I. Agudo, "Digital Identity and Identity Management Technologies", *UPGRADE - The European Journal of the Informatics Professional*, vol. 2010, pp. 6 - 12, 2010.
NICS Lab. Publications: https://www.nics.uma.es/publications

Digital Identity and Identity Management Technologies.

Abstract: *There are many technologies for identity management available in the form of open specifications, open source tools and commercial applications. Currently, there are some competing standards for identity management. At the beginning SAML was the only viable choice with a higher enough acceptance level. Recently, another technology called WS-Federation has also gain some attention from the community. Although this technology is not as mature as SAML, it modular design gives it some advantages over SAML. It this work we mainly focus on the WS-Federation and the family of specifications that surround it.*

Keywords: Identity Management, Identity Federation, Web Services

Introduction

It is hard to find a globally accepted definition of the term *Identity* and even harder to precisely define what is understood by *Identity Management*. A simplistic approach would consist on defining identity management as user accounts management in a software system. This was the general understanding some decades ago but in the last years, with the emergence of the Internet of Services, more complex aspects have arisen and the identity of its users have become crucial. In the first software systems, the identity of users was managed locally by the system administrator and was only valid for this particular application. In the Internet of Services, anyone can become a user of our applications and is the responsibility of the user to "manage" its identity in a proper way.

There are some concepts related with *Identity* that helps us understand the scope of *Identity Management* and its main challenges. Firstly, we have to make clear which *entities* can be attached to an identity. According to the Internet security glossary of RFC 2828, the term *entity* refers to "an active element of a system-e.g., an automated process, a subsystem, a person or group of persons that incorporates a specific set of capabilities." Although we mostly think of human beings when referring to entities, we cannot forget that in most of the cases we interact with computers instead of humans when using Internet.

Many definitions of the term identity can be found in the literature. The greatest common denominator of all these definitions is that an identity refers to some set of *claims, qualities or attributes* that make an entity unique and different from all other entities. Alternatively speaking, it is the individual characteristics by which an entity is recognized or known in a community or a given context. Consequently, an entity may have several identities depending on the context in which it interact. For example, a given person can be recognized as the CEO in the context of its company, but in a different context such as his bank or his house this reference might not be meaningful. Each identity can be referenced by one or more than one *identifiers* that are no more than special attributes that can be used to uniquely reference an identity.

The question of who you are is usually followed by the one of what you can do. In an environment where each entity may have different identities, the problem of deciding which *privileges* or *access rights* they own is not trivial. While the identity is the basis for authentication, privileges and access rights are the basis for authorization. In theory, authentication and authorization can be conceptually separated. In practice, however,

authentication and authorization are often combined and implemented in an authentication and authorization infrastructure (AAI), a privilege management infrastructure (PMI), or an alternatively-named but conceptually similar infrastructure.

The choice of one or another identity by a given entity determines not only its privileges but also the perception of the rest of entities in the system. When an entity interacts repeatedly with other entities on the system, some *trust relationships* can be established between them. Those trust relationships do not target other entities directly but their visible identities, i.e. the identities they use to interact with the rest of the entities in the system. Entities may behave well when using on particular identity but behave bad when using some other. The concept of trust is becoming highly relevant in Internet with growth of online social communities but is also highly relevant in other fields like sensor networks where the reliability of the network rely on the strength of trust relationships between the nodes of the network.

In order to prove the ownerships of its identities, an entity makes use of *ID cards*. An ID Card is seen as an abstract concept that attests for the legitimacy of an identity and/or its attributes, the same way we use our passport at customs.

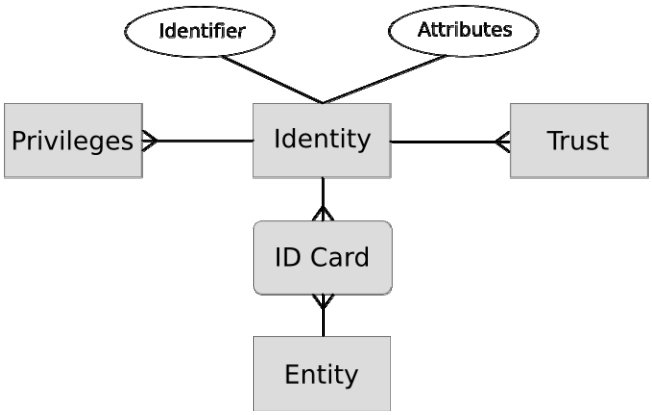

Figure 1. A simplified entity-relationship diagram (ERD) for the term *Identity*

Figure 1 shows an entity-relationships diagram that relates the terms: *Entity, Identity, Trust, Privileges* and *ID Card.* The ID Card can be seen as an associative element in the diagram that relates *entities* with its *identities.*

On the one hand, how identities and privileges are related falls into the field of *privilege management*. On the other hand, how entities and trust relates fall into the field of *trust management*. Then, the central part of the diagram fall under the umbrella of *identity management*. However it is very difficult to dissociate these three terms: Identity Management, Trust Management and Privilege Management.

Interoperability of Identity or Identity Federation

When people talk about identity, they sometimes underestimate the relevance of each of the terms mentioned before and most of the times focus on the notion of identification card (ID card). In essence, an ID card attests for the legitimacy of an identity and/or its attributes. There are ID cards for all kinds of purposes: passports and ID cards issued by the state, employee cards issued by companies, membership and customer cards issued by all kinds of

organizations and companies, student cards issued by universities, and so on. In spite of the fact that multiple-use ID cards are technically feasible, most ID cards in use today are single-use, meaning that they serve one single purpose or application. There may be many reasons for this fact; an important reason is certainly the fact that an ID card also serves customer relationship (so ID card-issuing organizations are reserved in sharing the card with other organizations and potential competitors). The omnipresence of single-use ID cards results in wallets that are filled with all kinds of cards. We know the problem from daily life, and we decide on a case-to-case basis which card to use in a given context.

The situation in the digital word is analogous. We can think of e-mail addresses as the most primitive form of identity in the digital world. It consists of one identifier without any attributes. Some people use more than one e-mail account, each of them in different contexts as they do with ID cards. They usually have an account for work and another for personal use, but they may have more in order to preserve their privacy, avoid Spam or even to be able to access online services that require a particular e-mail account. When we check our mail account we have to first demonstrate the mail server that we are the real owners of the account. For that purpose we typically make use of a combination of a username and a password. We can say that this combination serves as a kind of "ID Card" for our e-mail account that can be used against the mail server. Unfortunately, ordinary e-mail does not provide a mean to prove our identity to other users, for that purpose we need to make use of other standards for secure e-mail.

As we mentioned above, physical ID cards are usually not interoperable but nevertheless there is a common understanding of how an ID card looks like. There are eleven workgroups under ISO/IEC JTC1/SC17 working on "Cards and Personal identification" standards. They have produced a standard that defines the physical characteristics for identity or identification cards, ISO/IEC 7810:2003. Unlike the physical world, however, the form of ID cards is not yet settled down in the digital world. In fact, there are many possibilities to implement digital ID cards. An example of a widely used approach that may settle the basis for digital ID cards is digital certificates or public key certificates defined in the X.509 ITU-T standard. This standard specifies the format of the certificates as well as the algorithms and mechanisms needed for it deployment. Apart from the passwords and the certificates there are some other mechanisms that we can use to prove our identity. Security grid cards that we are used to in banking environments and cryptographic tokens, such as SecurID from RSA, that are widely used in corporate environments are only two more examples.

One of the main challenges that we have to face in the field of identity management is the interoperability of identities. It is not enough to be able to manage the identities within our system; we need to be able to provide mechanisms for the reusability of the identities of our users outside of our domain. This requires the establishment of interoperability mechanisms between the different stakeholders of the digital identity business.

When describing the supporting technologies for identity interoperability, or identity federation, that are becoming "de facto" standards we have to inspect not only the format used to describe the identity or credentials but also the communication protocols used to transport this information. In fact, the biggest differences between the two approaches that

we present in this work focus on the protocols and not in the format of the identity. Those two technologies, which offer approximately the same functionality, are: SAML 2.0 and WS-Federation. SAML 2.0 is mature OASIS standard specification (2005) and is widely deployed. Most European universities and US Universities are already using SAML with the support of the GÉANT network and the Internet2 consortium respectively. On the other hand, WS-Federation is a novel OASIS Standard built upon the WS-* family of specifications. The last version of WS-Federation was released in 2009. However, WS-Federation is attracting the attention of the internet society, mainly because of the success of the Web Services Security Suite (WS-*). From our point of view WS-Federation have not get enough attention yet whereas SAML has been widely revised in the past. This is the main reason why we pay more attention to WS-Federation in this work. One of the main differences between those two approaches, at least in their beginning, is that SAML was intended to solve the Single Sign-On (SSO) problem in web environments whereas WS-Federation tries to cover the area of web services. Both solutions have evolved and have partially converged during the last few years but their origins have left a mark in their developments that make then different in several aspects.

Identity Management in the Web Services World

The OASIS consortium is promoting relevant standards for identity management in the area of web services. In particular, the following four are the most relevant technical committees (TC) with regards to Identity Management:

- Web Services Security (WSS)
- Web Services Secure Exchange (WS-SX)
- Web Services Federation (WSFED)
- Identity Metasystem Interoperability (IMI)

The OASIS Web Services Security TC released in 2006 the last version of WS-Security, initially developed by IBM, Microsoft, and VeriSign. It provides three basics security mechanisms for Web services: sending security tokens as part of a message, message integrity, and message confidentiality. A security token represents a collection of claims, where the term claim is interpreted as a statement made about a client, service or other resource (e.g. name, identity, key, group, privilege, capability, etc.). The specification support different security tokens as described in the following associated profiles:

- Username Token Profile 1.1
- X.509 Token Profile 1.1
- SAML Token profile 1.1
- Kerberos Token Profile 1.1

This specification works in the application layer, providing end-to-end security, versus point-to-point as provided by TLS/SSL, by incorporating security features in the header of SOAP messages. It provides a standard set of SOAP extensions that can be used when building secure

Web services to implement message content integrity and confidentiality. This extension is referred as “Web Services Security: SOAP Message Security” or simply “WSS: SOAP Message Security”. Apart from allowing the use of different security token formats, it also supports multiple trust domains, multiple signature formats, and multiple encryption technologies. This specification cannot be considered as complete solution for secure web services but rather as a basic building block.

This specification cannot be considered a complete solution for Web services but instead a building block that can be used in conjunction with other Web service extensions and higher-level application-specific protocols. Indeed, most of the following specification relay in this one.

The OASIS Web Services Secure Exchange (WS-SX) TC focus on the definition of extensions to the previously defined OASIS Web Services Security (WS-Security) specification that enable trusted SOAP multi-message conversation (versus the simpler request-response mechanism) via the establishment of a shared security context, and also the definition of security policies regarding the format of the messages and the kind of tokens included in them. This technical committee is in charge of three specifications:

- WS-Trust.
- WS-SecureConversation.
- WS-SecurityPolicy.

WS-Security relies on the existence of certain trust relationships between the participants in the communications, i.e. the web service providers and requestors. Credentials presented by the requestor have to be trusted by the provider and vice versa. How these trust relationships are established is out of the scope of WSS and here is where WS-SX TC focuses, by adding additional primitives that enable the establishing and brokering of these trust relationships between SOAP message exchanges participants.

WS-Trust focus on the definition of a Security Token Service (STS) that issues security tokens in accordance with the WS-Security specification. The specification describes mechanisms for issuing, renewing, and validating security tokens. It establishes the format of messages used to request security tokens and its responses. It also provides mechanisms for key exchange. It makes use of WS-Addressing to describe endpoints.

WS-SecureConversation introduces the context authentication model. This model is based on the use of a new WSS token type, named Security Context Token (SCT), which is obtained using a binding of WS-Trust. A security context token implies or contains a shared secret that although can be used for signing and/or encrypting messages by itself, it is recommended to be used to derive other keys for signing and/or encrypting messages within this security context. A security context token can be created by a Security Token Service (STS) defined in WS-Trust, by one the participating entities alone or cooperatively via message exchanges among the participants. The mechanisms to distribute SCT are covered in WS-Trust. Security contexts are shared among the communicating parties for the lifetime of a communications session but its lifetime can be extended by renewing it or reduced by cancelling it.

WS-SecurityPolicy defines a set of security policy assertions that conforms to WS-Policy framework, regarding some security features introduced in: WS-Security, WS-Trust and WS-SecureConversation. These policy assertions cover aspects related to which token types, cryptographic algorithms and mechanisms are allowed in a secure exchange of messages. They can also be used for describing security requirements at a more general or transport-independent level. The main goal of this specification is to define an initial set of assertions that is both flexible and specific enough to ensure proper interoperability of security mechanisms between the participants in the communication. This specification also tries to make policy assertions simpler enough such as the policy intersection mechanisms introduced in the WS-Policy framework can provide a narrowed set of policy alternatives that are shared by the two participants that attempt to communicate.

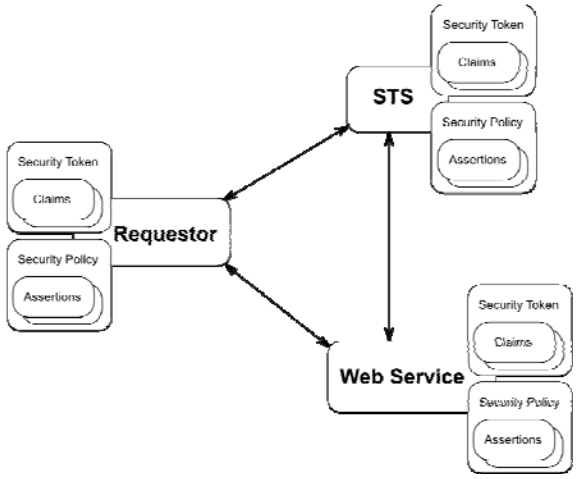

Figure 2. Arquitectura y elementos principales en WS-Trust

When the context of the text makes it clear enough we use WS-Trust to reference these three specifications. There are three main players in the WS-Trust model (See Figure 2). The client, *Requestor* in the figure, makes web service requests to web-service providers that in WS-Trust terminology are called *Relaying Parties*, in the sense that they rely on tokens issued by the *Security Token Service (STS)*. The STS is in charge of issuing, renewing and verify the security tokens among other tasks.

The OASIS Web Services Federation TC has recently approved (May 2009) version 1.2 of the Web Services Federation Language (WS-Federation) specification. WS-Federation defines mechanisms that allow different security realms to federate. In particular, it provides mechanisms for brokering of identity, attribute, authentication and authorization assertions between realms, and privacy of federated claims. This specification covers both Active and Passive Requestors. Active requestors are those applications capable of communicating using Web services (SOAP) whereas passive requestors are Web browsers or applications capable of communicating using HTTP. It devotes one full charter to privacy issues which gives an idea of the relevance of privacy when dealing with federated identity management.

Web service requestors must be compliant with WS-Security and WS-Trust mechanisms and be capable of interacting directly with Web service providers. The Web browser mechanisms

describe how the WS-* messages are encoded in HTTP. This definition allows WS-Trust, WS-Policy, and other WS-* mechanisms to be properly leveraged in Web browser environments.

The typical scenario covered by WS-Federation is this in which resources managed in one realm can be accessed by entities whose identities are managed in other realms. The mechanisms presented in this specification enable authorization decisions to be based on the sharing and interchange of identity, attribute, authentication and authorization assertions between realms.

The federation framework defined in this specification builds on top of the WS-* family of specifications, in particular WS-Security and WS-Trust, providing a rich extensible mechanism for federation. Therefore, it also allows for different types of security tokens, infrastructures, and trust topologies. In order to describe what aspects of the federation framework are required/supported by federation participants; the use of WS-SecurityPolicy is recommended.

All those specifications together allow identities from one realm to be properly recognized in any other. However, the problem that the final user faces by the co-existence of many digital identities is not covered in any of them. In fact, none of this they do not directly deal with digital identities but with security tokens directly.

The OASIS Identity Metasystem Interoperability (IMI) TC approved in 2009 the Identity Metasystem Interoperability specification that aims at integrate the Digital Identity into the WS-world using the Information Card Model. In the IMI specification, Digital Identity is specifically defined as a set of Claims made by one party about another party. If we look at the definition of security tokens given in WS-* specifications we see that both terms are very similar. It introduces the term Identity Selector which allow users to manage their Digital Identities and used them according to the context of the application. Although information cards are more oriented to Web browsers, they can also be used with Web Services. This specification also provides an extension to WS-Addressing to describe secure and verifiable identities for endpoints.

An *Information Cards* is a signed XML document representing a digital identity of a subject, i.e. a set of claims. We can consider two kinds of Information Cards: Self-issued or Personal Information Cards that are generated and signed by an individual and Managed Information Card that are generated and signed by a third-party Identity Provider. Information Cards can be used to both signing-in and signing-up. Self-issued card would be normally used to sign-up as they do not require a previous trust relationship between the subject and the relaying party, whereas signing-in might require a managed card. The Identity Selector is a piece of software in charge of managing Information Cards. It allows users to select the most appropriate Card in any given context. It prompts the user with a list of Cards that match security policies specified by the application that is acting as a Relaying Party. It is also in charge of retrieving a Security Token from the associated Identity Provider when the user chooses a card. When retrieving the security token, the subject must authenticate to the STS or Identity Provider. There are four supported authentication mechanisms:

- Username and Password Credential

- Kerberos v5 Credential
- X.509v3 Certificate Credential
- Self-issued Token Credential

The Identity Card model, also known as CardSpace, was promoted initially by Microsoft but partially thanks to the *promise of open specifications*[1] made by Microsoft some open source implementation of the model have emerged such as Higgins, Bandit, OpenInfoCard and Pamela. After the approval of the IMI specification as an OASIS standard it is probable that more a more companies provide support for it.

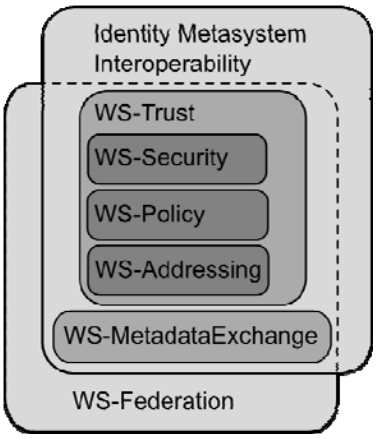

Figure 3. Dependencies of WS-* and Indentity Metasystem Interoperability specifications

In Figure 3, the modular structure of the family of WS-* specifications is shown. In the lower level we have the WS-Security, WS-Policy and WS-Addressing specifications that are in charge of providing the basic mechanism for the definition of the security tokens, the associated policies and the addressing mechanism. In the next level we can find the WS-Trust layer that focuses on the Secure Token Service. As we mentioned before, this layer also includes the WS-SecureConversation and WS-SecurityPolicy specifications. In the highest level we can find both the WS-Federation specification and the IMI specification. All those specifications together provide the mechanisms needed to cover all the aspects of Identity Management in the Web Services world.

SAML and related technologies

The *Security Assertion Markup Language* (SAML) is an XML-based standard for exchanging authentication and authorization data between security domains maintained by the OASIS Security Services Technical Committee (SSTC). There are two main actors in this interchange of information: the identity provider (Security Token Service in WS-* specifications) and the service provider (Relying Party in WS-* specifications). The initial purpose of SAML was to provide a Web Browser Cross-Domain Single Sign-On experience whereas WS-* specifications were originally targeted at providing a security extension for Web Services. However, we have already mentioned that WS-Federation also provides mechanisms for passive requestors, i.e.

[1] http://www.microsoft.com/interop/osp

Web Browsers. Hence, there is a functionality overlap between both specifications. The SSTC has also published a specification for federation metadata (Metadata for the OASIS SAML V2.0) that has been adopted by WS-Federation in it last version. We guess that in the near future we will see more initiatives for the convergence of both specifications

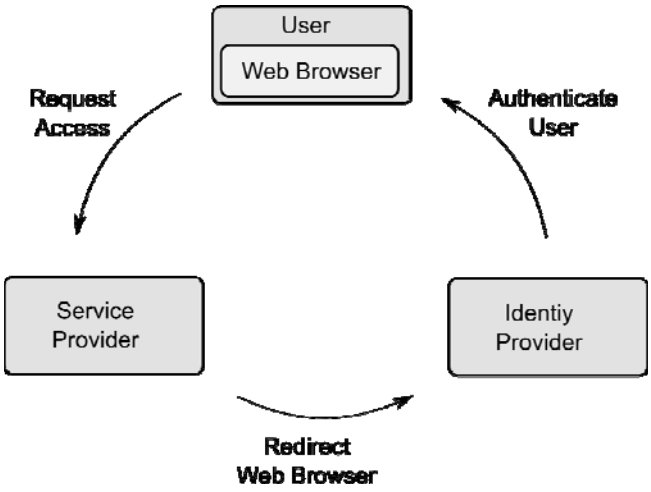

Figure 4. SAML Web Browser SSO Actors

As we can see in Figure 4 the overall idea of SAML is similar to the one subjacent in WS-Federation passive requestors' profile, although the terminology is different.

On top of the first version of SAML, the Liberty alliance proposed its Liberty Identity Federation Framework (ID-FF). Liberty Alliance is a large consortium of both companies and non-profit and government organizations that has played an important role in the evolution of SAML. Most of the changes proposed in ID-FF have been incorporated in SAML 2.0. We can say then that SAML is more mature than WS-Federation, but some SAML profiles that extend the core functionalities are still in development of have been just recently approved, e.g. SAML 2.0 Holder-of-Key Assertion Profile Version 1.0 was released on July 2009. Another initiative that supports SAML is Shibboleth, promoted by the Internet2 consortium.

The ID-WSF specification from Liberty alliance provides mechanisms that allow using SAML tokens in web services. In fact, the Liberty alliance is currently focusing its efforts in the integration of SAML in the web services world. The main reason is that the rest of proposals have been already adopted by SAML.

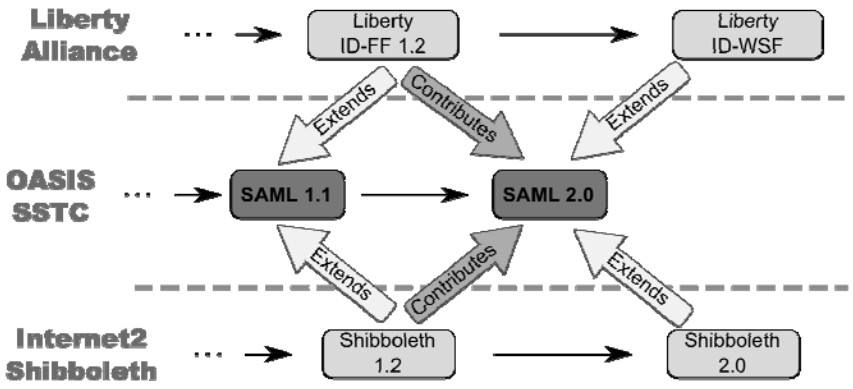

Figure 5. Evolution of SAML and its related specifications

Another initiative that has helped evolving SAML is the Shibboleth suite, promoted by the Internet2 consortium. Shibboleth is a set of applications that provides a full identity federation solution on top of SAML protocols. Shibboleth itself is built upon OpenSAML, developed by the same consortium.

The relationships between different versions of SAML, Shibboleth and Liberty are shown in Figure 5. We can see that both Shibboleth and Liberty has fully adopted SAML 2.0, which received feedback from both initiatives. In fact, both Liberty and Internet2, as members of the SSTC are also contributing to the completion of all SAML 2.0 related specifications.

Conclusions

We have seen that both SAML and WS-Federation provide a similar functionality. Indeed, the actors and the abstract information flows are almost the same. There is one entity that attests the user identity and another that trust this, let's call it, identity statement. In SAML terminology those two entities are called Identity Provider (IdM) and Service Provider (SP) respectively, whereas in WS-* terminology they are called Security Token Service (STS) and Relying Party (RP).

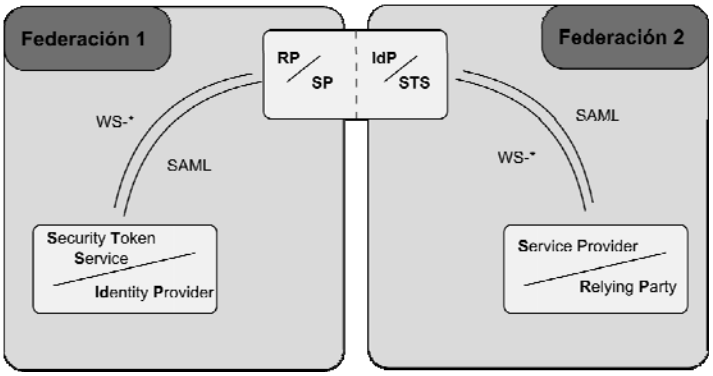

Figure 6. Bridging Federations with different technologies

Most commercial identity management solutions support both technologies by letting those two entities, the attesting entity and the trusting entity, to speak and understand these two different Identity Languages. It is a choice of the system administrator whether to allow the use of both or whether to stick to one of them. In case we had two different identity federations using each of them a different technology we could have an interoperability problem. A simple solution to help two a priori incompatible identity management systems to interoperate would be to place a common entity that switches roles in each of the federations, in one of them would act as an attesting entity whereas in the other would act as a trusting entity. This common entity is called an Identity Bridge and must be able to speak the languages of each of the federations we want to interconnect. It will basically act as a translator for them. A simple scenario is shown in Figure 6.

It is worth mentioning that Microsoft, Sun and Novell have made several joint efforts to validate the interoperability of their latest identity solutions with regards to the two specifications reviewed in this article. This puts into relevance the importance of having reference open specification for enabling the real interoperability of Identity.

Referencias

[X.509] ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Authentication Framework. (ISO/IEC 9594-8:2001)

[SAML] “Security Assertion Markup Language (SAML) v2.0” OASIS Security Services TC, March 2005. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[WS-Policy] W3C Recommendation, "Web Services Policy 1.5 - Framework", 04 September 2007. http://www.w3.org/TR/2007/REC-ws-policy-20070904/

[WS-Security] “Web Services Security v1.1” OASIS Web Services Security TC, February 2006. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[WS-Federation] "Web Services Federation Language (WS-Federation) v1.2" OASIS Web Services Federation (WSFED) TC, May 2009. http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.pdf

[WS-MetadataExchange] “Web Services Metadata Exchange (WS-MetadataExchange), Version 1.1”, August 2006. http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf

[WS-SecureConversation] “WS-SecureConversation v1.4”, OASIS Web Services Secure Exchange (WS-SX) TC, February 2009. http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.4/os/ws-secureconversation-1.4-spec-os.pdf

[WS-SecurityPolicy] “WS-SecurityPolicy 1.3”, OASIS Web Services Secure Exchange (WS-SX) TC, February 2009. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ws-securitypolicy-1.3-spec-os.doc

[WS-Trust] “WS-Trust 1.4”, OASIS Web Services Secure Exchange (WS-SX) TC, February 2009. http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf

Biography

Isaac Agudo has an MSc in Mathematics and a PhD in Computer Science by the University of Malaga. In 2002 he started working in the CASENET Project, found by the V Framework Program of the UE. In 2004 he won a grant from the Andalusia government to finish his PhD. Since 2008 he has been involved mainly in the European projects PICOS and SPIKE from the VII Framework Program among some other national and European projects.