- search;
- bind;
- unbind;
- compare.

The parameters that are passed to these functions and the session state are used to make appropriate calls to Unity IDM.

14 http://www.unity-idm.eu/
15 http://hdl.handle.net/11346/RCTT
16 http://directory.apache.org/apacheds/
17 http://directory.apache.org/apacheds/advanced-user-guide.html

4.2 Migrating the old user database

In order to be able to seamlessly switch to the new identity management system Unity IDM, the old user database records must be migrated. Because of the security concerns, all information except the passwords is migrated. Because Unity IDM does not offer a direct way to import the records from a LDAP server, we used the REST API provided by Unity IDM to accomplish this task.

4.3 Installation and Usage

The feature implementation has been published back to the Unity IDM open source project and can be found at https://app.assembla.com/spaces/unity-public/git/source/ldapEndpoint?type=branch. The installation of the software is the same as the default version with a minor exception of introducing new variables that need to be defined for particular installation; nevertheless, default values are provided. The installation manual can be found at http://hdl.handle.net/11346/RCTT in section Installation and operation manual.

The LDIF import to Unity IdM software is publicly available at https://github.com/kosarko/unity-rest

5 Gathering federation metrics

The CLARIN SPF consists of Service Providers and Identity Providers. The Identity Providers are taken from the national federations that are CLARIN SPF members. Every national federation can have different registration practices and rules.
On one hand, CLARIN can directly impose requirements on the Service Providers during its assessment process. This is one of the mechanisms to ensure high quality and conformance to the newest standards. But on the other hand, CLARIN can neither directly affect Identity Providers nor member federations.
Throughout the years, CLARIN has been involved in different activities to improve the quality of user experience when using federated access. As the first step, CLARIN initiated the creation of a baseline quality assurance tool[18] whose idea has been lately adopted by the eduGAIN inter-federation[19]. This tool can identify problems when a user wants to authenticate via a home institution but is prevented to get to the Login page of her institution.
Another step in the quality pursuit is to provide general statistics that would use specific metrics to measure the quality of federations and this is the goal of this task. Specifically for CLARIN SPF, we also want to cross-validate if the counts in this federation are correct.
The CLARIN Attribute Name Aggregator project has been extended with additional information to provide the required information.

5.1 Identity Provider Statistics in CLARIN Service Provider Federation

We collect information about the counts of federations, Identity Providers per federation, number of Identity Providers in the CLARIN SPF and in another inter-federation as a cross-check.
The total number of Identity Providers in a national federation is yet to be included because we found out that the statistics we harvest, provided by the REFEDS metadata explorer tool[20], are not applicable. The number of Identity Providers in the CLARIN SPF is calculated from its published metadata feed. The other inter-federation – eduGAIN – also publishes a metadata feed, which is used to get the numbers for this inter-federation. These three, rather simple, numbers provide very useful insight whether the federation parsing works as expected. For a CLARIN SPF member, the number of Identity Providers in CLARIN Service Provider Federation should be the same as the number of Identity Providers in the federation itself and the number should be less or equal to the number of Identity Providers in eduGAIN. Please note, that the real numbers may vary if the feed has been updated depending on the time when they are harvested.
The statistics are exposed at a public URL[21] which can be used as a source for monitoring.

The process of obtaining the statistics is shown in Figure 3.

18 https://github.com/ufal/lindat-aai-shibbie
19 https://technical.edugain.org/eccs/
20 https://met.refeds.org
21 http://hdl.handle.net/11346/RXXN

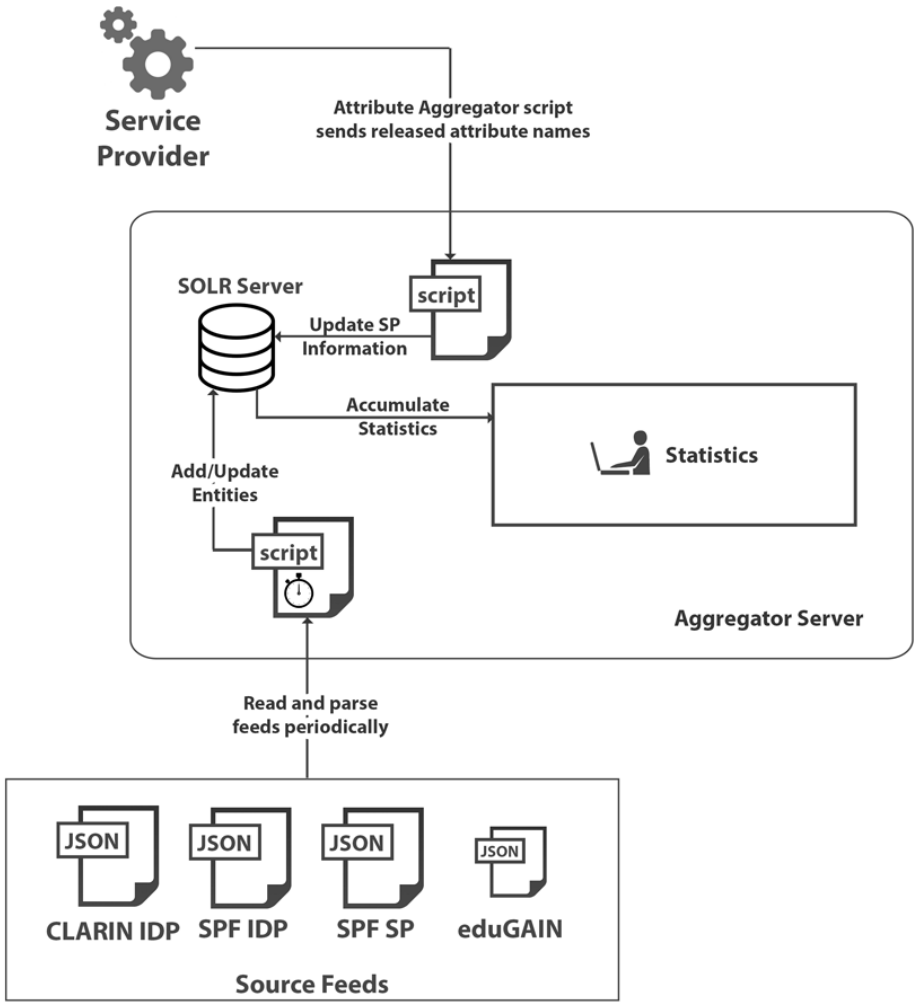


Figure 3: Gathering and showing statistics process flow.

5.2 Identity Provider Statistics per CLARIN SPF Service Operator

For every Service Provider that has deployed CLARIN Attribute Name Aggregator, the following statistics are being shown per federation:

- number of Identity Providers that have tried to access that Service Provider;
- number of Identity Providers that are part of SPF;
- number of Identity Providers that are part of eduGAIN;
- number of Identity Providers that meet the CLARIN recommended attribute release profile;
- number of Identity Providers that release at least one attribute that can be used for identification;
- number of Identity Providers that do not release any attributes.

These statistics can be used to evaluate the federations and help in improving the user experience in the problematic federations. Examples of the statistics can be found in Appendix 7.

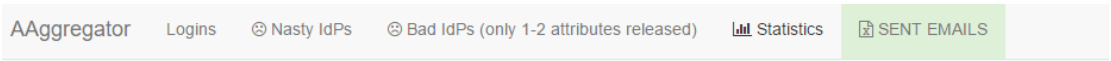
6 Conclusion

The CLARIN Attribute Name Aggregator provides unique ways of identifying and addressing problems of federated access on an inter-federation level for all member Service Providers in real time. Together with the statistics, it can be used to not only ensure the quality by monitoring but also to improve the quality and usability of the CLARIN SPF.
After extending the identity management software with all the required functionality, CLARIN Identity Provider can be moved to the new platform offering more secure, easier to manage and feature full platform for storing and using identities.

All the projects created in this deliverable are publicly available and fully documented; this allows other interested parties to fully re-use the outcomes.

7 Appendix

7.1 Attribute Aggregator Statistics per federations



IdP Statistics

In total 44 Federations
2482 IdPs are registered
out of which 2183 are in eduGAIN and 1350 in SPF

Federation	Total IdPs	Total IdPs from our feeds	In eduGain	In SPF
ACOnet Identity Federation (eduID.at)	**43**	**42**	17	42
ArnesAAI Federation	**50**	**14**	14	14
Belnet Federation	**40**	**18**	18	18
DFN-AAI Federation	**217**	**218**	45	218
eduID.cz Federation	**78**	**64**	64	64

7.2 Attribute Aggregator Statistics per Service Providers

SP Statistics

Clarin friendly = releases eduPersonPrincipalName or eduPersonTargetedID
ID friendly = Clarin friendly + releases eduPersonTargetedID-persistentID or mail
Nasty = releases 0 attributes

* Click on SP name to show/hide the breakdown of registration authorities.

Service Provider	IdP Count	In eduGain	In SPF	Clarin friendly	ID friendly	Nasty
https://ufal-point.mff.cuni.cz/shibboleth/eduid/sp						
	122	**92**	**89**	**91**	**73**	**12**
https://sp.clarin.si/						
	10	**8**	**10**	**7**	**6**	**0**
http://www.eduid.cz/	1	1	1	1	1	0
http://aai.arnes.si	3	3	3	3	3	0
http://ukfederation.org.uk	1	1	1	1	1	0
Registration Authority Unknown	1	0	1	1	0	0
https://idp.clarin.eu						
https://www.aai.dfn.de	4	3	4	1	1	0
http://sp.vs1.corpora.uni-hamburg.de						
	7	**2**	**7**	**3**	**3**	**0**
https://dspace-clarin-it.ilc.cnr.it/Shibboleth.sso/Metadata						
	6	**5**	**6**	**4**	**4**	**0**