A Model for Authentication Credentials Translation in Service Oriented Architecture*

Emerson Ribeiro de Mello**[1,2], Michelle S. Wangham[3]**, Joni da Silva Fraga[1]**, Edson T. de Camargo[1], and Davi da Silva Böger[1]

[1] Department of Automation and Systems
Federal University of Santa Catarina
Florianópolis, SC - Brazil
[2] Federal Institute of Santa Catarina
São José, SC - Brazil
[3] Embedded and Distributed Systems Group
Univali - São José, SC - Brazil
{emerson,fraga,camargo,dsborger}@das.ufsc.br,wangham@univali.br

Abstract. Due to the increasing number of service providers, the grouping of these providers following the federation concept and the use of the Single Sign On (SSO) concept are helping users to gain a transparent access to resources, without worrying about their locations. However, current industry and academic production only provide SSO in cases with homogeneous underlying security technology. This paper deals with interoperability between heterogeneous security technologies. The proposed model is based on the Credential Translation Service that allows SSO authentication even heterogeneous security technologies are considered. Therefore, the proposed model provides authentication credentials translation and attribute transposition and, as a consequence, provides authorization involving different kinds of credentials and permissions in the federation environment. By making use of Web Services, this study is strongly based on concepts introduced in the SAML, WS-Trust and WS-Federation specifications.

Key words: Web Services, Security, Single Sign-on

1 Introduction

The demand for sharing information among different administrative domains in a transparent and secure manner and the need for establishing trust relationships are both up-to-date requirements mainly within the Internet context. Clients and service providers are entities that are present in this domain and that seek some kind of interaction. In this scenario, there are three barriers to overcome: (1) the heterogeneity of the security infrastructures, present in the many corporate

* This work has been developed within the scope of the "Security Mechanisms for Business Processes in Collaborative Networks" project (CNPq 484740/2007-5).
** Supported by CNPq - Brazil

domains, (2) the establishment of trust relationships among unknown entities, and (3) the management of identities carried out not only by service providers but also by clients.

In distributed systems, traditional authorization models are based upon an authentication authority to mediate the trust among unknown parties (Trusted Third Party). Therefore, the interactions among distinct parties are done through the presentation of credentials issued by an authentication authority known by the parties involved. In more complex environments such as the Internet, this model of simple interaction is limited, since each domain has its own policies, security infrastructures, and also a particular way of managing the principals'[4] identities. In other words, each domain runs its authorization controls according to its local policies, without considering the attributes of other domains, and thus previous authentication is usually required in the domain itself.

Single Sign On (SSO) was developed to simplify the interactions among clients and different service providers. Under this approach, the client authenticates only once and makes use of this authentication in the interactions with other service providers. In open systems, it is desirable that each domain have the freedom to choose and adopt an authentication mechanism. However, the difficulty lies in assuring the SSO's interoperability among these domains, since the authentication information is no longer understood by all the entities present in the domains with different technologies. This problem motivated the present work.

This work aims to describe a model that enables the authentication with SSO, that is, the translation of authentication credentials, even when dealing with administrative domains with different security technologies. In this model, a principal can access resources in domains with security technologies, which are different from those in its source domain, by using the credentials provided in its own domain. In the entire system, access authorizations to resources depend on this authentication, which sensibly diminishes the data flux between the domains and the management of these data in the system as a whole.

According to [1], in the management of federated entities, each company builds a domain, in which its service, identity and credential providers are present. The deals established among domains can allow local identities of a domain to be accepted in the other domains participating in the agreement. Hence, a user with an identity registered in its domain can access resources in other domains of the federation without opening a new register or identity. In this study, the management of federated identities significantly contributes to the translation of the principals' authentication credential.

The Web Services technology, based on Service Oriented Architecture (SOA), is one of the promising solutions to the integration of administrative domains in distributed systems [2]. Although this technology contributes to overcome the challenges involving authentication and authorization in distributed applications, the large number of specifications and standards aiming at security not only imposes a certain level of complexity but also hinders its wide adoption [3].

[4] Users, processes or machines authorized by the systems policy.

With the support of Web Services security technologies, this article introduces a model to authentication and authorization in service oriented distributed systems. Based on the concept of federated identities, this model performs the translation of a principal's authentication credential among administrative domains with different security technologies (e.g. X.509, SPKI/SDSI) in a transparent manner. The model also supports the transposition of client attributes. In order to prove the applicability of the model, a prototype was implemented and integrated into a distributed application - an information portal in the area of entertainment.

2 Security in Web Services

The Web Services architecture is linked to XML and to the security extensions of this standard, defined by W3C, such as **XML-Signature**[4] and **XML-Encryption**[5]. These specifications allow for the representation of digital signatures and data encryption in the XML format, while the signed and/or encrypted data may be XML documents or not. These mechanisms make end-to-end security possible for Web Services using XML for data exchange and storage.

The **XACML** (eXtensible Access Control Markup Language) specification [6] describes both a language to express policy rules and a request/response protocol for decisions on access control. This specification uses two elements to implement the access control in distributed environments: PEP (Policy Enforcement Point, responsible for mediating and performing the access, and PDP (Policy Decision Point), which is called by the PEP to perform the policy processing and to decide, based on subject and resource information, whether or not access will be granted. Two more entities defined in this specification are PIP (Policy Information Point) and PAP(Policy Access Point). The former is responsible for retrieving information on the subject, environment and resource, while the latter is in charge of the access to the policy resource.

The **SAML** (Secure Assertion Markup Language) specification[7] is a security infrastructure projected to express information [5] about authentication, authorization and attributes of a given subject. Also, it allows for the exchange of this information among business partners. The SAML does not provide the authentication itself, but ways to express authentication information. The client can authenticate only once and use this same authentication in the other affiliated domains (Single Sign On). These characteristics make the SAML standard an important foundation to the management of federated identities in the proposed model.

WS-Security [8], the main security specification to Web Services, is based on XML-Signature[4] and XML-Encryption standards[5] to provide safe message exchanges. The specification aims at flexibility, and thus enables the use of a variety of security mechanisms. More specifically, this technology provides support

[5] With the objective of being interoperable, the information is expressed in security assertions.

to different kinds of security credentials[6], which enables a client to use multiple credential formats for authentication and authorization, multiple formats for signature and multiple technologies for data encryption. These characteristics are very important in order for interoperability between security technologies of different administrative domains to be reached.

The **WS-Policy** [9] specification provides an extensible and flexible grammar, which allows for the expression of competencies, requirements and general characteristics of Web Services. It defines a framework and a model to represent these properties as policies[7]. The structure of policy assertions, such as the kinds of credentials required and the encryption algorithms supported, are defined in the WS-SecurityPolicy specification [10]. The WS-Policy does not describe how these policies are published or how they are attached to a Web Service. The mechanisms to attach the policies to XML, WSDL and UDDI elements are defined in the WS-PolicyAttachement specification [11].

The **WS-Trust** specification [12] defines services and protocols and aims at the exchange of security attributes (e.g. SAML assertions), in order to allow for the communication between different administrative and security domains. The WS-Trust defines a trust model in which a client without the credentials requested by the service provider can request the credentials to an authority which owns them. This authority is named Security Token Services (STS). This service forms the foundation for the establishment of trust relationships and is responsible for issuing, exchanging and validating the credentials. Nevertheless, this specification does not tackle how to translate the information contained in the principal's security credentials, when dealing with domains with different technologies.

The WS-Security, WS-Trust and WS-Policy specifications provide a foundation to the **WS-Federation** specification, which describes how these specifications are combined in order to allow for the construction of domains of trust. The foundation for the establishment of trust in the WS-Federation is the STS Service. However, the WS-Federation adds to this service the Identity Provider (IdP) and the Attribute/Pseudonym service functionalities, combined or not in a single entity. An identity provider functions as an authentication service, in which the domain members authenticate and make use of that authentication. The Attribute/Pseudonym service enables the protection of the user's privacy, through pseudonyms[8].

The WS-Federation fits into the centralized model for the management of federated identities, whereby only one identity and credential provider is used by all the service providers of the federation. In this model, a user can access all the services present in the federation through the use of a single identifier. The model resembles the federated identity model; however, unlike its counterpart, it dispenses with credential mapping [1]. This constraint ties the client to a single

[6] Other security credential formats are UserNameToken, X.509, Kerberos and SAML.
[7] It introduces ways to express quality of service policies related to security and confiability.
[8] A random opaque identifier, which is not discernible by another entity.

identity provider in the federation. In a fashion similar to WS-Trust, the WS-Federation does not signal how the information within the original credential can be translated in the presence of different security technologies.

3 A Model for Translation of Authentication Credentials

After having defined the concepts of domain and federated identities, this section introduces the model in a general sense at first, based upon an IETF proposition [13] and upon an SAML specification proposition. Next, it describes the model in a more concrete manner, assuming Web Services specifications. The challenge of the proposed model is to handle a large set of security specifications for Web Services, which are still under development, for the most part, or have recently been launched.

3.1 Security Domains and Federated Identities in the Proposed Model

In the proposed model, each administrative domain groups clients and service providers according to their underlying security infrastructures. Domains based upon different technologies determine different security controls for the protection of their resources. The premise assumed in the model is that, in the security controls for the domains clients must prove their identities before an authentication authority and, based on this authentication, a verification is made of the associated rights to access the system resources.

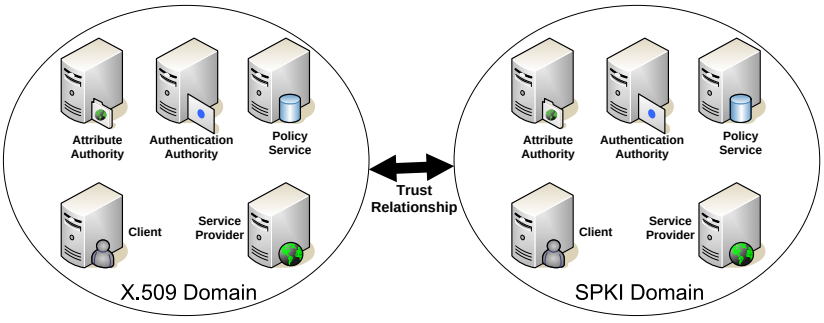

Fig. 1. Security Domains

In a security domain, the authentication and attribute authorities and the policy service are autonomous entities that manage their respective database for queries performed during authentication and authorization operations (see Figure 1). The authentication authority contains information on registers of clients and domain service providers. The attribute authority is responsible for

managing the attributes informed by the users during the registration process in the domain; for instance, email, credit card, ZIP code, etc. The policy service centers the authorization policies for the domain, and thus functions only as a repository.

Trust relationships are the basis so the identities and the credential attributes can be recognized in the different participating domains, and thus allow for the translation of authentication credentials when facing different security technologies, concept introduced in this propose.

Within the context of federated identities, a client may be affiliated to more than one domain, with distinct accounts and different attributes spread over different domains. As in studies that implement the concept of federated identities, such as WS-Federation [14], Shibboleth [15] and Liberty Alliance [16], in the proposed model, the service provider is also the one responsible for retrieving the user's information. The client provides only its identity or pseudonym and the service provider is the one who searches, among its federation partners, for the client attributes.

3.2 Model Overview

The authorization model discussed in this article is based on the model proposed by IETF [13], which describes two fundamental elements for policy management: **PEP (Policy Enforcement Point)** and **PDP (Policy Decision Point)**. PEP is an element, which controls the access to a resource, and its function is to apply policy decisions, whereas PDP is an entity, which defines the policy to be applied to the required service requisition. Fig. 2 shows the proposed model and highlights the steps of the authorization process.

Other entities were added to the IETF propositions as a starting point to the model, the **attribute** and **authentication authorities** that are called before the authorization process. As authorization verifications in the model are assumed to be local for service providers, then both PEP and PDP are supposed to have their implementations in the service providers that control the requests to those services.

Initially, the model is considered that domain clients have already been registered and that information on their attributes is available to service providers, in the attribute authority of the domain, according to a privacy policy defined by the user itself. This policy determines which attributes the user wishes to inform the providers of its domain and also providers of other domains. The client authenticates to its authentication authority and receives an authentication credential with information related to the issuing authority, the client's identity, the expiration date, etc. This credential must then be presented whenever the client wishes to access resources in the service providers located at the domains with which the client's domain has trust relationships.

As the proposed model is based on the security standards of the Web Service technology, it assumes a service-oriented format. The attribute and authentication authorities are the Security Token Service/ Identity Provider (STS/IdP) and the Attribute and Pseudonym Service (APS), respectively (see Fig. 2). This

proposal combines Security Token Service (STS) and IdP (Identity Provider) into a single entity, known as STS/IdP that is responsible to issue, validates and exchanges identity and authentication credentials. Both user's privacy and attribute provide are functions performed by Attribute and Pseudonym Service (APS), which works together with STS/IdP. The credentials issued by STS/IdP are **SAML authentication assertions** and the attributes retrieved from APS are **SAML attribute assertions**.

The security of SOAP messages is ensured through the *WS-Security* specification, which provides end-to-end security and avoids attacks such as replay and man-in-the-middle, using encryption and digital signature mechanisms in compliance with the requirements defined in the quality of protection policy expressed in compliance with the WS-Policy specification[9].

In a policy server, the authorization policies that protects the resource are expressed in XACML and the messages exchange between PEP and PDP follow the XACML specification. XACML was chosen since this de facto standard was specifically designed to represent access control policies. When PDP queries the authentication authority, the STS/IdP, to validate an authentication assertion received, it does so through the protocol defined in the WS-Trust specification. PDP's interactions with the STS/IdP of its domain are justified by the fact that the service provider may not support the format of the received credential and might need this credential to be translated into the technology it supports. It should be highlighted that, in the proposed model, STS/IdP acts in a similar manner as the PIP (Policy Information Point) of the XACML standard.

The challenges of the model appear when an access request involves different security domains, that is, the model must ensure interoperability among different administrative domains.The integration of different underlying technologies requires a standard language to represent security attributes, either authentication or authorization ones. As aforementioned The WS-Trust and WS-Federation specifications do not tackle how to translate the information contained in the principal's security credentials, when dealing with different technologies.

In this study, we chose to describe a case composed by two security domains, both based upon PKIs. Besides, clients and service providers were considered to be able to work only with the security technology employed in their domain, as the client domain is based on SPKI, and the service provider domain is based on X.509. However, the model is generic enough to allow for the use of other technologies.

In the proposed model, the translation of authentication credentials and the attribute transposition to different security domains is based on the Attribute/Pseudonym Service (APS), on the Security Token Service (STS/IdP), and also on a new service, the Credential Translation Service (CTS), introduced in this article.

[9] It should be noted that the security of sensitive information stored in the clients is beyond the scope of this article.

The Attribute/Pseudonym Service (APS) is directly linked to the STS/IdP and has two purposes: providing attributes associated with a principal[10] and managing a pseudonym system, which allows a principal to authenticate and to use this authentication in the remaining domain entities without having its real identity revealed. For the proposed model, the STS/IdP of the domains establishes trust relationships, and thus forms "federations". Then, in this environment, there is a need for the standardization of the attributes provided by each Attribute/Pseudonym service. In the literature, many studies have been concerned with defining a standard set of attributes needed for a federated environment [17–19]. Among them, approximately 40 attributes have been defined as *common identity attributes*[20]; 6 of which are highly recommended, 10 are suggested, and 25 are optional. In this proposal, we adopted a standard set of attributes composed of 6 highly recommended attributes and of 10 suggested attributes, as shown in the [20] document. This set is enough to allow credentials of an SPKI domain to be converted into X.509 certificates. Nevertheless, it can also be expanded in order to accommodate other security technologies, such as Kerberos tickets and biometric credentials.

All the entities of the domain trust their STS/IdP and thus a client identification in the presence of a service provider will be attested through an SAML authentication assertion issued by the STS/IdP. In the model, we consider the existence of trust relationships between STSs of different domains, given that these relationships allow the assertions issued in a domain to be valid in the remaining domains. In order to accomplish so, as an STS receives an SAML authentication assertion, it can ask the authentication authority that issued the assertion (the STS of the client domain) to provide additional information about this authentication so that it can assess the level of trust of the assertion.

The specification [21] defines an XML Schema for the creation of authentication context declarations - XML documents that allow the authentication authority to provide the relying party with this additional information. This information could include: the initial user's identification mechanisms (e.g. face-to-face, online, shared secret), the mechanisms for storing and protecting credentials (e.g. smartcard, password rules) and the authentication mechanism or method (e.g. password, certificate-based SSL, digital signatures). Additionally, this specification defines a number of authentication context[11] classes. Each class defines a proper subset of the full set of authentication contexts. For the purposes of the model, the X.509 and SPKI authentication context classes were used. The former indicates that the principal authenticated by means of a digital signature where the key was validated as part of an X.509 PKI, and the latter indicates that the principal authenticated by means of a digital signature where the key was validated via an SPKI Infrastructure.

[10] Respecting the privacy policy of this principal.

[11] Authentication context is defined as the information, additional to the authentication assertion itself, which the relying party may require before it makes an entitlement decision with respect to an authentication assertion [21].

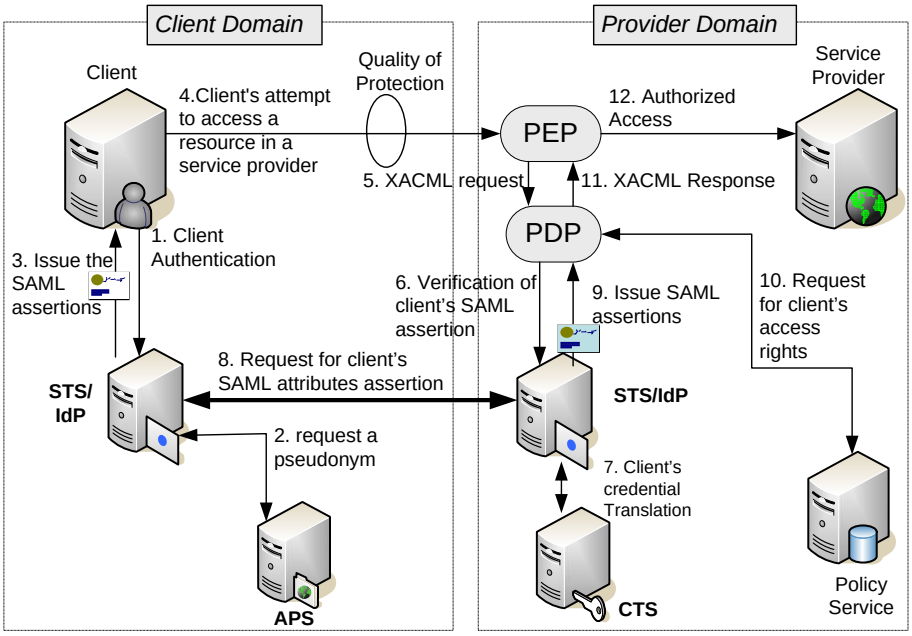

Fig. 2. Dynamics for Translation of Authentication Credentials

3.3 Dynamics for Translation of Authentication Credentials

Fig. 2 shows all the steps taken in order for a client in an domain to access resources in a service provider present in another domain, and also shows the interactions among the STS, APS and CTS (Credential Translation Service) services introduced below. To a better comprehension, consider a scenario where the client's domain is based on SPKI and the service provider's domain uses X.509.

Before a client can call the service provider, it needs to authenticate to its STS (Step 1, Fig. 2), and it may request a pseudonym to associate the authentication credential (Step 2). In Step 3, the client receives an SAML authentication assertion based on the security attributes provided by SPKI. In this case, the assertion only carries the client SPKI public key. In Step 4, the client calls the service provider and provides, together with the request, the previously received SAML assertion. The enforcement of an authorization policy starts at the service provider with the provider's PEP.

When PEP intercepts a client's request, it issues a XACML request with a policy decision request to PDP (Step 5). PEP's request to PDP defines one or more policy elements, as well as information on the desired access. In order to make the decision concerning the request, PDP may query authentication and attribute authorities, the STS/IdP and the APS, to validate the SAML authentication assertion received and to obtain the client attributes.

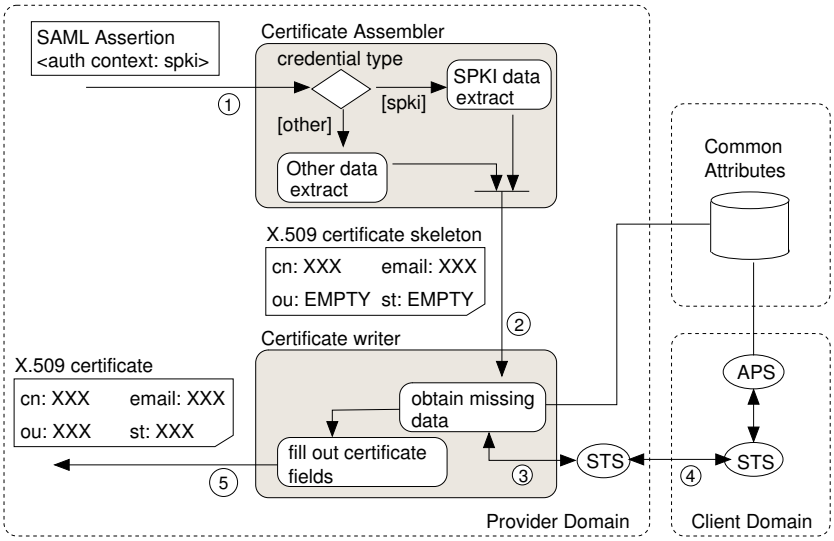

Fig. 3. Credential Translation Service

It is known that the service provider is only apt to work with X.509 credentials. Consequently, the SAML assertion received from the client is forwarded to its STS (Step 6), which is in charge of assessing the trust in the SAML assertion received and translating the assertion into an X.509 certificate. In Step 7, the STS calls the Credential Translation Service (CTS) (see Fig. 3) so that the CTS extracts, from the SAML assertion and authentication context, the necessary fields to compose an X.509 certificate. In this case, the SAML assertion does not show all the necessary attributes. Therefore, the CTS requests its STS/IdP to call the client's STS/IdP (which, in turn, calls its respective APS) in order to obtain the missing attributes (Step 8). Finally, the STS/IdP of the X.509 domain issues, to the service provider, (1) an X.509 certificate composed of information provided by the client's SAML authentication assertion, and (2) the attributes gathered in the client's STS/IdP (Step 9). At this point, the client is identified through a credential which the service provider understands and whose issuer (the STS) the service provider trusts. Therefore, it is the service provider's duty to ensure whether or not the client will have access to the service, based on its access control policies. Next (Step 10), PDP receives the authorization policy that protects the resource. After these queries, in Step 11, PDP returns the policy decision and PEP applies it, accepting or denying access (Step 12).

3.4 Credential Translation Service

The use of SAML assertions in tandem with trust relationships allows for the translation of authentication credentials, that is, a client that has authenticated to domain A can use this authentication information to access resources of a

service provider present in domain B. However, as aforementioned, clients and service providers are only able to operate with the underlying security technology of the domain and, in case the client and service providers use different technologies, there appears the need for a way of mapping the authentication information from one domain to another.

A solution to this problem lies in the use of the Credential Translation Service (CTS), which aims at the extraction of SAML authentication assertion information in order to compose a new authentication credential so that it can be understood by the domain entities (step 1 in Fig. 3). Therefore, a client making use of the SPKI technology will have its authentication attributes converted into SAML assertions which, when received by a service provider using X.509, will be converted again into this provider security technology (step 2 in Fig. 3). This functioning ensures that clients and service providers keep their security features and thus let the CTS in charge of translating the attributes of a technology into another.

The use of X.509 and SPKI authentication context classes, as defined in [21], aids the translation process of authentication credentials, since the CTS makes use of this information to perform the mapping of the SAML assertion received at the credential required by the provider (for instance, X.509 or SPKI). Nevertheless, information present at authentication assertions may still not suffice to the credential's translation. For instance, in a service provider in the X.509 domain, a client authentication is carried out by means of digital signatures, whereby the key was validated as part of an X.509 PKI. Additionally to the client public key, it is known that X.509 certificates carry other information such as organizational unit, city, etc. Also, it is common knowledge that, in the SPKI, this information is absent and thus the conversion of an SAML assertion issued in a SPKI domain into an X.509 certificate depends on the presence of all the necessary certificate's attributes. It is at this point that the Attribute/Pseudonym Service is needed (step 3 and step 4 in Fig. 3).

4 Implementation

In this section, we discuss the implementation of the proposed model prototype. In addition, we address the performance of our prototype implementation. Finally, we describe the integration of the prototype into a entertainment portal.

4.1 Implementation Details

A prototype was implemented in order to attest the flexibility of the proposed model and also the feasibility of its use in distributed applications based on a service-oriented architecture. Figure 4 shows the prototype's architecture. For the implementation Java was the programming language used and as an application server Apache Tomcat. Apache Axis 1.4 was used as implementation SOAP. Other open source libraries were adopted to compose the prototype, they are:

WSS4J library[12], XML-Security library[13], SunXACML[14], OpenSAML [15] and SDSI1.2 library [22].

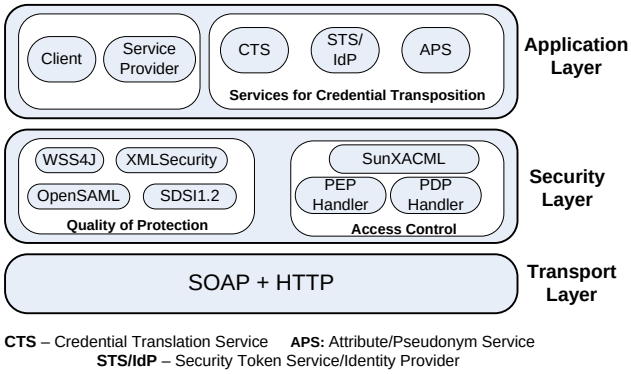

Fig. 4. Prototype Architecture

The prototype implemented consists of two security domains: one based on X.509 PKI, and other based on SPKI. One of the reasons these PKIs were chosen was that the XML Signature specification provides support for both. However, OpenSAML, XMLSecurity e WSS4J libraries provide support only to X509. The WSS4J library had its STS extended, having the capability to issue authentication credentials by using the SPKI infrastructure. The OpenSAML library was extended in order to sign SAMLTokens [23] with SPKI keys, because the current version of OpenSAML only supports X.509 keys. Finally, some extensions were implemented in the XML Security library, in compliance with the recommendations for extensions defined by Apache Foundation, in order to define which SPKI/SDSI elements can be inserted into an XML signature.

Each STS/IdP deals with security technology of its domain, as a consequence, for each technology was necessary to define an implementation of STS/IdP. In the prototype, the STS/IdPs must to process three requests types(see Figure 5): (**1**) member's authentication requests; (**2**) attribute requests in the form of SAML assertions; and (**1**)credential validation and translation request.

According to WS-Trust model, all such requests are to issue tokens. Thus, in all requests the element `wst:RequestType` in RequestSecurityToken (RST) message contains the URI field. The difference among request types is made up analyzing the other fields of the message. The requests of type 1 are made by a client to STS/IdP of its domain when it wants to access a service that requires

[12] `http://ws.apache.org/wss4j`
[13] `http://xml.apache.org/security`
[14] `http://sunxacml.sourceforge.net`
[15] `http://www.opensaml.org`

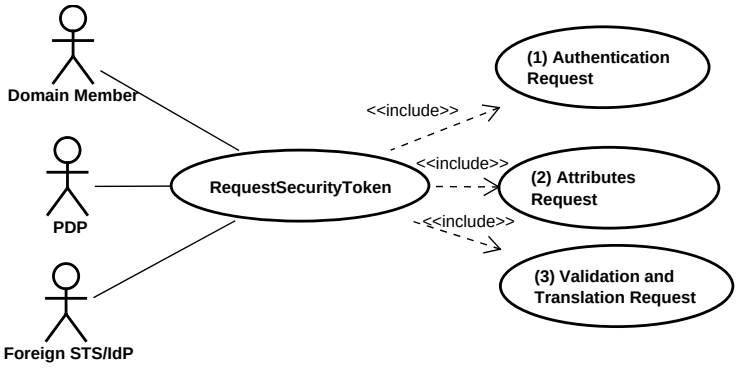

Fig. 5. Requests Handled by STS/IdP

authentication. An authentication request has the SAML assertions URI, defined in the SAML Token Profile, in the content of the element `wst:TokenType`. In those requests, STS/IdP uses the OpenSAML library to issue and to sign a SAML assertion that contains a subject (the requestor), an authentication statement and some other attributes.

The requests of type 2 also have the SAML assertions URI as the element content `wst:TokenType`, however differ from the requests of type 1 since they have the element `wst:Claims`, it contains a list of non-empty attribute names. When the STS/IdP receives such a request, it reads the list of attributes and accesses the APS to obtain required attributes. Then it uses the library OpenSAML to issue and sign a SAML assertion containing client's attributes. The requests of the type 3 are made by the PDP service provider to the STS/IdP of its domain when it wants to validate the authentication token received from the client. In those requests, the element `wst:TokenType` contains the URI indicating the type of credential-specific domain. When the STS / IdP receives a request of this type, it uses the library OpenSAML to validate the SAML assertion received in accordance with its trust over the issuer (STS/IdP). Then, it invokes CTS to know if is necessary to get more attributes of the client. If yes, then STS/IdP will request attributes of the STS/IdP that issued the SAML assertion. Finally, the CTS is required to translate the assertion to a credential in the domain technology. The credential generated by the CTS and signed by the STS/IdP is then returned to the service provider.

The services for implementation of credential translations (CTS and APS) are not implemented at libraries used, so it was fully developed in prototype. APS is an attributes repository of the domain's members and it was implemented as a STS/IdP local library. APS has two tasks in the current prototype: it retrieves attributes according to a list of identifiers; and it defines how to transport identifier inside of `wst:Claims` field.

The CTS is responsible for transforming a list of attributes in a specific credential of it domain. In the prototype, the CTS was developed as a STS/IdP local library and has two operations: one to obtain a list of attributes necessary to generate the credential; and another to carry out the translation. For each security technology supported by the prototype, the list of attributes needed and how those attributes will be processed on the credentials are different. For example, in X.509 domain, the attributes needed are the fields of a X.509 certificate, as the subject's X.500 name, the subject's public key, etc.

To facilitate the development of client applications, a library was implemented to aid the interaction between a client and the STS/IdP of its domain. A general PEP was implemented to help on the creation of new services. This PEP is able (1) to intercept client's messages, (2) to interact with PDP and (3) to enforce the PDP decision. SunXACML library was extended to carry out communication with the STS/IdP of service provider domain.

4.2 Performance

This section presents some results of tests applied with the purpose of evaluating the performance (processing time) of the prototype in three representative usage scenarios. That is, we intend to evaluate additional costs introduced by the credentials translation process and the use of security mechanism in compliance with WS-Security. We did 50 experiments using two computers with identical configuration – a 3.0GHz Pentium 4 with 1 GB RAM running Linux (kernel 2.6.24). Both computers had the Java 2 Software Development Kit (J2SDK), version 1.6.0.

In the first scenario, client and service provider are in the same domain that does not provide security mechanisms. Client sends a simple resource request to service provider. This scenario was ran across the department's network at midday and we found that, on average, it executed in 8.18 ms with a standard deviation of 3.02ms.

In the second scenario, client and service provider are in the same security domain (X.509). In this experiment, client does authentication process with its STS/IdP (steps 1 to 3 in Figure 2) and receives a respective SAML authentication assertion. In the next step, client invokes the service provider and at this point occours the enforcement of authorization policy. In this scenario, basic security properties of all SOAP messages are ensured by WSS4J library that implements WS-Security specification. On average, this scenario executed in 53.12ms over the same network with a standard deviation of 8.18ms.

In the third scenario we consider that the client's security domain is based on SPKI and the service provider's security domain is based on X.509. Client receives a SAML authentication assertion based on its SPKI's security attributes and than client involkes service provider. PEP entity intercepts client's request and it issues a XACML request to PDP entity. PDP queries the STS/Idp and the APS to validate the SAML authentication assertion and to obtain client's attributes. Finally, SAML assertion has to be translated into a X.509 certificate

because the service provider only work with this security mechanism. On average, this scenario executed in 410.74*ms* over the same network with a standard deviation of 28.07 seconds.

The difference in processing time between the first and second scenarios expresses the computational cost of digital signatures of asymmetric cryptography. Clearly, executing the authentication credentials translation process (third scenario) takes longer than using a more traditional means of SSO in homogenous security domain (e.g., based on X.509). The difference occurs, mainly due to the greater number of message exchanges. They are, overall, four invocations where all messages are signed with asymmetric cryptography. This cost is justified when the SSO authentication in heterogenous domains is desired front of different technologies that the third case provides. Additionally, the credentials translation process need to be executed only one time in a conversation, so this overhead is eliminate in the next invocations between these client and provider. Further, the prototype has not been optimized for performance. In future research is feasible implements a solution with much better performance.

4.3 Integrating the Prototype into a Distributed Application

An entertainment portal was integrated into the implemented prototype. The goal of the portal is to gather within a single interface several service providers that aim at personal entertainment - for example, movie theaters, amusement parks, video stores, theaters, etc. The open source portal used was *Stringbeans*[16]. he services offered via the Portal were implemented by providers in different domains and trust relationships among the portal domain and the service provides were established. When a client subscribes itself on the portal, this can customize their access defining what and how services are provided. Moreover, the client informs the portal his personal data. The portal register the data in the APS to permit that foreign STS/IdP access the attributes necessary to generate the credentials of this client.

In order to access the services offered via the Portal, a client must authenticate through valid login and password, over an SSL session, as shown in Fig. 6[17]. That portal, in the name of the client, asks the STS for an SAML authentication assertion. It should be noted that SPKI is the security technology supported in the portal. From this point onwards, all of this client's requests will be added to the SAML authentication assertion issued by the portal's STS, so that the client can access all the resources offered by the service providers that can be within an X.509 or an SPKI domain (see Fig. 6).

In Figure 7, the UML communication diagram illustrates the messages exchanged among the services (in the portal and service provider domains) to enforce the authentication credentials translation, according to what was implemented. It should be noted that while receiving the SAML assertion, the service

[16] `http://www.nabh.com/projects/sbportal`

[17] This occurs because the clients access the portal through common browsers and, in general, it did not supported SOAP messages

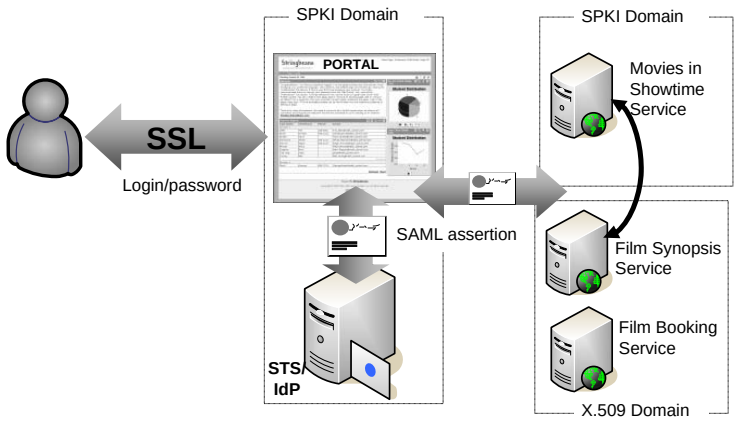

Fig. 6. Entertainment Portal

provider requests its STS to translate the assertion into an X.509 certificate. Hence, the portal is responsible for mediate the access among the clients and the service providers and the client can make use of a SSO authentication.

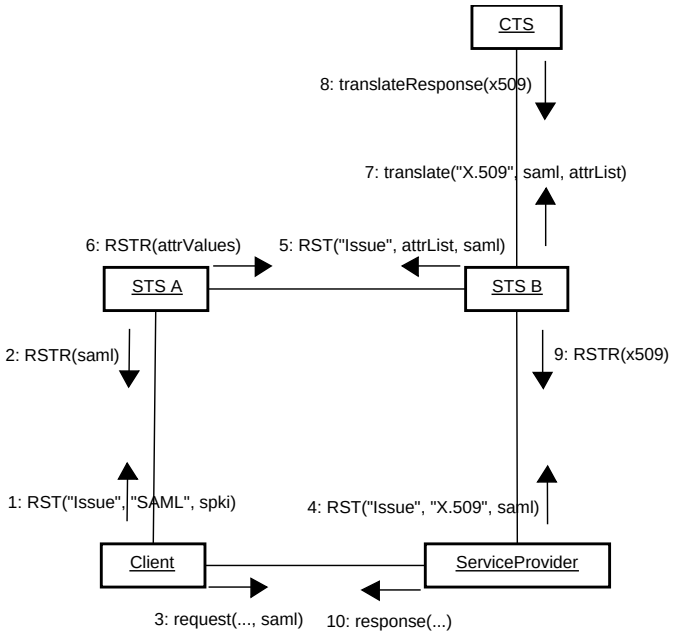

Fig. 7. UML Communication Diagram

5 Related Work

In the literature, some studies have struggled to gather a set of *Web* Service-related specifications to carry out authentication credentials translation. Most of these studies take the concept of federated identities through *Web* Services into account, but only consider the use of X.509 as security infrastructure [24, 15, 25]. An issue that still needs clarification is the nature of the interaction among domains with different security technologies.

CredEx [24] enables the safe storage of credentials and the dynamic exchange of different kinds of credentials, by using the protocol defined in the WS-Trust specification. However, the implementation only considers the previous association of a username and a password with an X.509 proxy certificate or vice-versa. Thus, the model neither encompasses different security technologies, such as SPKI or Kerberos, nor enables dynamic credential translation.

The Shibboleth framework [15] is an implementation of the browser profiles from the OASIS SAML specification, which provides a SSO service and attribute exchange from the user's home site to the site he is accessing. This framework is based on the concept of federations and also enables authentication credentials translation. Most organizations, which adopt Shibboleth use the X.509 standard for server authentication and password-based authentication for client authentication. The model proposed in this article differs from Shibboleth as it gives more flexibility to the members of the federation, which do not need to use a web browser when interacting with services. The proposed model also enables direct communication among service providers, that is, provider to provider. Service providers do not have to follow strict standardization in order to take part in the federation; it will suffice to be associated with a STS/IdP. Moreover, they can also keep their own security technology and use the Credential Translation when they need to deal with different technologies.

ShibGrid [26] and SHEBANGS [27] projects aim to provide Shibboleth based authentication for grid infrastructures, especially the UK National Grid Service (NGS). The authentication infrastructure behind NGS is based on X.509 certificates and proxies[18] This infrastructure is known as the Grid Security Infrastructure (GSI). SHEBANGS has adopted a proxied push model, wherein users first contact a credential translation service, which, after Shibboleth-based authentication and attribute retrieval, generates a credential stored in a MyProxy server. Access to that credential is then achieved by the user logging on to the NGS portal with details returned to them from the credential translation service. On the other hand, ShibGrid uses MyProxy servers to link two user identities (Shibboleth identity and X.509 certificate DN) instead of using the credential translation service. In both projects, the authentication credentials translation occurs only when there are two security domains (Shibboleth/SAML and Grid Security Infrastructure/X.509).

[18] NGS portals use MyProxy servers [28] as the means by which grid credentials are obtained.

Cardea [25] makes a dynamic assessment of the authorization requests, taking into account features of the resource and of the request instead of only assessing local identities. Users are identified by proxy X.509 certificates. Unlike what is proposed in the present study, Cardea builds the federation through the SAML standard, that is, SAML authorities, rather than through WS-Trust and WS-Federation specifications; besides, Cardea does not take into consideration different security technologies involved in the authentication process.

The TrustBuilder project[19] is investigating trust negotiation, an attribute-based access control (ABAC) model in which parties conduct bilateral and iterative exchanges of policies and certified attributes to negotiate for access to system resources. In the architecture proposed in this project, a policy compliance checker translates the attribute credentials from X509 certificate or a neutral format, such as XML, into statements in the policy language [29]. However, the Trust Builder approach does not deal with SSO's interoperability among heterogeneous security technologies.

In [30, 31], a Credential Conversion Service (CCS) is proposed to integrate authorization schemes. This service is responsible by translate non SAML-based credentials, such as X.509 attribute certificates into SAML Attribute Statements. In this approach, whole security domains need supported a authentication mechanism based on SAML assertions.

6 Conclusion

The concept of federated identities, which forms the basis of this research, (1) favors an effective and independent translation, (2) enables the client to use the resources of the federation through authentication carried out in its domain and (3) potentially increases the number of prospective clients for service providers.

Through the definitions proposed in the model, a client that is unknown to the service provider may undergo identity authentication thanks to the trust established among the domains. The provider does not need to know its prospective clients in advance, which thus makes it easier for clients to use the service and enhances business opportunities among service providers.

The model has overcome translation drawbacks by proposing the Credential Translation Service to translate the credentials that service providers failed to understand. Moreover, in the proposed solution, the model has gathered several security specifications in Web Services, which is not a simple task, given the large number of specifications and their complexity. In the future, by means of the Credential Translation Service, it will be also possible to translate not only X.509 and SPKI/SDSI but also authentication credentials in compliance with the Kerberos format (tickets) and according to biometrical profiles (e.g. digital fingerprinting). This will enable communication with providers and clients that support those technologies.

As regards the scalability of the proposed model, communication among domains should suffice to make credential exchange possible, as each domain has its

[19] http://dais.cs.uiuc.edu/dais/security/trustb.php

own credential provider. Certainly, this dynamics does not pose scale problems, since this is what happens in current Internet applications. In other words, in the proposed model there is no centralizing entity in charge of mapping the credentials. Each domain is responsible for the required mapping and, thus, the system scale is assured; it will suffice that well requested domains provide solutions for load distribution through, for example, the grouping of machines.

References

1. Jøsang, A., Pope, S.: User centric identity management. In: AusCERT Asia Pacific Information Technology Security Conference 2005. (May 2005)
2. W3C: Web Services Architecture. W3C Working Group. (February 2004) http://www.w3.org/TR/2004/NOTE-ws-arch-20040211.
3. Vogels, W.: Web services are not distributed objects. Internet Computing **7**(6) (November 2003) 59–66
4. Bartel, M., Boyer, J., Fox, B.: XML-Signature Syntax and Processing. W3C. (February 2002) http://www.w3.org/TR/xmldsig-core.
5. Imamura, T., Dillaway, B., Simon, E.: XML Encryption Syntax and Processing. W3C. (December 2002) http://www.w3.org/TR/xmlenc-core.
6. OASIS: eXtensible Access Control Markup Language (XACML) version 2.0. Organization for the Advancement of Structured Information Standards. (February 2005)
7. OASIS: Security Assertion Markup Language (SAML) 2.0 Technical Overview. Organization for the Advancement of Structured Information Standards. (June 2005)
8. OASIS: Web Services Security: SOAP Message Security 1.0. OASIS. (March 2004) http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.
9. WS-Policy: Web Services Policy 1.5. (March 2007)
10. WS-SecurityPolicy: Web Services Security Policy Language. (July 2005)
11. WS-PolicyAttachment: Web Services Policy Attachment. (March 2006)
12. WS-Trust: Web Services Trust Language (WS-Trust). (February 2005) http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Trust.asp.
13. Yavatkar, R., Pendarakis, D., Guerin, R.: A Framework for Policy-based Admission Control. IETF RFC 2753. (January 2000)
14. WS-Federation: Web Services Federation Language. (July 2003) http://msdn.microsoft.com/ws/2003/07/ws-federation.
15. Shibboleth: Shibboleth Architecture. (June 2005) http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.
16. Liberty: Introduction to the Liberty Alliance Identity Architecture. Liberty Alliance. (March 2003)
17. Internet2, EduCause: eduperson http://www.educause.edu/eduperson.
18. Wahl, M.: A Summary of the X.500(96) User Schema for use with LDAPv3. IETF RFC 2256. (December 1997)
19. Smith, M.: Definition of the inetOrgPerson LDAP Object Class. IETF RFC 2798. (April 2000)
20. InComm: Incomm federation: Common identity attributes http://www.incommonfederation.org/docs/policies/federatedattributes.pdf.

21. OASIS: Authentication Context for the OASIS Security Assertion Markup Language (SAML) v2.0. Organization for the Advancement of Structured Information Standards. (March 2005)
22. Morcos, A.: A Java implementation of Simple Distributed Security Infrastructure. Master's thesis, MIT (May 1998)
23. OASIS: Web Services Security: SAML Token Profile. Organization for the Advancement of Structured Information Standards. (December 2004)
24. Vecchio, D.D., Basney, J., Nagaratnam, N.: Credex: User-centric credential management for grid and web services. In: International Conference on Web Services, Orlando, Florida - EUA (2005) 149–156
25. Lorch, M., Proctor, S., Lepro, R., Kafura, D., Shah, S.: First experiences using xacml for access control in distributed systems. In: ACM Workshop on XML Security. (October 2003)
26. Spence, D., Geddes, N., Jensen, J., Richards, A., Viljoen, M., Martin, A., Dovey, M., Norman, M., Tang, K., Trefethen, A., Wallom, D., Allan, R., Meredith, D.: Shibgrid: Shibboleth access for the uk national grid service. In: Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), IEEE Computer Society (2006) 75
27. Jones, M., Pickles, S.: Shebangs final report. Technical report, University of Manchester (2007)
28. Basney, J., Humphrey, M., Welch, V.: The myproxy online credential repository: Research articles. Softw. Pract. Exper. **35**(9) (2005) 801–816
29. Winslett, M., Yu, T., Seamons, K.E., Hess, A., Jacobson, J., Jarvis, R., Smith, B., Yu, L.: Negotiating trust on the web. IEEE Internet Computing **06**(6) (2002) 30–37
30. Cï¿½novas, O., Lï¿½pez, G., Gï¿½mez, A.F.: A credential conversion service for saml-based scenarios. In: In Proceedings of 1st European PKI Workshop. (2004) 297–305
31. Lopez, G., Canovas, O., Gomez-Skarmeta, A.F., Otenko, S., Chadwick, D.: A Heterogeneous Network Access Service based on PERMIS and SAML. In: In Proceedings of 2nd EuroPKI Workshop. (July 2005)