Mixing Heterogeneous Authentication and Authorization Infrastructures through Proxies

Angelo Furfaro[1] [a] and Giuseppe de Marco[2] [b]
[1] DIMES, University of Calabria, P. Bucci 42C, Rende (CS), Italy
[2] ICT Centre, University of Calabria, P. Bucci 22B, Rende (CS), Italy

Keywords: AAI, Single Sign-on, SAML2, OIDC, OAuth2, Digital Identity, Proxy.

Abstract: The ever increasing diffusion of digital services offered by institutional organizations and the need of interoperability among them have made crucial the role of Authentication and Authorization Infrastructures (AAIs). Numerous formats and technologies for data exchange have been developed in recent years and some of them have become very popular. This paper discusses the main challenges an organization has to face in making its services seamlessly available to end-users and client systems across multiple AAIs. An effective solution, relying on Authentication and Authorization Proxies, like SATOSA, which allows the interoperability of hybrid types of service providers and consumers, is described. In particular, a scenario is considered which envisages the support of heterogeneous (public) digital identity technologies for access to digital services on a university campus.

1 INTRODUCTION

During the last few years, many digital services, offered by both private and public organizations, got linked through Authentication and Authorization Infrastructures (AAIs) in order to support interactions across organization boundaries. Functionalities offered by AAIs are achieved thanks to the adoption of open protocols and standard data interchange formats, e.g. SAML2 (Cantor et al., 2005) based XML and OIDC (OpenID Foundation, 2014), or its underlying protocol called OAuth2 (IETF, 2012), based on JSON Web Token (JWT).

In the context of a SAML2 based AAI, we can distinguish these main roles: i) the *Service Provider (SP)*, which is a system offering a digital service to its clients, in terms of functions and/or resources, ii) the *Identity Provider (IdP)* or *Authorization and Authentication endpoint*, which is in charge of transferring to the relevant SP suitable information to determine the identity and also the authorization profile of the requesting entity if available, and iii) the *Attribute Authority* which release attributes. A SP may exploit a *Discovery service* (IETF, 2014b) allowing the user to choose the IdP though which to authenticate. Analogous roles are present in other AAIs with some little differences in their responsibilities. For example, in OIDC contexts SPs are called *Relying-Parties (RPs)* and IdPs are known as *OpenID Providers (OPs)*. In OAuth2 they are referred to as *Clients* and *Authorization Servers (ASs)*, respectively. Moreover, in OIDC/OAuth2 contexts the authorization decision is taken by the OP/AS while in SAML2 this is done by the SP on the basis of what received by the IdP. Functionalities of SAML2 Attribute Authorities can be implemented in OAuth2 by resorting to Resource Servers which implements a more general and distributed approach to resource management.

These settings allow users to register and authenticate once, across several different trusted authentication endpoints hosted by different organizations, and then access various resources made available by digital services through such AAIs (Lenz and Zwattendorfer, 2016; Jensen, 2012).

Because of these obvious benefits induced by the evolution of identity management technologies and by the possibility of outsourcing such role to trusted actors, e.g. governments, AAIs have experienced a rapid rise and a wide adoption (TechVision, 2018).

The European Union has recently introduced eIDAS (electronic IDentification, Authentication and trust Services) (Bender, 2015) which is a regulation on a set of standards for electronic identification and

[a] https://orcid.org/0000-0003-2537-8918
[b] https://orcid.org/0000-0002-3108-592X

Furfaro, A. and de Marco, G.
Mixing Heterogeneous Authentication and Authorization Infrastructures through Proxies.
DOI: 10.5220/0009392401240131
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 124-131
ISBN: 978-989-758-423-7

trust services for electronic transactions. The member States developed their own Identity Provider system accordingly, an example of which is the Italian Public System for Digital Identity called SPID (AgID - Agenzia per l'Italia Digitale, 2017). The National Institute of Standards and Technology (NIST) also introduced its own digital identity model in June 2017 (Grassi et al., 2017). Some recent studies evaluate the use of AAIs in conjunction with blockchain technologies to develop an infrastructure supporting service accountability across organizations (Furfaro et al., 2019). An approach to retrieve and transport new attributes through the eIDAS infrastructure in the context of academic services has been described in (Berbecaru et al., 2019).

In addition to the aforementioned public systems for digital identities and Research and Scholarship Institutions, the global scenario includes also other AAIs under the control of free-market players like Google, Facebook, Microsoft, Amazon and several others. Each AAI operates by relying on its own and homogeneous federation mechanisms with unified data processing and privacy policies in strict compliance with national and corporative regulations.

Nowadays, the consequent emergence of different protocols and data interchange formats and also the implementation of these in autonomous federative contexts make developers of digital services to face a new kind of boundaries due to the diversity of technologies adopted by different federations. This has made necessary to devise solutions that can enable the fruition of services, traditionally confined into the protocol borders of a specific organization, in the context of more AAIs despite the existence of different protocols. This has led to the development of AAI Gateways and Proxies that mediate the interactions between heterogeneous AAI actors, e.g. SAML2 SPs and IdPs needing to cooperate with OIDC/OAuth2 entities, handling protocol and data exchange format differences and thus enabling interoperability.

This paper describes a technologically sustainable solution aimed at integrating different formats and data exchange protocols through the adoption of the SATOSA proxy. It presents some examples and case studies that involve the adoption of AAI proxies, and finally a real use case developed within a European University Campus. Section 2 describes some of the leading technologies in Federated Identity Management contexts. An overview of the usage of proxy systems in computer networks and of their adoption in AAI contexts is given in Section 3. Section 4 describes the solution, based on SATOSA, adopted at University of Calabria. Finally, Section 5 concludes the paper.

2 COMMON SCENARIO

The typical form of identity federation involving higher education institutions is the so called *multilateral federation*, which relies upon a trusted 3rd party in charge of securely register and reliably publish all entities metadata (Trust Registry) in order to enable trusted interoperation between all IdPs and a SPs. Example of such federations, in the field Research and Education community, are InCommon (https://www.incommon.org/) and EduGAIN (https://edugain.org/).

Federation management processes require the use of a set of tools for the validation of the entities requesting to participate. Such processes and tools highly depend on the specific technology adopted in the relevant federation.

For example, in the case of SAML2-based federations, entities' metadata are collected and validated by a federation operator office and then they get aggregated into a single file which is digitally signed. Each entity needs to periodically download an updated copy of this metadata file, whose size consequently increases with the growth of the federation, in order to ensure synchronization within the federation so as to allow all the entities to recognize each other. To overcome the issues due to the handling of this type of registers, the Metadata Query Protocol (MDP) (Young, 2019a; Young, 2019b) has been recently introduced for enabling the dynamic and trusted retrieval of metadata about named entities.

Figure 1 shows a common SAML2 authentication session: i) a user-agent connects to a SP; ii) the SP redirects the user to a Discovery Service which allows to choose the relevant IdP; iii) the Discovery Service redirects the user-agent back to the SP which gets a reference (entityID) to the chosen IdP; iv) the SP issues a authentication request to the IdP and redirects the user-agent to it; v) the user-agent submits the user's credentials to the IdP; vi) the IdP produces a SAML response, regarding the outcome of authentication request; vii) the user-agent transfers the achieved SAML response to the SP.

OAuth2 federations have a similar approach regarding the management of internal trust registry and tools, but they offer additional features enabling various federation strategies. This way, alternative federative mechanisms adopted on top of WebFinger and Discovery Metadata Registry (OAuth Working Group, 2016; IETF, 2018), Dynamic Client Registration resources (IETF, 2015) and other kind of resources (called endpoints), allows federation operators to implement additional features, security checks and introspections functionalities, that handle client authentication requests and token validations on top

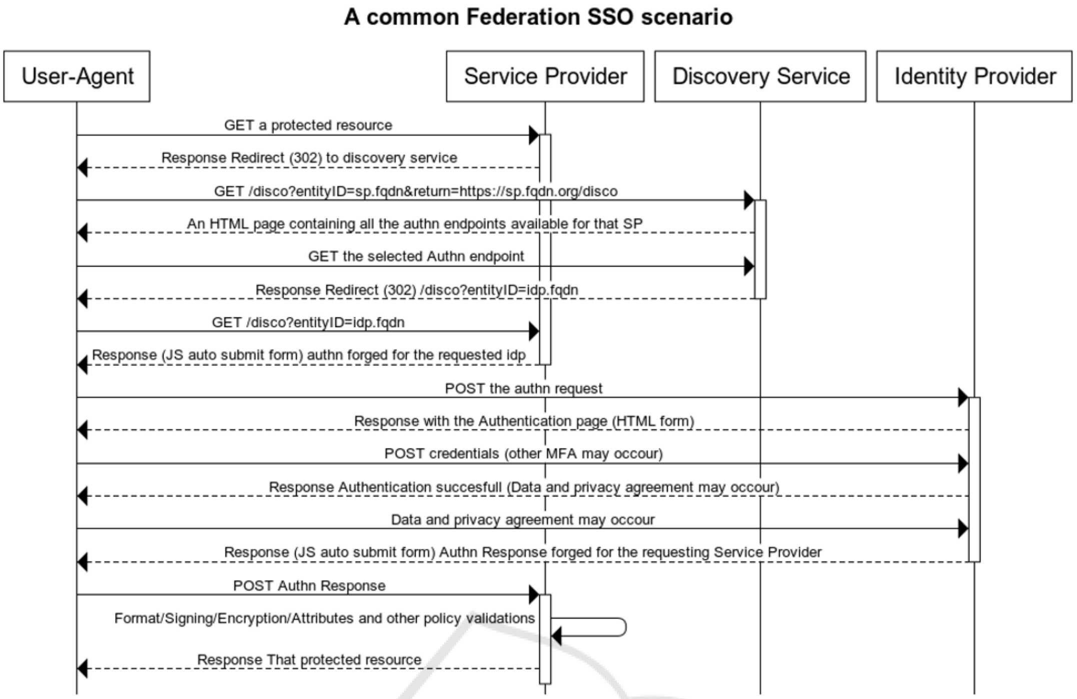

A common Federation SSO scenario
User-Agent
Service Provider
Discovery Service
Identity Provider
GET a protected resource
Response Redirect (302) to discovery service
GET /disco?entityID=sp.fqdn&return=https://sp.fqdn.org/disco
An HTML page containing all the authn endpoints available for that SP
GET the selected Authn endpoint
Response Redirect (302) /disco?entityID=idp.fqdn
GET /disco?entityID=idp.fqdn
Response (JS auto submit form) authn forged for the requested idp
POST the authn request
Response with the Authentication page (HTML form)
POST credentials (other MFA may occour)
Response Authentication succesfull (Data and privacy agreement may occour)
Data and privacy agreement may occour
Response (JS auto submit form) Authn Response forged for the requesting Service Provider
POST Authn Response
Format/Signing/Encryption/Attributes and other policy validations
Response That protected resource

Figure 1: Sequence Diagram of a SAML2 Authentication in a Federative Context with HTTP Redirect or POST Bindings.

of more articulated methods. In OIDC federations, where some of the OAuth2 core endpoints are involved, it would be nowadays possible to apply a further federation API that improves the federation architecture and strategies in a very significant manner (Roland Hedberg, 2019). It is also important to consider that to date there is no federative standards in the field of SAML2, the birth of the federations based on this protocol are based on good practices developed over time, otherwise in OIDC context the determination of OIDC Federation 1.0 (Roland Hedberg, 2019) is introducing an innovative approach to AAI federations.

Despite what technological innovations may be introduced, the federative mechanisms would only adopt the rigid approach to guarantee the bonds of trust towards and from every members, data privacy and integrity through the exchange channels. Once a federation adopts a technology and a shareable regulation, all of its members must in turn adhere to them in order to be able to join. Authorization and Authentication federation can be confined into a single organization (*local federation*), between two organizations (*bilateral-federation*), or can be a wider and more complex infrastructure built on top of the relationship between many organizations. More federations can join together by sharing the compliance to a same set of technological and administrative protocols (*multi-lateral federation*). In these latter cases, thousands of entities are involved in a federation of daily use. If we were to think about introducing a new federative mechanism or any other adjustment based on a different federative technology, in a legacy context, we should think about how to introduce this changes for all the involved services paying attention to the continuity of all them and also taking care of the cost-effectiveness of these actions. Let us consider the following scenarios requiring an adaptation/hybridization of the involved entities:

- An organization, whose digital services rely on a legacy infrastructure which is part of a SAML2 federation, need to add the support to OIDC to its SPs and IdP;
- An organization which has a solid infrastructure, built over the years above OAuth2, which needs to join also another federation using only SAML2;
- An organization which has to include for instance Microsoft ADFS IdPs in eduGAIN/InCommon federations. Many federations have problems with enforcing common policies, like releasing attributes dependent on entity category. There is the need to augment the set of attributes released by a non-compliant IdP with attributes from another source (AA or any other data store).
- An organization that would like to introduce the Identities Level of Assurance (Refeds, 2018) offered by the National Digital Identity Federation

(AgID - Agenzia per l'Italia Digitale, 2017) in its Credentials Provisioning systems in order to verify remotely the identity linked to the accounts. This would require a hybridization of the Credential Provisioning with another authentication layer to let the users to choice whether to authenticate with the legacy IdP or with a National Digital Identity Provider.

All the above scenarios fairly give the idea of the involved technological constraints and the perception that a systematic adaptation of an entire infrastructure can determine costs in time and resources. As discussed by the following sections AAI proxies are an effective solution allowing to overcome such issues.

3 ACHIEVING INTEROPERABILITY THROUGH PROXIES

In the field of computer networks, a *proxy* is a server application acting as an intermediary between two network endpoints, e.g. between an HTTP user-agent and a web server. A proxy behaves like a server, gathering the incoming requests from the clients, and plays the client role with respect to the actual target servers handling the entire session with corresponding requests and responses in between the two endpoints. A proxy can work in a *transparent operating mode*, i.e. it does not modify or masquerade in any way requests or responses (IETF, 2014a), or can be setup in other operational modes which enforce some kind of translation or data processing and manipulation. For example, proxies can be employed to add structure and encapsulation to systems or to carry out data security checks. The adoption of proxies would also allow to prevent security issues (Kobata and Gagne, 2006), protecting destination endpoints or translating a data transmission protocol from one to another, or to enable partitioning and load balancing of HTTP services across multiple machines.

In AAI contexts, proxies are used to aggregate many entities behind a single one. Some common use cases are the following:

- A proxy can be exploited to hide the presence of more digital services, all of which are under the control of a given organization, allowing them to appear like a single SP with respect to an IdP. All the SPs are federated to the the proxy, which acts as an IdP for them, through a registration process which is internal to the organization. The proxy masquerades all the authentication and authorization requests from many SP in a way that it would act as a single SP with respect the destination IdP. This implies that only the proxy must be registered as a SP in the target federation context.
- A proxy can be used as a *gateway* (or *reverse proxy*) which retrieves resources on behalf of a client from one or more servers. Gateways are often used to encapsulate legacy or untrusted information services. In AAI context, a gateway can be used to expose transparently more SPs behind it.

In order to understand how the adoption of proxies in AAI contexts allows to overcome the boundaries, due protocols and data exchange formats, to interconnect endpoints belonging to different federations, some potential scenarios are described in the following.

Let us consider a made-up organization `TheCampus`, e.g. a University, that has its own authentication infrastructure based on SAML2. Suppose that `TheCampus` wants to adopt a free market e-mail service provider, `ThatMail.com`, which supports only OAuth2. To handle this issue, a Proxy can be properly configured to behave as an OAuth2 Authorization Service endpoint with respect to `ThatMail.com` service and, at the same time, as a SAML2 SP federated to the `TheCampus` IdP. A user belonging to `TheCampus`, which wants to access `ThatMail.com` service (OAuth2 client), gets first redirected to the Proxy which, in turn redirects her/him to the `TheCampus` IdP. After the user is correctly authenticated, her/his attributes are transferred to the Proxy. Then, the Proxy builds from these attributes the proper OAuth2 Authorization which is sent to the `ThatMail.com` service.

Suppose that `TheCampus` joined, through its IdP, a Research and Scholarship Federation for allowing its users to access bibliographic resources provided by a third-party by means of a SAML2 SP. In addition, `TheCampus` would like to enable the users to authenticate also via a public OIDC Provider (IdP). Usually, such type of SAML2 SP is configured to exploit a Discovery Service or a *where are you from* (WAYF) web resource to let users to choice the relevant IdP. To support this scenario, `TheCampus` configures the Proxy to assume the role of the official SAML2 IdP in the Research and Scholarship Federation and to behave like a Relying-Party (SP) with respect to the OIDC provider. This way, `TheCampus` users accessing the bibliographic SP are first routed to the Proxy by the SP which then redirects them again to the `TheCampus` Discovery Service through which they can choose between the public OIDC provider or the local SAML2 IdP. In this case, the Proxy should be configured to handle an internal routing to manage which configuration to be used regarding the actual authentication endpoint. Even more, two different Authentication endpoints could have different configurations, even if both are based on SAML2, meaning that the Proxy should adopt different configurations depending by the target entity to deal with. In addition, the Proxy can work also as an *account linking* service, if it has to *link* the identity of the authenticated user to the corresponding TheCampus account, on the basis of the matching of the attributes returned by OIDC Provider (claims) with those received by Attribute Authorities or any other data sources.

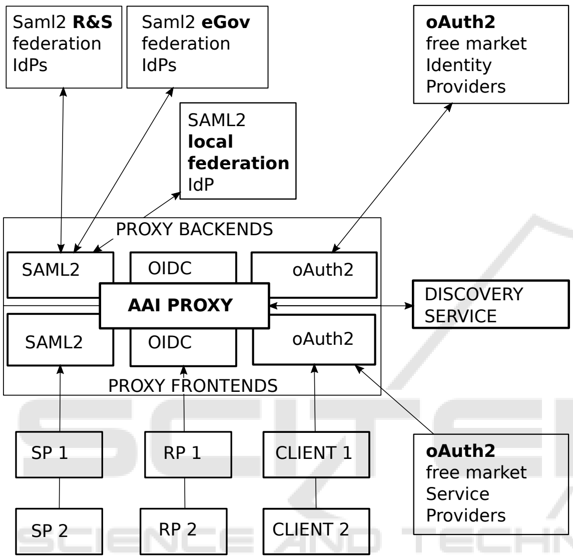

Saml2 **R&S** federation IdPs
Saml2 **eGov** federation IdPs
SAML2 **local federation** IdP
oAuth2 free market Identity Providers
PROXY BACKENDS
SAML2
OIDC
oAuth2
AAI PROXY
DISCOVERY SERVICE
SAML2
OIDC
oAuth2
PROXY FRONTENDS
SP 1
RP 1
CLIENT 1
oAuth2 free market Service Providers
SP 2
RP 2
CLIENT 2

Figure 2: SAML2 Service Providers, OIDC Relying-Parties and OAuth2 Clients behind the Proxy Backends Reach the Entities Masqueraded by the Proxy Frontends.

A Proxy can translate different data exchange formats, add or modify attributes and rewrite all data managed along the transaction. It can recover the attributes needed in a Research and Scholarship Service Provider (e.g. the schacPersonalUniqueID) from what returned by the National public provider (e.g. the Taxpayer Identification Number) applying rules of text manipulation, even more it can fetch additional attributes from other sources, e.g. LDAP, RDBMS or even a custom API or Attribute Authorities, owned by the TheCampus.

Figure 2, illustrates a more general scenario where the AAI Proxy enables the interoperation among the following assets used by TheCampus:

- a number of SAML2 SPs belonging to the TheCampus;
- some OAuth2 SPs (Clients) external to the TheCampus corresponding to external services owned by third parties;
- some OIDC SPs (Relying-Party) belonging to TheCampus;
- a public IdP working with SAML2 but having custom rules;
- the above public IdP working with a growing brand new OIDC infrastructure;
- a European IdP (eIDAS) based on SAML2 but with custom rules;
- a free market OAuth2 IdP that can guarantees at least the minimal Level of Assurance on the accounts managed by it;

The Proxy is the central entity handling all the requests and responses. It relies on SAML2, OAuth2 and OIDC backend and frontend modules to enable interoperability among all the previous entities and it is properly configured with custom backend/frontend routing and attribute translation rules. Figure 2 includes also a Discovery Service as an independent entity which can be exploited by the backend modules.

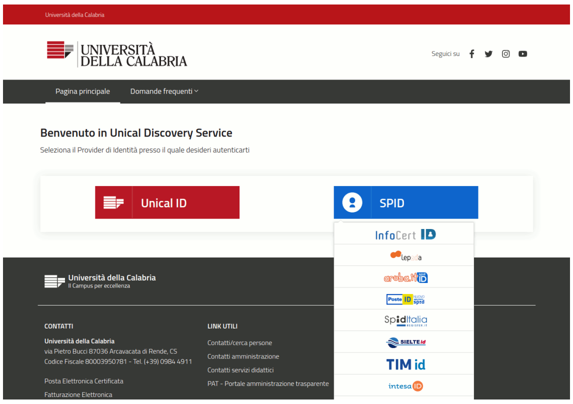
4 SATOSA USE CASE

SATOSA (Identity Python, 2019) is an open source proxy for translating between different authentication protocols such as SAML2, OpenID Connect and OAuth2. Its development began on 2015 at the Umeå University and since 2017 it is maintained and developed by the IdentityPython Organization (Identity Python, 2017).

SATOSA supports some well known use cases to link different kind of endpoints or simply can deal with legacy configurations that needs to be dynamically translated. For example SATOSA can enable SAML2 SPs to work with multiple SAML2 IdPs having different configurations. SATOSA allows to connect a SAML2 SP to multiple social media IdPs that works only with OAuth2 or OIDC. It also makes possible to mirror an IdP by generating SAML2 metadata corresponding that provider and create dynamic endpoints which are connected to a single IdP. SATOSA supports all the scenarios exemplified in Section 3.

Here, we describe the adoption of the SATOSA Proxy in a real setting at University of Calabria (Unical), an Italian university campus with a population of circa 20000-25000 daily users.

The digital Identity Management infrastructure of Unical stores users' attributes on top of a common Research and Scholarship model, in one or more LDAP server accordingly to eduPerson (REFEDS, 2019) and SCHAC schemas (REFEDS, 2015). Legacy RDBMS data sources have also been adapted



Benvenuto in Unical Discovery Service

Seleziona il Provider di Identità presso il quale desideri autenticarti

Università della Calabria
Il Campus per eccellenza

Università della Calabria
via Pietro Bucci 87036 Arcavacata di Rende, CS
Codice Fiscale 80003950781 - Tel. (+39) 0984 4911

Figure 3: The Discovery Service Developed in Campus for the User Selection of an Authentication Endpoint (IdP).

to work as LDAP backends, through the adoption of `slapd-sql` and `slapd-sock` backends. Its attribute release policy also provides the code of conduct related to type of organization in relation to the information provided to service providers (Internet2, 2017). The Identity management infrastructure also handles the provisioning, upgrade and de-provisioning of the users' accounts, their personal data and authorization attributes. Each digital identity must guarantee a minimum Level of Assurance (LoA) to be in compliance with the law and regulations in force. The `Unical` identity infrastructure is equipped with a SAML2 IdP to which more SPs are linked to. Some of these SPs are internally operated, others are run by third-party free market providers (i.e. the bibliographic services) and some other belong to third-party organizations and made accessible through the EduGAIN federation.

In this setting, a SATOSA Proxy has been put in operation to work as a SAML2 SP federated both with the `Unical` IdP and with all the SPID compliant Italian IdPs. This required two SAML2 backends having with different configuration and made it necessary to develop a dedicated SPID compliant backend and a routing system based on the target endpoint. In addition to the `Unical` IdP, the internally operated SPs have been also federated to the SATOSA proxy as alternative IdP. This way, users can choose to authenticate either with their SPID identity or with their `Unical` identity through a custom Discovery Service (Fig. 3) which has been developed in strict compliance with the SPID regulation and in adherence with the `Unical` visual identity. The sequence diagram of Figure 4 illustrates a typical authentication flow involving the discovery service and the intermediation of the SATOSA proxy.

In order to make the SATOSA Proxy to work in a real SPID compliant production environment, we developed a purposely designed SAML2 backend and a custom microservice for managing the routing of the incoming authentication requests to the correct SAML2 backend.

Table 1: Contributions to pySAML2 and SATOSA Projects.

Project	Pull req id	Description
pySAML2	597	Added configurable signing and digest algorithm to SP and IDP
pySAML2	602	xsd type - Added date type
pySAML2	625	Register common namespace prefixes
pySAML2	632	[Documentation] name_id_format, allow_create and metadata folder
pySAML2	634	xmlsec temporary files deletions
SATOSA	214	Added support for selectable SIGN and DIGEST algs in saml2 backend
SATOSA	216	signature and digest algorithm policy configuration
SATOSA	220	Micro Service, Decide backend by target entity ID
SATOSA	226	Encrypt assertions in frontend authnresponse
SATOSA	240	MicroService Ldap attribute store, more connection parameters in configuration

The newly introduced SAML2 backend was developed as a common SPID SP within SATOSA. Making it compliant to the AgID technical specifications required some modifications to both pySAML2 and SATOSA source code. Table 1 summarizes the contributions we made to the official projects.

To asses the suitability to operate in a production environment, the backend undergone a set of security checks: some of which are required by the AgID accreditation procedure and others are part of the standard production profile for the `Unical` digital services.

The choice of adopting SATOSA as the proxy solution for `Unical` was mainly due to its adherence to specifics standards in force in Research & Scholarship contexts. As alternative solutions we also have preliminary evaluated the following two options: i) *Shibboleth IdP 3.4.x* configured with an "External Authentication" (https://wiki.shibboleth.net/confluence/display/IDP30/ExternalAuthnConfiguration) and an additional Shibboleth SP coupled with it and ii) *Keycloak* (https://www.keycloak.org).

The first option would have required a more complex and expensive integration procedure and would have addressed only SAML2 cases. The second option has not been taken because Keycloak is a monolithic Identity Manager. The use of SATOSA , although introduced significant development costs, represented the most flexible and maintainable solution over time for its modular design and its ability to evolve towards the inclusion of new technologies.

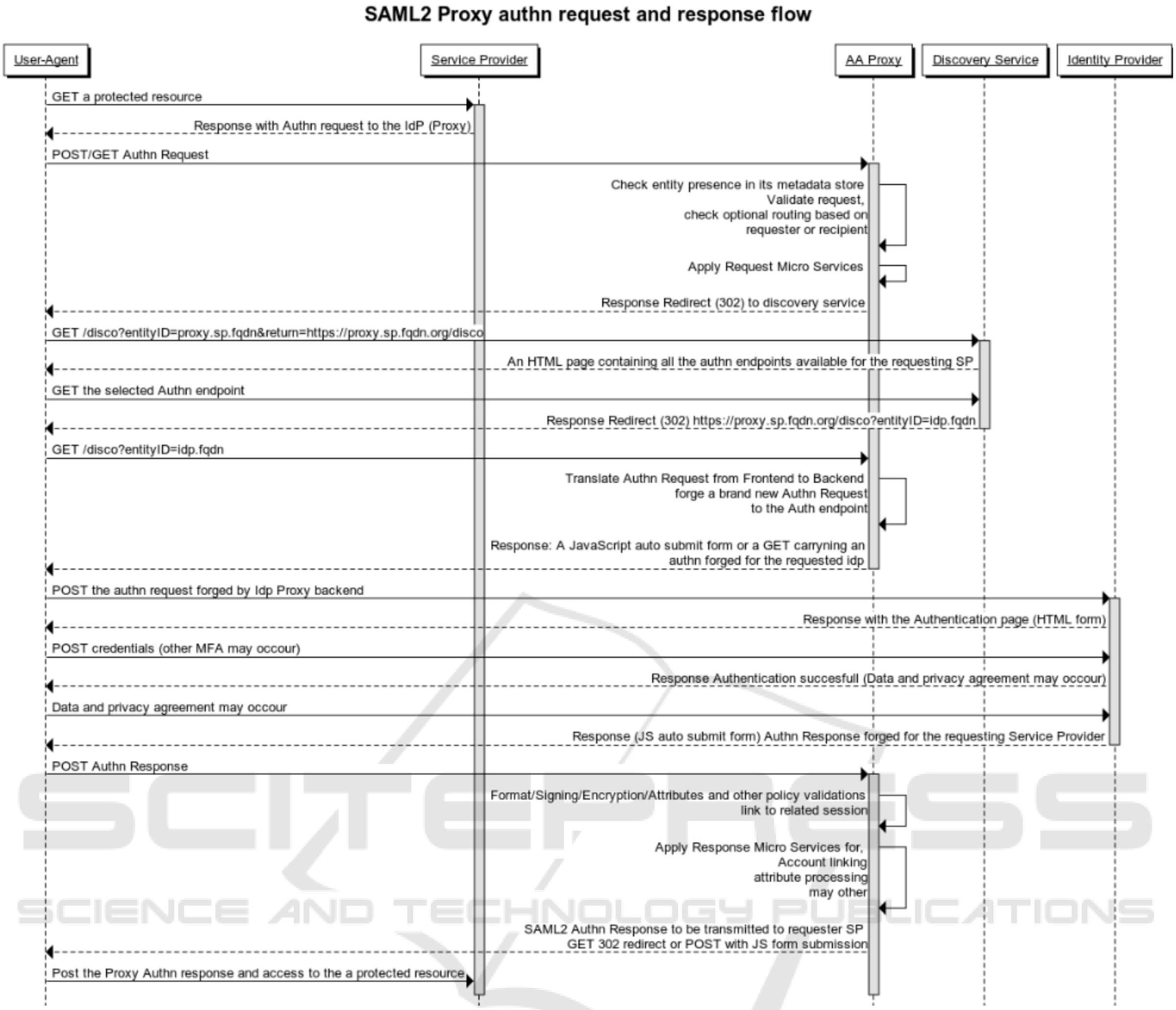

SAML2 Proxy authn request and response flow
User-Agent
Service Provider
AA Proxy
Discovery Service
Identity Provider
GET a protected resource
Response with Authn request to the IdP (Proxy)
POST/GET Authn Request
Check entity presence in its metadata store
Validate request,
check optional routing based on
requester or recipient
Apply Request Micro Services
Response Redirect (302) to discovery service
GET /disco?entityID=proxy.sp.fqdn&return=https://proxy.sp.fqdn.org/disco
An HTML page containing all the authn endpoints available for the requesting SP
GET the selected Authn endpoint
Response Redirect (302) https://proxy.sp.fqdn.org/disco?entityID=idp.fqdn
GET /disco?entityID=idp.fqdn
Translate Authn Request from Frontend to Backend
forge a brand new Authn Request
to the Auth endpoint
Response: A JavaScript auto submit form or a GET carryning an
authn forged for the requested idp
POST the authn request forged by Idp Proxy backend
Response with the Authentication page (HTML form)
POST credentials (other MFA may occour)
Response Authentication succesfull (Data and privacy agreement may occour)
Data and privacy agreement may occour
Response (JS auto submit form) Authn Response forged for the requesting Service Provider
POST Authn Response
Format/Signing/Encryption/Attributes and other policy validations
link to related session
Apply Response Micro Services for,
Account linking
attribute processing
may other
SAML2 Authn Response to be transmitted to requester SP
GET 302 redirect or POST with JS form submission
Post the Proxy Authn response and access to the a protected resource

Figure 4: A SAML2 Authn Flow with HTTP-Redirect or POST Bindings in Acts Using the AA Proxy. The Discovery Service Is Configured in the Proxy Backend.

5 CONCLUSIONS

The adoption of SATOSA within the `Unical` identity management infrastructure introduced the need to customize the SATOSA source code to improve its behaviour and to cope with the local requirements. In particular, customizations and adjustments were made on error management, on the routing system based on the endpoint entity ID, on some specifications for the signature and encryption algorithms employed.

The use of the SATOSA proxy allowed to update the `Unical` process for credential provisioning such that each citizen, owing a SPID identity, could easily ask for a Campus' digital account by exploiting her/his public identity. This allowed to inherit the SPID's LoA for the accounts generated in this way. Because SPID IdPs are considered *authoritative* sources of identity attributes, there is no longer need to accomplish a face-to-face interview in the credential release and activation phase. The provisioning system is also strengthened by the support of an Attribute Authority based on OAuth2, that the SPs can use to recover the user's authorization profile by querying specific attributes starting from the received SPID attributes. The adoption of the SATOSA produced a significant simplification in the workflow for the accreditation of a `Unical` SP to the SPID federation. The SATOSA proxy is the only `Unical` service which is directly federated with the SPID system and it acts as an aggregator service for the other `Unical` SPs which only need to federate with it. Without the SATOSA proxy, each `Unical` SPs, wanting to allow SPID based authentications, would had to repeat the SPID registration procedures increasing costs in time and bureaucracy for both sides, `Unical` and AgID office. Thanks to the use of the proxy the registration process remains locally confined.

Finally, the imminent renewal of some critical services, e.g. the IT document flow management applications, have raised the need to comply with OAuth2. Through SATOSA it has been possible to guarantee this integration by keeping the legacy authentication endpoint, i.e. the SAML2 IdP, without any change. It was just necessary to add and configure the proper OAuth2 AS frontend in the SATOSA Proxy, so as to connect all the OAuth2 clients that need it.

ACKNOWLEDGEMENTS

We would like to thank: the SPID project staff (Michele D'Amico, Walter Arrighetti, Antonio Florio, Paolo Smiraglia and Antonio Giovanni Schiavone); Roland Hedberg as OpenID Connect Federation 1.0 Editor, pySAML2 and SATOSA architect; the Identity Python Community and in particular John Paraskevopoulos.

This work has been partially supported by the project P.O.R. "SPID Advanced Security - SPIDASEC" (CUP J88C17000340006).

REFERENCES

AgID - Agenzia per l'Italia Digitale (2017). Spid - regole tecniche. https://media.readthedocs.org/pdf/spid-regole-tecniche/latest/spid-regole-tecniche.pdf.

Bender, J. (2015). eIDAS Regulation: EID - opportunities and risks.

Berbecaru, D., Lioy, A., and Cameroni, C. (2019). Electronic identification for universities: Building cross-border services based on the eIDAS infrastructure. *Information*, 10(6):210.

Cantor, S., Kemp, J., Maler, E., and Philpott, R. (2005). Assertions and protocols for the OASIS Security Assertion Markup Language (SAML) v2.02. Oasis standard specification.

Furfaro, A., Argento, L., Sacca, D., Angiulli, F., and Fassetti, F. (2019). An infrastructure for service accountability based on digital identity and blockchain 3.0. In *2nd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock 2019)*. IEEE.

Grassi, P. A., Garcia, M. E., and Fenton, J. L. (2017). Digital identity guidelines: revision 3. NIST Special Publication 800-63-3.

Identity Python (2017). Your identity stack in python. https://idpy.org/.

Identity Python (2019). SATOSA. https://github.com/IdentityPython/SATOSA.

IETF (2012). The oauth 2.0 authorization framework. https://tools.ietf.org/html/rfc6749.

IETF (2014a). Hypertext transfer protocol (http/1.1): Message syntax and routing. https://tools.ietf.org/html/rfc7230.

IETF (2014b). Web services dynamic discovery (ws-discovery) version 1.1. http://docs.oasis-open.org/ws-dd/discovery/1.1/wsdd-discovery-1.1-spec.html.

IETF (2015). Oauth 2.0 dynamic client registration protocol. https://tools.ietf.org/html/rfc7591.

IETF (2018). Oauth 2.0 authorization server metadata. https://tools.ietf.org/html/rfc8414.

Internet2 (2017). Research and scholarship for idps. https://spaces.at.internet2.edu/display/InCFederation/Research+and+Scholarship+for+IdPs.

Jensen, J. (2012). Federated identity management challenges. In *2012 Seventh International Conference on Availability, Reliability and Security*. IEEE.

Kobata, H. and Gagne, R. (2006). Securing computer network communication using a proxy server. US Patent App. 10/766,871.

Lenz, T. and Zwattendorfer, B. (2016). Towards cross-border authorization in european eID federations. In *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE.

OAuth Working Group (2016). Oauth 2.0 bound configuration lookup. https://tools.ietf.org/id/draft-hunt-oauth-bound-config-00.html.

OpenID Foundation (2014). Welcome to OpenID Connect. https://openid.net/connect/.

REFEDS (2012, 2019). Standard and specs - eduperson. https://wiki.refeds.org/display/STAN/eduPerson.

REFEDS (2015). Standard and specs - schac. https://wiki.refeds.org/display/STAN/SCHAC.

Refeds (2018). Refeds assurance framework ver 1.0. https://refeds.org/assurance.

Roland Hedberg (2019). Openid connect federation 1.0. https://openid.net/specs/openid-connect-federation-1_0.html.

TechVision (2018). The future of identity management (2018-2023).

Young, I. (2019a). Metadata Query Protocol. Internet-Draft draft-young-md-query-11, IETF Secretariat. http://www.ietf.org/internet-drafts/draft-young-md-query-11.txt.

Young, I. (2019b). SAML Profile for the Metadata Query Protocol. Internet-Draft draft-young-md-query-saml-11, IETF Secretariat. http://www.ietf.org/internet-drafts/draft-young-md-query-saml-11.txt.