Secure idenTity crOss boRder linKed - eLearning and Academic Qualifications Pilot at IST

Simon Esposito
simon.esposito@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisbon, Portugal

June 2015

Abstract

The demand for federated identity management systems is growing. With the current thrive of online services on the web, the interest of governments in offering online public services for citizens and organizations is on the rise. This creates the need for digital identity management systems that can provide a high level of authentication assurance while keeping the risk of using such services at a minimum. The Secure idenTity acrOss boRders linKed (STORK) Large Scale Pilot (LSP) has been presented as a solution to this demand, by creating a unified European electronic identification and authentication area between the European Union (EU) Member States (MSs). In this work we will present the proposed solution and implementation of Within the scope of the academic STORK LSP.

Keywords: STORK, Identity Management System, Federated Identity Management, Federated Attribute Exchange, eGovernment services.

1. Introduction

The STORK project[1] aims at creating a single digital identification and authentication area over Europe, establishing electronic Identities (eID's) for individuals and legal entities.

To achieve this goal, the interoperability of electronic Identity Management Systems (IdMs) across the European MSs is crucial. Countries like Austria [12, 7, 10], Estonia[8, 5], Finland[11], The Netherlands and others already have IdMs in place that were implemented without the establishment of clear common goals across the EU. This has caused a high heterogeneity of implementations which makes interoperability difficult and poses several security and privacy challenges.

Some MSs also deployed SmartCards (SCs) for their citizens [1] which function as electronic identity cards. These cards are called European Citizen Card (ECC), and have built-in support for the Public Key Infrastructure (PKI) enabling Electronic Identities (eIDs). This means that these eID cards provide the necessary functionalities to citizens that allows them to authenticate with high assurance to online services. However, there are countries (e.g. the United Kingdom) that still don't provide any type of Identity Card implementation at all.

Because of the heterogeneous IdM solutions already adopted by those MSs, interoperability becomes an issue as many different technologies are used leading to the demand for a federated standardization of such systems [9]. This arises other problems such as: legal issues; assuring the user privacy across the different IdMs transactions; give the user information about the policies of release of his attributes; enforcing these policies at the Service Providers (SPs); and allow the user to maintain his anonymity (to an extent). Moreover, for usability purposes, the user shouldn't need to authenticate to all IdMs individually, but instead, a relation of trust should exist between organizations to provide Single Sign On (SSO) across them in order to mitigate this issue.

For the project to succeed, a reasonable level of assurance must be given in order to let them build trust in the system and let them rest assured that their data is securely protected and safely delivered across MSs. The residual risk of using the system will only be accepted by citizens and organizations if a certain level of trust, assurance and usability is achieved, enabling the uptake of the system.

[1] https://www.eid-stork2.eu

2. Background

2.1. Federated Digital Identity

An identity federation is an established partnership between organizations, countries or any number of entities to allow the exchange of individuals identities information and attributes using common agreed namespaces, protocols and technologies to enable interoperability and create a network of trusted relationships between the federation members. Because federations involve multiple organizations, problems such as liability, trust and security become much more relevant [3], as many more stakeholders are involved in a federation, making federated digital identity management much more challenging than managing eIDs within a single organization, even if at a multi-domain level.

2.2. Identity Management Models

IdM models can be divided into four categories: isolated, centralized, federated and decentralized (distributed) [2, 4, 13].

Isolated Model

In the isolated model, the Identity Provider (IdP) and the SPs are the same entity. This is the most common model currently on the Internet, where SPs are responsible for the management of their users credentials. This model is the most popular, but also the one with the most drawbacks: it is very burdensome for the users, which need to remember their credentials for each SP, often leading to poor/insecure password choices, hence lowering security; it is also burdensome for the SPs themselves, as managing user credentials has infrastructure and security costs; and finally, SPs control over credentials might disclose the user private information without consent, leading to privacy loss. SPs often claim to protect the users privacy through Privacy Policy Agreements (PPAs), but to what extent they are enforced is often a grey area.

Centralized Model

The centralized model separates the IdP and SPs roles, having a single centralized IdP that stores all the users credentials/attributes. This is often used in mid-to-large organizations where SSO solutions are in place, alleviating some of the burden of the users having to manage multiple credentials, as they will use the same set across all the services provided within the organization. It also alleviates the management of credentials on the organization itself, as it eases the management of high numbers of users by having a single place of storage. This model however, suffers even more of the privacy issue, as more data of the user is stored in a single place; which is also a security threat as the centralized IdP is a single point of failure.

Federated Model

The federated model establishes an identity federation where SPs rely on any IdP within it. By relying on the trust network established, the SP might even not previously know the IdP where a user authenticates. This has many advantages: SPs need not store any user information, enhancing privacy and being more cost-effective. It provides SSO functionality across different SPs, enhancing usability and overall security for the user. A drawback of this model is the introduction of the IdP discovery problem. Another is that, since the user information might be scattered across different IdPs, a solution for attribute aggregation must be implemented. These drawbacks introduce additional complexity to the system but alleviate the SPs from identity management altogether, having to exclusively manage the federation trust network (which usually involves the PKI).

Distributed Model

In the distributed model, the IdP is a security token (such as a SmartCard, SIM card or any other) held by the user itself. The credentials are securely stored in the token and are released upon user authentication through a Personal Identification Number (PIN) upon request. These tokens are issued by a trustworthy entity (such as the government or a national bank), which validate the trustworthiness of the stored credentials. This is the model that gives more control to the user over his credentials because he is in possession of the tokens. A disadvantage however, is that SPs need to handle a possible wide range of different tokens and the inherent technologies and standards. The tokens also have other problems: they usually have little storage capacity (e.g: SmartCards) and also, the secure update of outdated credentials within the tokens is difficult. These solutions have to rely on distributed Attribute Providers (APs) (the tokens) and additionally have to face discovery and attribute aggregation problems.

2.3. The Portuguese IdM Architecture

The Portuguese national IdM relies on its infrastructure and the European Citizen Card (ECC) issued to all citizens as an authentication token. The eID SmartCard contains two digital public key certificates: one for user authentication and another for digitally signing documents. Both functionalities are protected by PIN codes that are sent to the citizens via a secondary channel (postal service) so that they are more difficult to compromise, as an identity thief would need to have access to both

the eID and the PIN number to be able to commit fraud.

The national IdM is called SCAP (Sistema de Certificao de Atributos Profissionais - Certification System for Professional Attributes). The global architecture of SCAP can be seen on figure 1.

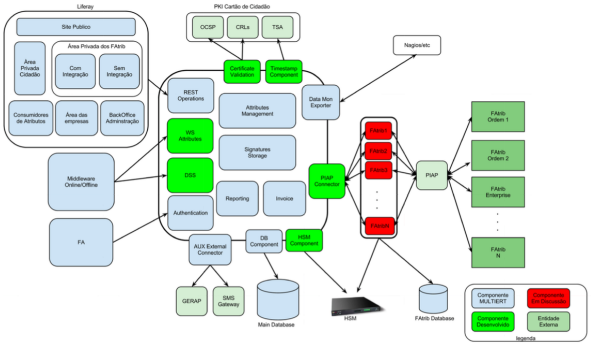

Figure 1: SCAP global architecture.

The goal of SCAP is to provide a national system to enable citizens to retrieve digitally certified attributes from other public institutions or APs. The attributes can then be consumed by service providers as credentials. An example would be the release of an attribute stating that a citizen is a civil engineer, since this attribute is certified by another national institution called "Ordem dos Engenheiros" (OE) which is responsible for qualifying civil engineers with the entitlement to sign charters.

As can be seen in Figure 1, the system is comprised by several different modules and frameworks, but on this document we will only discuss the most relevant:

- **FA:** Citizen authentication module;
- **PIAP:** Interface with APs module;
- **HSM:** The Attribute Signature hardware module;

The FA module allows a citizen to authenticate himself using his Portuguese ECC with a SmartCard reader plugged into his computer and having the provided middleware installed on the machine. When a SP requests authentication, the user's browser is redirected to the FA using the *SAML Web Browser SSO Profile*. The FA web page loads a signed Java applet that is executed by the browser (after asking the user if he trusts the signature used to sign it and the applet itself). The applet then communicates with the SmartCard reader and prompts the user for his PIN number to get access to the ciphered information within the ECC.

A token with the user's identity is released from the ECC and sent securely to the FA through the middleware to authenticate the user. If some citizen's attributes were also requested by the SP, if they are available on the government's database (e.g.: the citizen's address or NIC) they are retrieved by SCAP, otherwise the request is relayed to the PIAP.

The Plataforma de Interoperabilidade da Administracao Publica (PIAP) is the component that establishes secure connections with all the APs using a Simple Object Access Protocol (SOAP) web service. When an attribute request is received, the PIAP asks all the APs it knows for attributes, receives them and sends them back to Sistema de Certificao de Atributos Profissionais (SCAP).

The attributes aren't signed by the APs themselves, but instead, SCAP is responsible for the digital signature of the attributes retrieved from the possible several APs. The signature keys are established when an AP joins the infrastructure and are always stored by SCAP. This way the APs only need to worry about securing the connection with the PIAP, but not with private key management, which is all centralized and controlled by Agncia para a Modernizao Administrativa (AMA).

This leads us to the function of the Hardware Security Module (HSM), an hardware component for cryptoprocessing that signs the attributes with the private key of the respective AP.

After all the above processes are terminated, the SAML Response is constructed with the signed assertions and it is returned to the requesting SP.

This overview doesn't include the integration with STORK, which requires, among other work, the deployment of the national PEPS and its connection with the infrastructure described above.

2.4. Secure idenTity crOss boRders linKed

The STORK project [14, 6] aims to solve the problems not addressed by MODINIS, while keeping the key features of the PRIME Project, allowing an European citizen to access eGovernment services and other SPs belonging to institutions and organizations residing in other MS, authenticating to them using an ECC card or some other token issued in his residing MS. This implies the roaming of a citizen's attributes across different European countries.

2.4.1 Architecture Components

The STORK architecture is comprised by two main components: the middleware and the PEPS (Pan-European Proxy Service). The former is a middleware that is installed on SPs and IdPs and allows them to communicate directly; the latter is a network of proxies that act as intermediaries for cross-border identity requests, creating links of trust between MSs and bridging the isolated "islands" that form the national IdM infrastructures. These two components can function seamlessly

together, rendering the STORK infrastructure capable of handling both the centralized and distributed models. The data format used to exchange authentication and attribute information in the communications between PEPS and related entities (SPs and IdPs) is the Security Assertion Markup Language (SAML) standard, and the messages are sent using a Representational State Transfer (RESTful) approach.

The PEPS Model

This model relies in a network of proxies established within the EU. At least one proxy per MS is necessary at a national level, which is responsible to bridge the national IdM infrastructure with the PEPS network of proxies. This allows establishing an interoperability layer on top of the different MSs deployed infrastructures that abstracts from the specificities of the individual solutions. This model scales well not only because of this abstraction, but also because it creates a system of trust relay, where each MS only needs to manage the trust relation between the national PEPS proxy and the existing national IdM infrastructure by inherently trusting in the network of PEPS proxies. Fig. 2 shows an example of the trust circles that are created by the PEPS model and its basic process flows.

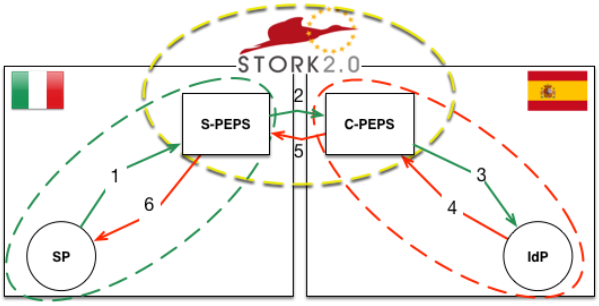

Figure 2: Trust networks in the PEPS model. In this example, an Italian SP is trying to access a Spanish IdP. The Italian SP trusts the respective national PEPS, that in turn trusts the Spanish PEPS, which trusts the Spanish IdP. This trust relay solution minimizes the management of trust relations, allowing the PEPS model to scale to a European level.

The process flow and basic function is described in Figure 2. A PEPS server can either act as a so-called S-PEPS (PEPS in the state of the SP) or as a C-PEPS (PEPS in the state of the citizen). This nomenclature specifies the role of the PEPS in a given transaction: the S-PEPS receives an authentication request from a SP (1) and acts as the intermediary that routes it to the correct C-PEPS (2) (this is achieved by querying the user where he is from on the S- PEPS through a Where Are You From (WAYF) interface), that in turn is responsible to route the request to the correct national IdP (3). The IdP sends the response back to the C-PEPS (4), which then sends it back to the requesting S-PEPS (5). The S-PEPS asserts to the SP that a user has been successfully authenticated on the IdP by sending back the SAML response (6), otherwise an error message is sent. If the SP also requested some user's attributes on step (1), the S-PEPS queries the user if he authorizes such request before step (2), and when receives the response from the C-PEPS in step (5), prompts the user again to authorize the release of the attributes to the requesting SP.

The middleware Model

The middleware component enables the decentralized deployment of STORK. It has the advantage of providing better privacy and end-to-end security. Despite the advantages, the decentralized model faces scalability issues because it forces SPs to have to support the different foreign eID tokens that exist, which may be based on different protocols and technologies. To support these requirements, the middleware component has a modular and extensible design, so that the different IdM technologies support can be added through modules to the core component, called Modular Authentication Relay Service (MARS) or the (Virtual Identity Provider) VIDP core. The structure of the MARS can be seen in Figure 3.

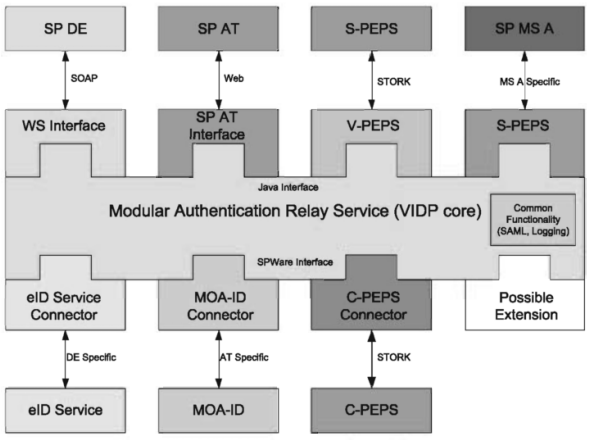

Figure 3: Modular Authentication Relay Service (MARS). Adapted from [14]

The core component is responsible for handling the SAML messages, logging and runtime deployment of the extension plug-ins and plug-ons. These extensions allow the implementation of the country-specific national IdM components (also called SP-

Ware) to integrate them into the STORK infrastructure. The plug-ons and plug-ins must implement two MARS interfaces, respectively: The Java and the SPWare interfaces. The former handles incoming authentication and attribute requests from the SPs by implementing the necessary protocols to interpret them (e.g: SOAP for the German SPs and REST for Austrian SPs), and route them to the latter, which is implemented by the SPWare connectors that are responsible for implementing the necessary technology to connect to the national-specific IdM infrastructure. Figure 3 illustrates the German and Austrian Java plug-ons, as well as the respective SPWare connectors for the German (eID Service) and Austrian MOA-ID IdMs.

As seen in Figure 4, the VIDP has a S-PEPS interface and a C-PEPS connector. This is because the middleware also supports the centralized PEPS model, and is in fact running in the proxies themselves, allowing interoperability between the centralized and distributed model the mix model. For the middleware to support both the distributed and centralized models, the following components are part of the middleware architecture:

V-PEPS: This plug-on is responsible for receiving STORK authentication requests messages from a S-PEPS.

C-PEPS Connector: The C-PEPS connector plug-in is the endpoint of the VIDP that routes a S-PEPS request to the citizen's country C-PEPS.

These components create the necessary conditions to enable full interoperability of the STORK infrastructure for countries that have either centralized or decentralized IdM architectures. As seen in Figure 4, Spain has a centralized model, against the decentralized approach of Austria. The diagram shows how the middleware behaves in 4 possible SP to IdP pair interactions:

Austria to Austria: In this case, the SP would contact the national infrastructure running the MARS middleware using the SP AT Interface. Since an Austrian citizen made the request, the VIDP would route the request to the MOA-ID connector.

Austria to Spain: Imagine a Spanish citizen wanting to access an Austrian SP. Once again the SP would make the request to the MARS middleware through the SP AT interface, but this time, because the IdP is in Spain (which uses a centralized infrastructure and relies on the PEPS model), the VIDP would route the request to the C-PEPS Connector, that would then contact the C-PEPS (acting as a stub) in Spain, that would then act as intermediary for the Spanish IdP using the SAML protocol.

Spain to Spain: For a national request within Spain, the national SP would contact the national S-PEPS to route the request to citizen's respective country (Spain in this case). The SP would create the SAML Request and send it to the S-PEPS running the VIDP, which would route it to the C-PEPS connector, redirecting it to the C-PEPS. To note that in this case the S-PEPS and the C-PEPS are essentially the same middleware running on the same machine, as the middleware would act as both entities in the case of a non cross-border use case.

Spain to Austria: Finally, in a Spain to Austria scenario, picture an Austrian citizen trying to access a Spanish SP would trigger the SAML Request to the national S-PEPS. Because Austria has a decentralized model, the Spanish S-PEPS would act a proxy/stub to the MOA-ID. Once the request reaches the VIDP through the V-PEPS interface, the MOA-ID connector would be used to directly connect to the Austrian IdP .

Note that in the above examples the route that the SAML Response would take has been omitted, as it would be the inverse of the request. The return flow has also been omitted from the diagram in Fig. 6 for simplification.

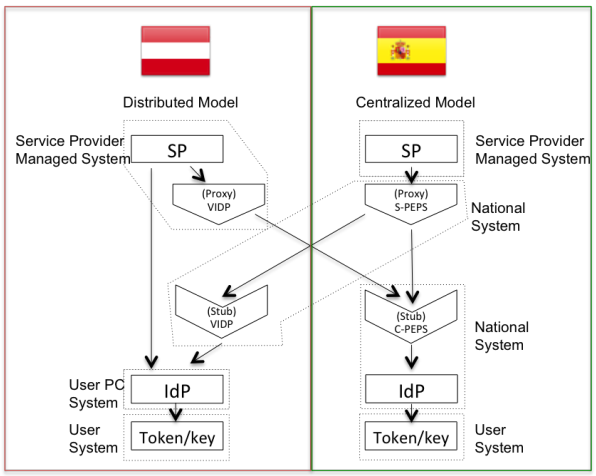

Figure 4: Mix Model Distributed and centralized models integrated by deploying the stub/proxy of the distributed model in the centralized systems.

2.4.2 Quality Authenticator Model

To create a single authentication and SSO area within the EU, mechanisms to control the quality and assurance of a given authentication need

to be established. The variety of technologies used within the countries with functional IdM infrastructures leads to different quality levels of authentication and quality assurance depending on the technology used. For example, a username/password tuple is the weakest type of authentication, while cryptographic physical tokens (such as the various tokens with PKI features in Austria, or the ECC in Estonia/The Netherlands or Portugal), provide the strongest authentication method.

In such a diversified environment, it is important for SPs to be able to determine what is the strength of the technology that was used to authenticate a given user before authorizing access or consuming his attributes. This way, the SP can estimate the risk of giving access to a sensitive service depending on the strength of the authenticated technology used.

To solve this problem, the Quality of Authentication and Assurance (QAA) levels were designed as an important part of STORK. There are four QAA levels, 4 being the highest (authentication using an ECC) and 1 the lowest (authentication through a username/password tuple).

Because the attributes retrieved from the STORK infrastructure may be released from several different IdPs, they suffer the same assurance problems as the authentication methods themselves, as some IdPs may be more reliable than others, and the individual attributes may also have different assurance levels. For example, a university may release the address of a student, however, this information may be outdated, making it unreliable and lowering its assurance. If, however, this information was provided by a government IdP that retrieved the attribute from the national residents registry, it would have a higher level of assurance. The same may not apply to a student's diploma though, because this information is managed by the university itself, giving it maximum authority over its disclosure, which gives it the highest level of attribute assurance.

To enable IdPs to specify the assurance level of the individual attributes, the Attribute Quality of Authentication and Assurance (AQAA) were designed, based on the QAA, and having the same four levels of assurance that directly map to the QAA levels. This permits that when an authentication is combined with multiple attributes, it is possible to make a separate statement of the AQAA level for all the individual attributes as well as the QAA of the eID level, giving SPs fine grained control over under what circumstances they would authorize a user or accept his attributes for a given operation, letting them have some control over the associated risk. In situation where a single AQAA level statement is to be applied to a set of attributes, the lower AQAA level within the set is to be applied. This is because the weakest AQAA in the set defines its trustworthiness as a whole.

3. Implementation

3.1. Trusted Attribute Display Service

The Trusted Attribute Display Service (TADS)[2] is a web application that relies on the STORK infrastructure (using the Pan-European Proxy Service (PEPS) model) for authentication and attribute retrieval. The goal of TADS is to give European citizens a tool that allows them to materialize on paper the digital attributes that they retrieve from STORK, enabling the dematerialization of identity assured credentials with legal validity in a MS, and materialization into another, thus establishing a trust circle between digital and printed documents. TADS is to be deployed as a trusted SP, where users can retrieve and gather several attributes (primarily academic) from different APs, and aggregate them into a signed, printable Portable Document Format (PDF) document. They can then present a printed or digital version of the document to the verifier (whom is interested in verifying the document validity), which will access the TADS service for validation. If the verifier receives a digital version of the document, he only needs to check the validity of its digital signature. A Quick Response (QR) code on the printed documents allows the verifier to scan them using a webcam from the TADS verification service page and retrieve the digitally signed PDF document, thus allowing him to validate the document and the contained credentials. Figure 5 shows an overview of the involved entities and interactions in an exemplified usage flow.

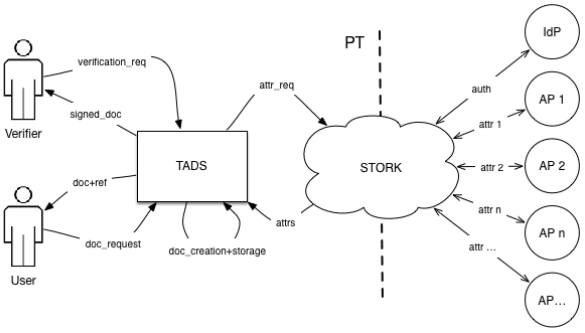

Figure 5: TADS interaction flow overview.

3.1.1 Software Architecture

The software architecture of TADS is presented in Figure 6. Node.js applications are composed by modules provided by the community that are managed through the NPM (Node Package Manager)

[2]https://github.com/STORK2/TADS

tool. The most relevant modules are represented in the figure. We will briefly describe those modules and other components.

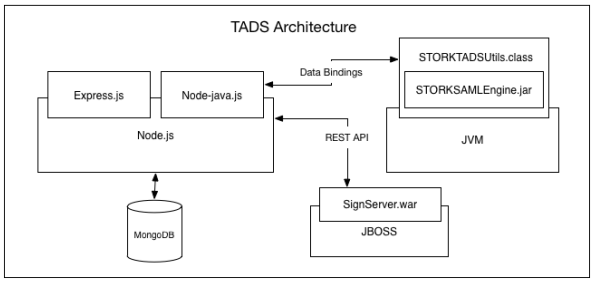

Figure 6: Architecture of TADS.

Simple and Complex Attributes: TADS allows the user to create a PDF containing either simple or complex attributes retrieved from the infrastructure. Simple attributes are just key-value pairs, making it easy to interpret and insert into the PDF. Complex attributes are more complex XML structures that require additional processing.

To uniquely identify the users in the system and keep track of their documents, the *eIdentifier* attribute is used, as it uniquely identifies a citizen within the infrastructure. For this reason, the attribute is always requested as mandatory alongside authentication.

Express.js is a framework that provides Application Programming Interfaces (APIs) to develop web applications on top of Node.js following the Model-View-Controller (MVC) architectural pattern. It gives the developer the necessary tools to easily manage the web application routing and middleware components that handle things such as the views template engine, cookie and session management and serving static files.

MongoDB is a NoSQL document-oriented database that follows a schema-less model where documents are stored in structures that are very similar to JavaScript Object Notation (JSON).

STORK SAML Engine Library is a Java library that provides an API for SPs to create and validate SAML Requests and Responses in the context of the STORK project. The SP needs to configure the SAMLEngine in collaboration with the national entity responsible for the deployment of the PEPS, as it is necessary that the PEPS trusts the SP, meaning that the SP public key certificate used to sign the SAML Request needs to be added to the PEPS keystore, among other parameters regarding namespaces and the SP identifiers that are used to log the SAML transactions.

STORKTADSUtils class: As seen in Figure 6, the Node.js application doesn't invoke the STORK-SAMLEngine library directly, but instead uses a wrapper class created specifically for Trusted Attribute Display Service (TADS). This is to expose a simpler API to interact between Node.js and the library.

Node-java Module: Because the provided SAMLEngine library is in Java, this node-java module is necessary to be able to bridge the Node.js application and the library API. The module allows a Java class to be instantiated from the Node.js application and creates the necessary data bindings to directly interact with it.

SignServer[3] is an open-source application framework written in Java that allows to perform a wide range of cryptographic operations for other applications. For TADS, it acts as a local server that exposes a REST API that is invoked by TADS to add a cryptographic signature to the user created documents.

3.2. e-Learning Application

The eLearning pilot at IST is integrated with the institution's academic web platform (FenixEdu-Academic) and the Central Authentication Service (CAS) deployment. The system will allow a foreign student to access an online course on the Fenix platform by authenticating at the IdP of his country of origin and obtaining the necessary credentials that prove his enrolment in the course through the STORK infrastructure, granting him access to its contents and resources.

3.2.1 Software Architecture

Figure 7 shows the overall architecture of the application.

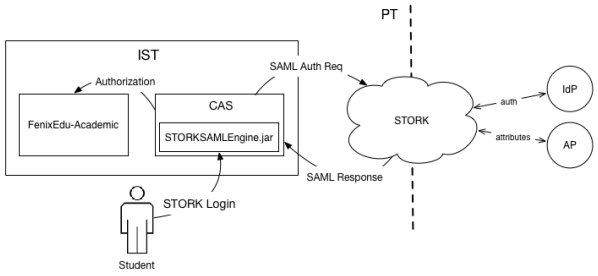

Figure 7: eLearning architecture overview.

The involved entities are the following:

- Fenix-edu Academic platform: This is where the course will be created and hosted. The platform provides the tools to maintain and manage all the necessary information to run the course and its contents;
- Central Authentication Service: This is the SSO server deployed at IST that is used for

[3]www.signserver.org

access control and authorization for all the services offered by the informatics department, including Fenix. This means that it has to be connected to the STORK infrastructure to make it accept STORK attributes as credentials and grant the user access to Fenix and its eLearning course;

- The national PEPS and the STORK infrastructure.

Academic is one of the modules of the FenixEdu software platform and it allows to manage curricular units within the university. The platform is accessible from a web browser by students and teachers giving them access to interfaces that allow them to see and manage the courses contents.

For the eLearning pilot, a course called STORK101 has been created on the platform, accessible to a student of another institution by presenting the necessary STORK attributes and identity credentials.

To allow Academic to grant access to an external eLearning student via STORK without changing the core business logic of the platform, two steps were required:

1. Develop the necessary changes for the CAS server to be able to create and validate SAML Requests and SAML Responses, respectively;

2. Develop the necessary auxiliary classes and an API on Fenix to allow the creation of a student on-demand, so that he can be identified by the system and logged into a session.

3.2.2 CAS Integration

The CAS server does not offer an out-of-the-box solution that works for authentication through the Security Assertion Markup Language (SAML) assertions used within the STORK context. Because of this, it was necessary to develop two different authentication flows:

The SAML Request flow: This is the web flow that allows a user to begin the authentication process via the STORK infrastructure. This implies the addition of an option on the CAS login page that users can select if they want to authenticate using their European eID credentials. Moreover, because the S-PEPS needs to know to which country to redirect the SAML Request, a WAYF page where the user selects his country of origin was added to the CAS user interface. Finally, a controller that constructs the SAML Request and sends it to the S-PEPS by redirecting the user's browser has also been created.

The SAML Response flow: Is the web flow that validates the SAML Response when the user is redirected back to the CAS server after going through the STORK authentication and attribute retrieval flow. The SAML Response comes within the HTTP request as an URL parameter. The necessary web flow was set up so that if a SAML Request parameter is detected, it is routed to the custom authentication handler created for the SAML validation.

3.2.3 FenixEdu-Academic Integration

Creating and enrolling a student on-demand in the STORK course on Fenix requires a number of steps. Some of these steps are workarounds, because the platform doesn't support capacities and on-demand enrolments due to its business rules being aligned with the processes of the institution.

Authentication is the responsibility of CAS, which includes authentication through a SAML Response validation. However, CAS needs to be able to create a new student "on demand" on Fenix, meaning that it has to be able to invoke an API developed for this purpose. The API follows a RESTful architecture and is secured by username and password for access control and HTTPS. Two distinct functions are provided by the API:

Student enrolment: Receives a JSON object containing the following parameters: *eIdentifier*, *givenName*, *surname*, *gender*, *eMail* and *textResidenceAddress*. The function searches for a student entry that has an IdDocument whose value matches the received *eIdentifier*. If a match is found, the ISTID is returned as a string in the entity body HTTP field with code "200 OK", otherwise, the student is created using Academic domain methods providing the remaining attributes as parameters. Because no actual value is retrieved from STORK for the IdDocument object associated to the Person, the *eIdentifier* is stored as the IdDocument value of the Person object created for the new eLearning student so that it can be identified and retrieved. Finally, a new ISTID is returned in the HTTP entity body response with code "201 CREATED".

NIC lookup: Receives a JSON object containing the NIC of the student. A domain function is called to search and retrieve a Person object providing his NIC value. If a match is found, the corresponding ISTID is returned in the HTTP entity body with code "200 OK", or an empty string with HTTP code "404 NOT FOUND" otherwise.

Once the student has authenticated at the CAS server and is redirected to the Academic platform, a filter intercepts the HTTPS request and looks for the generated ticket in the parameters. Then, the CAS client establishes a secure connection with the server to validate the retrieved ticket and upon successful validation, the client receives the principle with the ISTID of the user. Now that Academic

can identify the authenticated student, a browser SSO session is established, so that during the validity of the current session the user doesn't have to sign in again to access his personal area on the platform. In the case of a STORK student, he will be able to access the contents of eLearning course, while IST students can access the courses they are currently enrolled in.

4. The Attribute Provider

The AP has the main purpose of serving student academic related attributes upon receiving a request from the STORK infrastructure.

4.0.4 Software Architecture

The AP has has been developed in a separate module that is injected as a dependency on the Academic platform, and is included in the assembled WAR file upon compilation. This tight integration into the Academic platform was necessary because the students information stored in its database is accessible through its domain level methods. The AP endpoint is automatically available upon deployment of the Fenix platform.

The AP receives requests directly from the PIAP (see section 2.3) via the SOAP web service, being completely agnostic of the SAML messages and protocol, however, the attributes in the response have to be compliant with the STORK specifications.

4.0.5 Implementation

The PIAP is the Portuguese IdM interface between the STORK infrastructure and the national APs. The protocol used for the communications is the SOAP web service standard whose interface is defined by a Web Services Description Language (WSDL) file that specifies the format and content types of the request received by the APs and its response to the PIAP. The communication model is asynchronous, meaning that the APs must respond to the PIAP immediately after receiving a request, acknowledging it with a HTTP "200 OK" message. After the acknowledge is sent, the request is processed and once all the attributes have been constructed, another web service provided by the PIAP is invoked from the AP, with the response in the request parameters.

5. System overview

Now that we have explained all the components of the academic pilot, we will give a macro view of how it all links together. Figure 8 shows the overall components and their interactions.

Both the TADS and eLearning may start a STORK authentication and attribute retrieval flow by forwarding the SAML Request to the national PEPS server of the national IdM infrastructure.

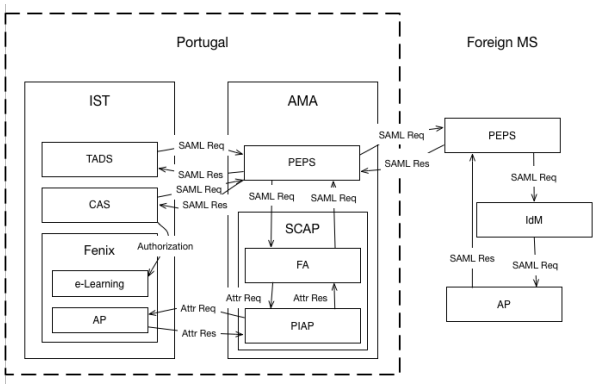

Figure 8: Illustration of the overview of all the elements of the pilot.

There, the requests follow different paths depending on the country of origin of the user, reaching the respective IdP and possibly AP(s), obtaining the response through the reverse path and obtaining the assertions.

6. Results

At the time of writing of this document, the pilot is still ongoing, but cross border tests haven't been possible yet due to delays in the deployment of the national STORK infrastructure. All the pilots were successfully deployed in a test environment using a local deployment of all the different components of the PEPS model to simulate the whole route the SAML messages would go through, as well as authentication and attribute retrieval. Because of this, no user tests with foreign partners were possible using the infrastructure.

6.1. TADS

The TADS server is currently deployed[4] and is able to request some attributes from the national IdM. So for, only attributes that are available on the national citizen database is retrievable (e.g: NIC, NIF, givenName, surname, age, gender). The national APs still aren't reachable, meaning that the full chain that a request¡-¿response flow would go through nationally to retrieve attributes from other institutions (e.g:TADS↔PEPS↔FA↔PIAP↔AP) hasn't been tested yet. In the test environment deployment of the infrastructure, we were able to successfully create documents containing simple and complex attributes.

6.2. AP

The AP was successfully integrated with the national IdM, and testing from the national entity managing it successfully retrieved test attributes. However, as mentioned above, the PIAP is still be-

[4] tads.tecnico.ulisboa.pt

ing integrated with SCAP and the national PEPS, and because of this it isn't possible to test the retrieval of academic attributes through TADS.

6.3. eLearning

The eLearning pilot was also successfully deployed in a test environment, creating and enrolling a student presenting STORK credentials and attributes to the CAS server and being able to access the STORK101 course on Fenix. For the same reasons described above and because the deployment of the national PEPS is still not always available, cross-border tests weren't possible so far.

7. Conclusions

This work proposed and developed software solutions for the Portuguese STORK Academic LSP, which consisted in the integration of both the developed SP (TADS) and the pre-existing Fenix SPs with the STORK infrastructure. The development of a SP from square one and the integration with the infrastructure of previously existing systems (Fenix and CAS) presented different challenges.

These challenges allowed for a better understanding of the implication of such an infrastructure at an European level and the effort required for SPs and APs to embrace it, as well as the difficulties faced into its integration with already existing IdM solutions at a national level. As we were overcoming these challenges, useful information was gathered for the pilot and its partners.

We succeeded in the implementation of all the requirements and created the necessary conditions for user testing of the national STORK infrastructure. It is clear however, that from the gathered experience and information, while achievable, full interoperability is still far from being optimal, as usability is still poor from the users perspective and it's somewhat difficult to teach them the benefits of strong authentication, as well as how the STORK usage flow protects their data from misuse and disclosure to third parties.

References

[1] B. P. Bruegger, D. Hühnlein, and M. Kreutzer. Towards global eid-interoperability. In *BIOSIG*, pages 127–140, 2007.

[2] R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. *Security & Privacy, IEEE*, 6(2):24–29, 2008.

[3] J. Jensen. Federated identity management challenges. In *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, pages 230–235. IEEE, 2012.

[4] A. Jøsang and S. Pope. User centric identity management. In *AusCERT Asia Pacific Information Technology Security Conference*, page 77, 2005.

[5] A. Kalja, J. Pold, T. Robal, and U. Vallner. Modernization of the e-government in estonia. In *Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET'11:*, pages 1–7. IEEE, 2011.

[6] V. Koulolias, A. Kountzeris, H. Leitold, B. Zwattendorfer, A. Crespo, and M. Stern. Stork e-privacy and security. In *5th International Conference on Network and System Security*, pages 234–238. IEEE, Sept. 2011.

[7] H. Leitold and A. Tauber. A systematic approach to legal identity management–best practice austria. In *Proceedings of the Information Security Solutions Europe 2011 Conference*, pages 224–234, 2011.

[8] T. Martens. Electronic identity management in estonia between market and state governance. *Identity in the Information Society*, 3(1):213–233, 2010.

[9] R. McKenzie, M. Crompton, and C. Wallis. Use cases for identity management in e-government. *IEEE Security & Privacy*, 6(2):51–57, 2008.

[10] G. Moniava, E. Verheul, and L. Schoenmakers. *Extending DigiD to the private sector (DigiD-2)*. PhD thesis, Masters thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, 2008.

[11] T. Rissanen. Electronic identity in finland: Id cards vs. bank ids. *Identity in the Information Society*, 3(1):175–194, 2010.

[12] T. Rössler. Giving an interoperable e-id solution: Using foreign e-ids in austrian e-government. *Computer Law & Security Review*, 24(5):447–453, 2008.

[13] J. Torres, M. Nogueira, and G. Pujolle. A survey on identity management for the future network. *Communications Surveys & Tutorials, IEEE*, 15(2):787–802, 2013.

[14] B. Zwattendorfer, I. Sumelong, and H. Leitold. Middleware architecture for cross-border eid. In *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, pages 303–308. IEEE, 2012.