Guidelines on token translation services

Published Date:	13-06-2017
Revision:	1.0
Work Package:	JRA1
Document Code:	AARC-JRA1.4C
Document URL:	https://aarc-project.eu/wp-content/uploads/2017/03/AARC-JRA1.4C.pdf

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

Table of Contents

1 Introduction 3

2 Use cases and Examples 3

2.1 Embedded token translation operation 3

2.2 Standalone token translation service 4

3 Guidelines 5

4 References 7

5 Glossary 7

Table of Figures

Figure 1.1: “Embedded” token translation 4

Figure 1.2: “Standalone” token translation 5

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

1 Introduction

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

2 Use cases and Examples

From the architectural point of view, while keeping in mind the federated AAI landscape described in “Blueprint Architecture” [AARC-BPA-Web], token translation operation might happen “seamlessly” to the user, or it may require an action from the user in order to perform the token translation operation. For the easier distinction between the two modes of operation, we will call the first “embedded” TTS, and the second one a “standalone” TTS. These two modes of operation are what is most commonly found in real life scenarios, although other modes are possible, and therefore will be further considered in this document.

2.1 Embedded token translation operation

Some services are created in a way that translate user attributes or tokens without the user action, or they are implemented in a way that user is not even aware of it. The need to translate credentials in the first place may arise due to the different technologies that user initially employs for authentication and the technology service itself internally utilize, for example. Additionally, user access the service in a continuous manner, without the need to change its user agent. An example could be that user does not have to authenticate through browser, and then use the generated credentials for a non-web access. However, we would still consider token translation operation to be “embedded” in a situation where initial registration for the service is a separate action from subsequent utilization of service. In terms of its position within the Blueprint document, this type of operation happens in the End Service layer or directly in the Proxy, as shown in Figure 2.1.

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

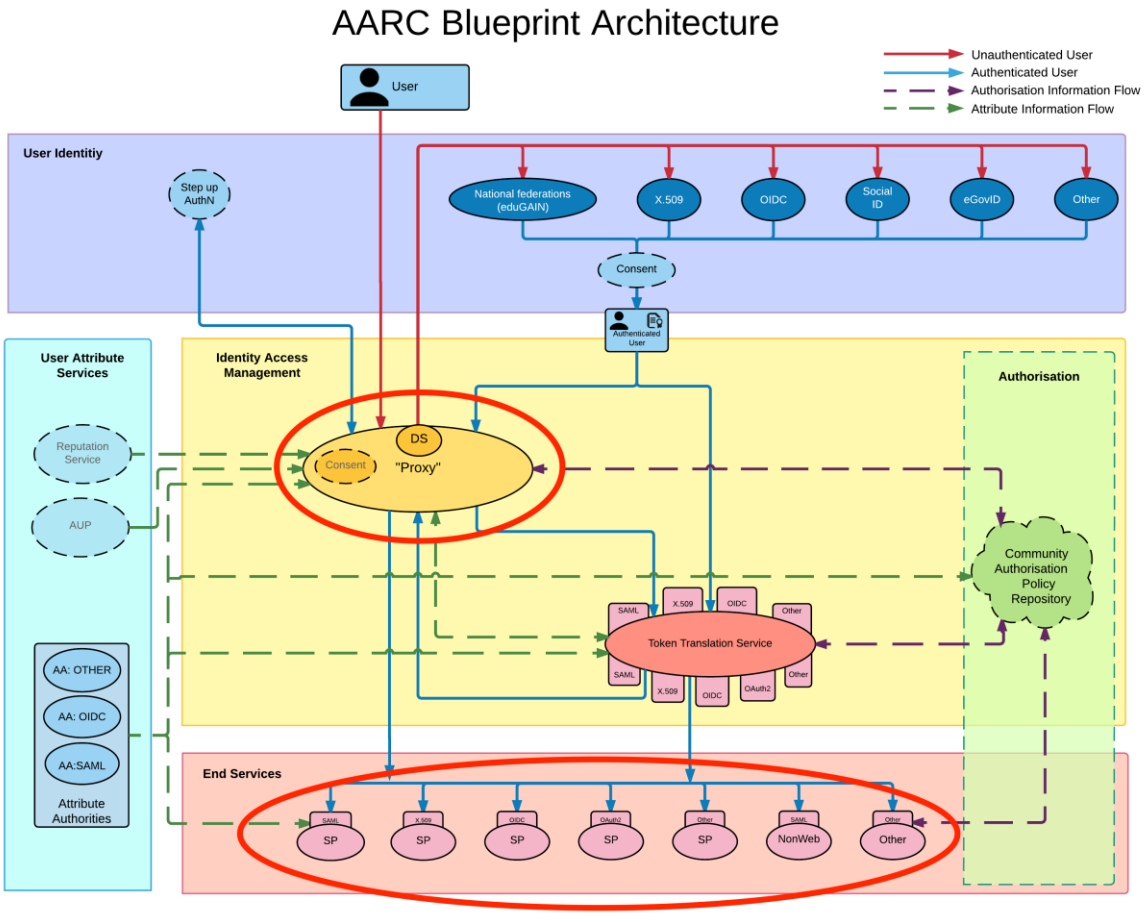

Figure 2.1: “Embedded” token translation

2.2 Standalone token translation service

This type of operation requires an explicit action from the user. If the token translation or generation happens in a browser, for example, user may access the portal where it selects which types of tokens should be generated, depending on the desired service. In this example, these tokens may later be used for a non-web access. Token translation functions as a “bridge” between user authentication and authorisation, and final generated credential. TTS takes user’s information (e.g name, mail, LoA, etc.) to generate/provision a credential later utilised to access/use a desired service. Again, usually this is done across different technologies, i.e. OIDC->SAML, OIDC->SSH keys, etc., but it can be also done between same technologies. In the latter case, something like “attribute enrichment” might come into place, or at least some additional information about the user should be supplied at this stage. These TTSs are usually deployed in two ways, as a service specific instance or they are operated as a centralized/shared service. The difference is mainly at which organizational level they are managed, whether they are used by a single, specific service, or they are trusted on a higher level, for example inside a federation. In the latter case, the credentials are not limited to a single, particular service, but the same credential can be used to access many services run by multiple

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

organisations/providers. While the operating principle might be the same as with the service specific TTS, the demand for security and trust level is usually higher, since the impact of abuse is higher, too. One example is generating IGTF certificates, since IGTF certificates are accepted by many sites across the world, because they have a well-defined policy that imposes requirements on the identification of the person to whom it may be issued. For a central TTS this means that it may only issue certificates if the incoming authentication was of a high-enough LoA. For both service specific and centralized TTS, their location inside the Blueprint Architecture is in the Translation layer, represented in Figure 2.2.

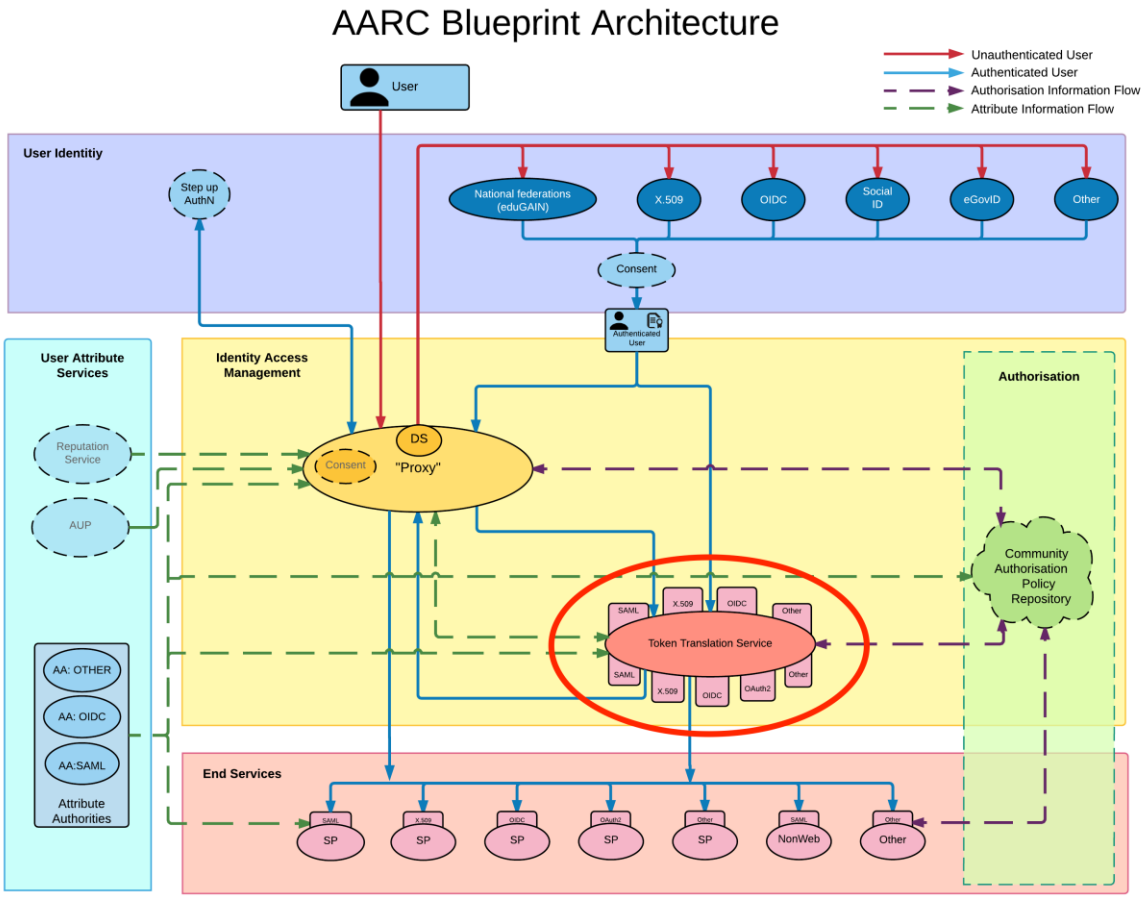

Figure 2.2: “Standalone” token translation

3 Guidelines

There are few guidelines to keep in mind with operating Token Translation Services. Some of them are listed below, but this is not an exhaustive list.

- **Consistency of user information**

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

While there are already solutions that translate SAML to OIDC and vice versa, or OIDC to X.509, SAML to X.509, or OIDC to SSH keys, one important point to watch is how information is translated between technologies. TTSs need to properly translate information included in the original token, to information included in the translated token. The different parts of the token or of the information need to be carefully considered, i.e. which token part is used for user authentication ("who are you") and which part is used for authorisation ("what roles/rights are granted to you") and how these are translated across different technologies. Best practices and recommendations for translating between federated authentication and X.509 certificates are listed in [AARC-JRA1.4I]. For SAML <-> OIDC mapping, there is an ongoing effort from the OpenID Connect for Research and Education Working Group (OIDCre) [OIDCre-SAML-OIDC]. Furthermore, in AARC, an effort was devoted to guidelines for implementing SAML authentication proxies for social media IdPs [AARC-JRA1.4G].

- **Deployment considerations**

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

- **Security considerations**

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

- **Transparency, data protection and data minimisation**

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

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

4 References

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

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

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

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

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

5 Glossary

AA	Attribute Authority
AAI	Authentication and Authorisation Infrastructure
GDPR	General Data Protection Regulation
IdP	Identity Provider
IGTF	Interoperable Global Trust Federation
LoA	Level of Assurance
OIDC	OpenID Connect
OIDCre	OpenID Connect for Research and Education
REFEDS	Research and Education FEDerations group
SAML	Security Assertion Markup Language
SP	Service Provider
SSH	Secure SHell

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