Federated Identity Management Solutions

Jyri Kallela
Helsinki University of Technology
jkallela@cc.hut.fi

Abstract

Federated identity management allows users to access multiple services based on a single authentication. This reduces the number of digital identities that users have to manage, and lowers the administrative costs of the organisations providing the services. Several technologies and frameworks have been developed to carry out the necessary activities related to identity federation, such as linking user accounts across services and enabling Single Sign-On. This paper covers some of the most important solutions currently available, namely the Security Assertion Markup Language, the Liberty Alliance framework, Shibboleth and WS-Federation. It presents their overall architecture and provided features, and also discusses some of the deployment and usability aspects related to the solutions.

KEYWORDS: Identity management, federated identity, single sign-on, SAML, Liberty Alliance, Shibboleth, WS-Federation

1 Introduction

The number of services people are using on the Internet today is growing rapidly. In addition to traditional Web surfing, the Internet is also increasingly used for other activities such as shopping, booking holidays, sharing photos and videos, keeping in touch with friends, and so on. These more advanced services often require the user to set up a personal account at the service's Web site, which usually includes at least coming up with a username and password for the site and remembering them in the future. Thus, the increasing number of services poses a new kind of challenge to the users: how to manage all the digital identities in the various Web sites they use?

Identity management is not problematic to users alone, but is becoming a challenge to organisations as well. If they wish to practice e-business efficiently, companies must set up various online resources for their customers and partners. Managing all the user identities for the services locally and often also across organisational boundaries can lead to noticeable administrative costs for the businesses.

As a solution to these problems, the concept of *federated identity management* has emerged over the last few years. It is a general term that currently refers to a set of technologies and standards which enable users to interact with many services by signing in to only one. One of the basic functions offered by these identity federation technologies is Single Sign-On (SSO), which means the procedure of authenticating a user on one Web site, and using that authentication as a validation to access some protected resources on other sites. More complex scenarios can also be handled, such as sharing some attributes of a user's account across domains, and letting the user control what information can be shared. A requirement and the basic foundation for this kind of cooperation is a trust relationship between the parties.

A number of different solutions currently exist in the field of federated identity management. There are open standards developed by large consortiums, proprietary solutions by individual companies, and smaller-scale open-source projects. Many of them include similar functionality, but they differ in the scope of the solution and its applicability to different scenarios. This paper focuses on some of the most important solutions currently available. It presents an overview of their basic architecture and then discusses some specific aspects regarding their scope, deployment and use, such as their suitability for different environments, the covered use cases, and how they differ from the user's perspective.

The rest of this paper is organised as follows. First, the Security Assertion Markup Language (SAML) [9] is introduced. It is a standard for exchanging security-related information between organisations. Then, two SAML-based identity federation solutions are presented, namely the architecture framework from the Liberty Alliance [15, 14], and Shibboleth [11]. The last solution covered in this paper is the Web Services Security (WSS) based family of standards, most importantly WS-Federation [7]. After presenting the technical architecture of these solutions, they are discussed and compared in more detail. Finally, the paper is concluded.

2 Identity Federation Architectures

2.1 Security Assertion Markup Language (SAML)

SAML [9] is an XML-based standard developed by the Organisation for the Advancement of Structured Information Standards (OASIS). It is a framework for describing and exchanging security-related information between organisations. The goal of the framework is to deliver a vendor-independent way of achieving SSO and identity federation capabilities between services in multiple domains. The most recent version of the framework at the time of writing is 2.0.

At the heart of the standard, SAML introduces the entities referred to as the *asserting party* and the *relying party*. The asserting party can generate signed SAML *assertions*, which contain one or more *statements* about a *subject*. The subject can be any entity that can be identified in the security domain, such as a user or a computer, who is usually authenticated using an appropriate method. However, SAML itself does not specify or require any specific authentication mechanisms.

After generating the assertions, the asserting party can share them with a relying party. The relying party verifies that the assertion is valid, and can then make a decision to accept the assertion and start providing services to the subject specified in the assertion. The decision to rely on information from the asserting party requires that a trust relationship exists between the two parties.

The SAML framework is based on a few main components, which work as building blocks that can be combined in various configurations to support different kinds of use cases. They are illustrated in Figure 1. Next, we cover each of these components and their role in the overall framework.

Assertions are the format in which security information is exchanged between SAML parties. An assertion contains the following information: the subject of the assertion, the conditions that are used to validate the assertion, and the statements about the subject.

An assertion can contain three kinds of statements. First, authentication statements are generated by the entity that authenticated the subject. They usually contain at least the authentication method used and a timestamp specifying the moment of authentication. Second, authorization decision statements specify the actions a subject is permitted to do in the system. Third, attribute statements contain some specific attributes about the subject. For example, a typical SAML assertion could state that the user's name is John Doe, he was authenticated using password authentication, and that he is entitled to buy items from a specific online store.

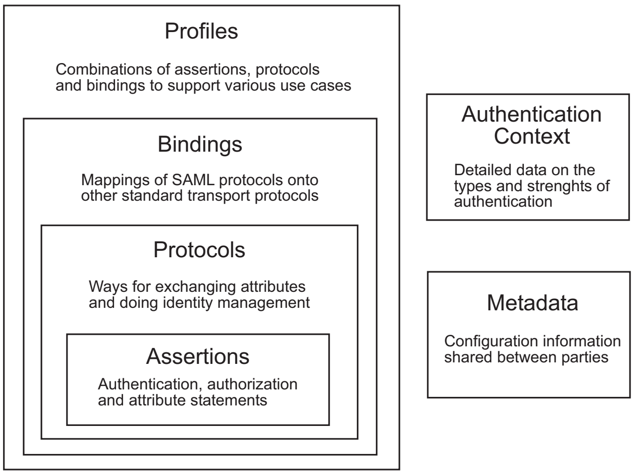
Profiles
Combinations of assertions, protocols and bindings to support various use cases
Bindings
Mappings of SAML protocols onto other standard transport protocols
Protocols
Ways for exchanging attributes and doing identity management
Assertions
Authentication, authorization and attribute statements
Authentication Context
Detailed data on the types and strenghts of authentication
Metadata
Configuration information shared between parties

Figure 1: The main components of SAML

Protocols define the means by which parties can exchange assertions and other information needed in performing the functions supported by the SAML framework. There are in total six general protocols specified. For example, the Authentication Request Protocol defines how a relying party can request assertions containing authentication statements. This protocol can be used, for example, in a SSO scenario by a service provider, when it needs to request another party to authenticate a specific user.

Bindings specify how the SAML protocols can used with different kinds of transport protocols. One design goal in SAML is that the protocols are generalized and the protocol messages can be used together with various underlying technologies. For example, the SAML specification defines bindings for HTTP Redirect and POST methods and Simple Object Access Protocol (SOAP) Web service messages. In addition, a SAML assertion can be retrieved by resolving a URL.

Profiles tie assertions, protocols and bindings together by defining how they can be used in combination in certain usage scenarios. For example, the Web Browser SSO Profile defined in the SAML specification specifies how SSO can be achieved in standard Web browsers by using the SAML protocols and their HTTP bindings.

In addition to these main components, the SAML framework specifies two other concepts that further support the deployment of a SAML environment. The first one is **Metadata**, which can be used to share configuration information between SAML parties. For example, the protocol bindings supported by an entity or key information for encryption and decryption can be expressed with SAML metadata documents. The second concept is **Authentication Context**, which means detailed information about how a certain subject has been authenticated, including the type and strength of the used authentication method.

2.2 The Liberty Alliance Framework

The Liberty Alliance is a global consortium of more than 150 companies and organisations, whose goal is to develop open standards for federated identity management and identity-based Web services. In addition, the alliance offers best practices and guidelines to businesses that use their standards, and a certification program for products utilising their specifications.

The main offering of the Liberty Alliance is their federated identity management architecture framework. It consists of three main components: the Identity Federation Framework (ID-FF) [15], the Identity Web Services Framework (ID-WSF) [14], and the Identity Services Interface Specifications (ID-SIS). These are illustrated in Figure 2.

The Liberty Alliance framework is built on other standardized technologies, such as XML, SOAP and SAML. The first versions of ID-FF were built on top of SAML. After finishing the ID-FF version 1.2, the Liberty Alliance submitted the specifications to OASIS, which incorporated many of the features in SAML 2.0. This made SAML a superset of ID-FF and a foundation for the Liberty framework. [4]

Next, we go through the content and relationships of the main components of the framework.

2.2.1 Identity Federation Framework (ID-FF)

The goal of the ID-FF [15] is to enable basic identity federation capabilities among organisations belonging to a so-called *circle of trust*. The circle of trust is formed between organisations using Liberty-enabled technology that have trust relationships defined by operational agreements. In a circle of trust, an organisation can take either the role of an *identity provider* (IdP) or a *service provider* (SP), or both. An identity provider manages the user identities and authenticates the users, and a service provider can accept an identity federation from an identity provider. The basic ID-FF functionality does not require any additional software in the user's machine other than a standard Web browser.

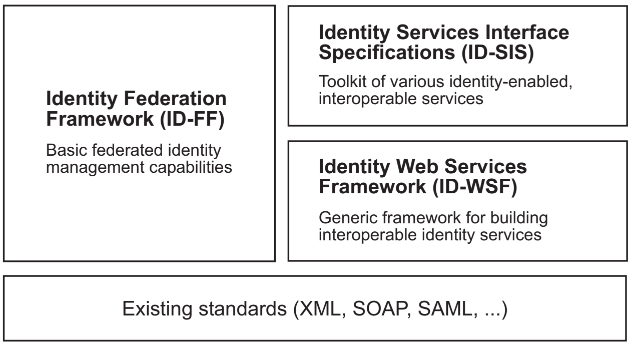
Identity Federation Framework (ID-FF)
Basic federated identity management capabilities
Identity Services Interface Specifications (ID-SIS)
Toolkit of various identity-enabled, interoperable services
Identity Web Services Framework (ID-WSF)
Generic framework for building interoperable identity services
Existing standards (XML, SOAP, SAML, ...)

Figure 2: Components of the Liberty Alliance framework

The ID-FF itself consists of three components. The first one is **Web redirection**, which leverages the standard HTTP Redirect and HTTP POST mechanisms to create a communication channel between the IdP and SP. The second one is the **Web services** component, which is used to execute some of the Liberty protocol steps directly between the system entities by using SOAP messages. The third component is **metadata and schemas**, which refers to various types of information exchanged between the IdP and SP, either via a protocol or out-of-band. Metadata can include information such as the user handle for the federated identity, an authentication context, and security certificates of the parties.

The ID-FF defines mechanisms for handling basic identity federation functions, such as account linking and SSO. In a circle of trust, each member site can have a different local identity for the user. An IdP can ask the user if he wants to allow introductions to other members in the circle of trust. If the user accepts, the other member sites will notice the introduction, and ask the user if he wishes to federate his identity between the two sites. If the user accepts the federation, a pseudorandom *opaque user handle* will be created at the IdP and SP, which will be mapped to the user's local identity at both sites. An example of a federation is shown in Figure 3.

After a federation has been established between an IdP and one or more SPs, SSO can be carried out between these sites. The user can login at the IdP, which then creates assertions for the user, enabling him to access the SP sites without the need for re-authentication. Multiple IdPs can also exist in a circle of trust, allowing the user to login at any one of them. In addition to just the identity, various user attributes can also be shared with the user's consent. These federation functions are achieved with the various protocol bindings and profiles defined in the ID-FF.

2.2.2 Identity Web Services Framework (ID-WSF)

The ID-WSF [14] leverages functionality in the ID-FF to create a framework for creating, using and consuming so-called *identity services*. They are Web services that can either retrieve or update identity information or perform certain actions based on identity information. The ID-WSF provides templates for creating services on top of the framework and specifies some supporting functionality, such as a discovery service for locating identity services, and mechanisms to provide security. The goal of the ID-WSF is to define a generic way of deploying interoperable Web services for securely exchanging and handling identity information. These Web services can then be used to support more complex use cases than those enabled by the basic ID-FF alone.

2.2.3 Identity Services Interface Specifications (ID-SIS)

The ID-SIS extends the ID-FF and the ID-WSF to define a diverse range of actual identity-enabled Web services. At the time of writing, a number of services have already been specified by Liberty, such as a directory access service, personal and employee profile services, a contact book service, a geolocation service and a presence service. The framework will also address the emerging need to enable identity federation through mobile technology. The goal of ID-SIS is to provide a versatile toolkit of interoperable and secure identity Web services for various application purposes. [13]

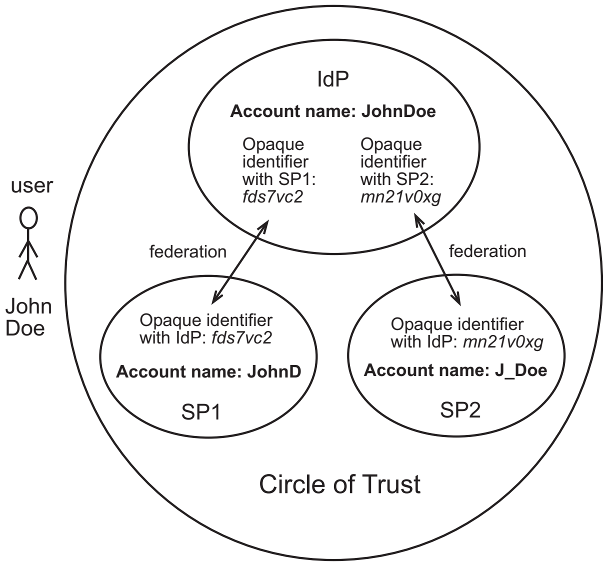
IdP
Account name: JohnDoe
Opaque identifier with SP1: fds7vc2
Opaque identifier with SP2: mn21v0xg
user
John Doe
federation
federation
Opaque identifier with IdP: fds7vc2
Account name: JohnD
SP1
Opaque identifier with IdP: mn21v0xg
Account name: J_Doe
SP2
Circle of Trust

Figure 3: Circle of trust and identity federations

2.3 Shibboleth

Shibboleth [11] is a project founded and controlled by Internet2. The consortium's initial purpose was to develop a Single Sign-On and identity federation system for the needs of academic institutions, such as universities. Currently, however, Shibboleth seems to be developing into a general federated identity management solution that can be adopted by any type of organisation. The latest production version of Shibboleth at the time of writing is 1.3, and it is based on SAML version 1.1 specifications. The version currently in development is 2.0, which will adopt and extend the features of SAML 2.0.

Like the Liberty Alliance framework, Shibboleth also uses the concepts of an identity provider and a service provider.

An IdP is an entity that manages user identities and attributes, and generates authentication and attribute statements to service providers. In Shibboleth, an IdP is divided into the following four subcomponents:

- The **authentication authority** is integrated to the identity provider's authentication service. It issues authentication statements to the other components.
- The **Single Sign-On service** handles the SSO process by initiating the authentication of a user, obtaining the required assertion and generating a HTML form which redirects the user back to the service provider.
- **Artifact resolution service.** An *artifact* is a concept of SAML. It means a reference to an assertion instead of the assertion itself. If artifacts are used, a service provider can send an artifact to this service, which then returns the actual assertion requested.
- The **attribute authority** issues attribute assertions based on requests by service providers. Before issuing assertions, it first authenticates and authorizes all received requests.

Correspondingly, a service provider is an entity that manages secured resources, which are accessed based on authorization statements received from an identity provider. Shibboleth uses the concept of a *security context* that needs to be established for a client before allowing it to access the secured resource. A SP consists of the following two subcomponents:

- The **assertion consumer service** manages the SSO functions in the service provider's endpoint. It processes the received authentication assertion or artifact, initiates the request of optional additional attributes, establishes the security context and redirects the user to the protected resource.
- The **attribute requester** can interact with the IdP's attribute authority to exchange additional attributes once a security context has been established. This interaction happens directly between the services using a SAML protocol binding, without using the client's browser.

An additional entity specified by Shibboleth is the WAYF ("Where are you from?") service. Since a SP does not necessarily know which IdP it should use to authenticate the user, it can redirect the user to a centralized WAYF service. The WAYF service then provides the user a means to select the IdP which should be used and then continue to its site. The WAYF service acts as a proxy between the SP and IdP, and relays the Shibboleth authentication request to the IdP's SSO service.

To handle the SSO usage scenario, Shibboleth specifies two SSO profiles for Web browsers, which are based on corresponding SAML profiles. The *Browser/POST profile* uses SAML assertions directly, while the *Browser/Artifact profile* relies on artifacts, which the assertion consumer service must dereference with a request to the IdP's artifact resolution service.

An example of the basic SSO protocol flow in the Browser/POST Profile is shown in Figure 4. As with both profiles, the sequence begins with a request for a protected resource at the SP.

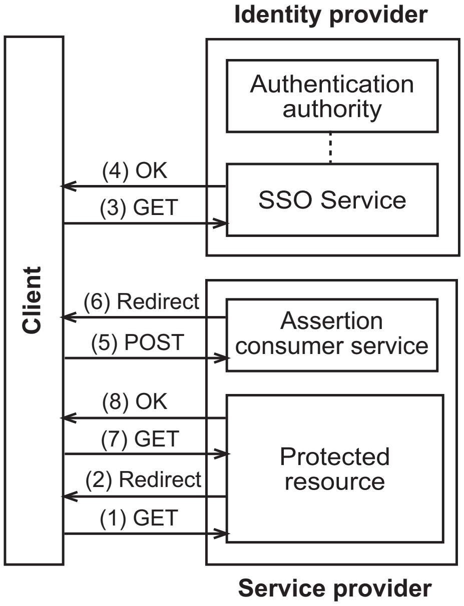

Figure 4: Basic protocol flow in Shibboleth SSO

In step 1, the client requests a target resource at the SP and in step 2, the SP redirects the client to the SSO service at the IdP. Then, the client performs an authentication request to the SSO service, which identifies the user and obtains an authentication statement from the authentication authority. In step 4, the SSO service responds with a document containing a HTML form, and a digitally signed, base64-encoded SAML assertion is included in the form parameters. In the fifth step, the client issues a POST request to the assertion consumer service at the SP. In step 6, the assertion consumer service processes the authentication response, creates a security context for the client and redirects it to the target resource. In step 7 the client requests the target resource at the SP and in the last step, the SP returns the resource to the client.

Both SSO profiles can also optionally use a WAYF service. In the previous example, the SP could redirect the client to the WAYF service in step 2, and the user could select the correct identity provider before being redirected to it. In addition, if the SP needs to retrieve additional attributes about the user in order to make the authorization decision, an attribute exchange step can be added to the SSO sequence. In that case, the assertion consumer service initiates the attribute exchange, and the attribute requester interacts with the attribute authority before a security context is established and the client is redirected to the target resource.

In the Shibboleth project, a *federation* consists of a group of organisations that have agreed upon a set of common practices and policies for identity federation, such as the used security technologies, procedures for handling sensitive personal information and the type of organisations that can participate in the federation. In addition, the participants of a federation usually have to define a common set of user attributes that will be exchanged in order to make cross-domain identity federation feasible. To address this issue, Shibboleth provides built-in support for a pre-defined attribute schema called *eduPerson*, which is a specification originally developed for the needs of higher education communities [6]. It is based on the Lightweight Directory Application Protocol (LDAP) standard, which makes it possible to integrate Shibboleth into existing LDAP directory services, enabling them to be used inter-organisationally.

2.4 The Web Services Security Framework

A slightly different approach to federated identity management than those discussed above is included in an initiative made by Microsoft and IBM, along with some other major companies. Together, they have proposed a set of specifications that extend the basic Web services protocol stack with additional security features. The main goal of this effort is to improve the interoperability of different information systems across the Internet, and to create a secure context for their communication.

The Microsoft/IBM Web Services Security (WSS) framework, also known as WS-*, is composed of a number of specifications. At its core are the basic, widely adopted Web services technologies, such as XML and SOAP. The WSS framework extends this foundation with a layer called WS-Security, which adds some basic security mechanisms to Web services, such as message confidentiality, integrity and authentication. The features of WS-Security are further extended by the next three specifications on the protocol stack, which are called WS-Policy, WS-Trust and WS-Privacy. On top of this foundation lie various other WS-* specifications, including WS-Federation, which is the main focus of this chapter. The overall structure of the WSS architecture is presented in Figure 5. [1]

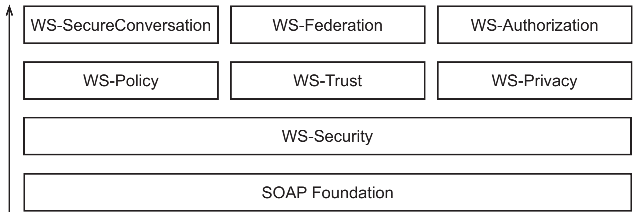

Figure 5: Elements of the Web Service Security framework

Before discussing the identity federation capabilities of WS-Federation, it is important to understand some of the related underlying specifications and their concepts. Therefore, we next go through the main aspects of the relevant base specifications of the WSS framework.

2.4.1 WS-Security, WS-Policy and WS-Trust

WS-Security [5] defines mechanisms for adding confidentiality and integrity protection to SOAP messages. It leverages the existing XML Encryption and XML Signature standards and defines how these standards can be utilized in SOAP messages by including their elements in the message header fields. In this way, intermediaries handling the message can be enabled to encrypt or decrypt parts of the message or enforced to validate the message's integrity before processing it further. In addition, WS-Security defines a generic mechanism for attaching security tokens to SOAP messages and specifies a way to reference these tokens. The design is extensible so that different kinds of tokens can be used, such as SAML assertions or X.509 certificates.

WS-Policy [12] describes a generic way to describe and exchange different policies related to Web services. For example, senders and receivers can specify their own requirements, capabilities and used algorithms to other parties. The WS-Policy specification only provides a logical framework for describing policies. Additional specifications extend WS-Policy to support actual policy usage scenarios. Most notably, WS-SecurityPolicy specifies ways to define security assertions and attach them to particular subjects, and WS-MetadataExchange specifies how metadata information can be defined and attached to service endpoints.

WS-Trust [8] specifies ways to establish trust relationships between different parties. It introduces the concept of a *Security Token Service* (STS), which is a Web service that can issue and validate security tokens. In addition, it can convert a security token of one type into a token of another type, which enables trust to be brokered between domains using different security mechanisms. WS-Trust also specifies a simple request/response protocol for obtaining security tokens. The *Request Security Token* (RST) message contains the type of the request (Issue, Cancel, Renew or Validate) and the type of the token to be requested. The *Request Security Token Response* (RSTR) message is sent as a response, containing the requested token if the requester is authorized to receive one. By using the security features of WS-Security and the policy negotiation capabilities of WS-Policy, WS-Trust provides a basic building block for establishing trust that supports a number of different use cases and underlying security mechanisms. [2]

2.4.2 WS-Federation

WS-Federation [7, 3] builds upon the base WSS specifications to define mechanisms which enable resources to be shared securely between different domains. The party managing access to a secured resource is called the *resource provider*, which makes access decisions based on claims made by the identity provider. The IdP in WS-Federation is basically a STS that issues identity tokens to users based on their authentication.

In addition to the IdP service, WS-Federation defines a number of other specialized federation services that are based on the generic STS model. The *authorization service* makes authorization decisions based on claims presented by the requestor. For example, it might allow a requestor to access a specified resource after validating its identity token. It does this by returning an access token in a RSTR message. The *attribute service* allows a resource provider to request additional information about a requestor, either to personalize the service or to use the information to make an authorization decision. The attribute service might have an interface to an existing attribute repository, such as an LDAP directory. In addition, the release of information can be carefully controlled based on requestor authorization and privacy rules. Finally, the *pseudonym service* is a special type of an attribute service which enables alternate identity information about a user to be maintained. It maps the user's identity into a pseudonym that can be used by a resource provider to identify that user. This allows the user's identity to be recognized by the resource provider while keeping it separate from the original identity held at the IdP, thus maintaining the user's privacy.

Once a federation agreement has been established between organisations, they need to share various configuration information related to the common services and policies to be used. To help exchange this information securely between the participants, WS-Federation defines a metadata model and a document format for publishing metadata about the services. It should be noted that the metadata specified by WS-Federation is meant to be used in addition to other service metadata, such as that defined by WS-Policy, and does not replace it.

The general federation model of WS-Federation builds on the WS-Security and WS-Trust specifications, which are meant to be used in a Web services environment. However, in the case of an individual user, the mechanisms for requesting and issuing security tokens have to be performed within the limitations of only a standard Web browser. For this reason, WS-Federation defines a profile specifically for so-called *passive Web requestors*. The profile consists of a model for performing the WS-Federation operations using only general HTTP technologies, such as POST, GET and Redirect messages, as well as cookies. Using this profile, all the functionality of WS-Federation can be achieved in a Web browser context. The necessary information is exchanged using GET query strings or POST parameters, with underlying SSL/TLS to secure the traffic. If needed, the requested security tokens can also be transported as references, which can later be resolved by the resource provider by requesting the actual tokens directly from the IdP.

3 Discussion and Comparison

Next, the solutions introduced in Section 2 are discussed and compared in terms of some key aspects related to federated identity management systems.

3.1 Scope and Functionality

Different types of organisations can have very different needs for doing identity federation. It is clear that these cannot all be addressed by a single solution. Instead, various federation systems currently exist for different purposes, and an organisation should select a system with a suitable scope for its needs.

SAML defines general mechanisms for doing SSO and identity federation, and specifies a number of profiles and bindings to execute those scenarios in various environments and use cases. In addition, it defines a common metadata model to enable federating partners to exchange various configuration information between them. In this way, it is a really comprehensive and extensible specification. However, SAML is more of a general framework and building block for other identity management systems, and not really a complete solution by itself.

The Liberty Alliance framework adds additional functionality on top of SAML and defines rules and guidelines for implementing the specifications and forming federations. It is mainly targeted for business interactions. Organisations belonging to a circle of trust can federate identities in a distributed manner. Users can have separate accounts with multiple IdPs and SPs, and these accounts can be linked by using opaque user handles. SSO can be achieved after a federation has been established between an IdP and SP. In addition, the circle of trust can include multiple IdPs, which enables users to authenticate with any one of them in order to access protected resources at a SP. These basic functions can be achieved with only a standard Web browser. In addition, the ID-WSF in Liberty enables various identity Web services to be developed, which further increases the usage scenarios that can be achieved with Liberty by enabling also machine-to-machine interactions.

Shibboleth is also based on the SAML specifications. It was developed mainly to address the needs of universities and research institutions, and is therefore smaller in scope compared to Liberty, enabling only SSO capabilities between organisations. Shibboleth uses a centralized model where user identities are stored centrally by IdPs, and a user has one home organisation which is used for authentication. During SSO, some user attributes can also be exchanged between the identity and service provider, depending on the policies agreed between the parties in the federation. The use of Shibboleth is limited to Web browsers, since only two HTTP-based profiles are specified for carrying the protocol messages.

WS-Federation overlaps with the Liberty framework on much of its functionality. Like Liberty, it offers both identity federation and SSO functionality, and enables separate accounts to be linked using the pseudonym service. WS-Federation also supports both Web services and Web browser environments, therefore enabling various different usage scenarios. Despite their similarities, however, the Liberty framework and WS-Federation differ in the approach of the solution. The Liberty specifications are targeted only for identity federation and management, and present an overall solution for achieving it. In contrast, WS-Federation is a part of the larger WSS specification stack, relying on the underlying WS-Trust and WS-Policy features in many of the features.

3.2 Deployment Aspects

As said, SAML is more of a generic framework for federated identity management than a complete solution by itself. Even though ready-made SAML implementations exist, the SAML specifications mainly define the technical aspects related to identity federation and metadata exchange, and offer no support in establishing the actual system configurations and the needed operational agreements between organisations. Therefore implementing and deploying a federation system based on only the SAML specifications can be impractical if many organisations are involved.

The Liberty Alliance framework and Shibboleth offer more specific guidelines for deploying the systems in practice, and ready-made products exist based on both solutions. However, they still require the proper business agreements to be made between the parties, which can limit their scalability concerning the number and type of organisations involved in a federation. The Liberty framework offers a lot of functionality, including the various identity Web services that can be deployed. Its vast scope also makes it difficult and costly to deploy, therefore making it mainly suitable for commercial use between business organisations. In comparison, Shibboleth is a more lightweight system, but also more limited in the scope and features provided. This makes it appealing for some scenarios where more reduced functionality is sufficient, for example when only browser-based SSO is needed.

WS-Federation competes directly with Liberty in terms of the scope and features offered. The flexible way in which the different services in the system can be deployed can make it a more scalable solution than Liberty. Also, the general STS concept of WS-Trust is beneficial in that it enables various kinds of security tokens to be used, which allows better interoperability between organisations using different security mechanisms. However, the adoption of WS-Federation might currently be risky for organisations, since the specification is still in draft state and thus prone to changes. In addition, the overall future direction of the WSS protocol family is not that clear, and many of the proposed specifications have not yet been completed.

3.3 Usability

An important viewpoint in discussing federated identity management solutions is also to consider them from the user's perspective, since improving usability in Web-based services is one of the key motivations for deploying these kinds of systems in the first place.

All of the solutions support a standard Web browser for carrying out the main functionality. Forwarding the user to an IdP and back to the SP is done with normal HTTP mechanisms. In addition, they all offer some means for the user to select his preferred IdP, and remember this selection if the user has enabled the use of cookies in his Web browser. The Liberty framework and WS-Federation differ from Shibboleth in the SSO functionality in that they the require the user to first opt-in to a federation before SSO can be carried out, whereas in Shibboleth this is done for the users by default. However, once the user has accepted the federation in Liberty and WS-Federation, the selection is remembered and SSO is done automatically in subsequent uses. Also, the federation can be cancelled by the user at a later time in both solutions. All three federation systems support a so-called Single Sign-Out operation, where the user is logged out from all of the services in the federation simultaneously.

A main concern for the users of an identity management solution is also naturally the security and privacy provided by the system. This includes the secure transportation of all user data, and the ability for the users to control what information is released about them to the other parties in the federation. In all of the solutions covered in this paper, secure channels are required for doing the sensitive operations such as user authentication. Shibboleth offers the most functionality related to user privacy, since it allows the parties of a federation to agree on a specific Attribute Release Policy (ARP), which dictates what user attributes can be shared between the entities [6]. In addition, Shibboleth requires that only the minimum necessary attributes are shared during SSO, and that user consent is required for exchanging any additional attributes. Liberty and WS-Federation also require that attributes are released based on the user's consent, but the actual mechanisms for agreeing on the attribute exchange policies are not yet clearly specified.

4 Conclusion

The concept of federated identity management includes various standards, technologies and solutions that enable users to access multiple services in the Internet with only a single user identity. This model of identity management can benefit both the users and service providers, since users only need to remember the credentials for one account, and service providers can reduce the costs related to the management of identity information. In addition, a number of other benefits can be achieved with federated identity, such as increasing the collaboration and interoperability between partner organisations and improving the security, privacy and usability of the services.

Many solutions and technologies exist for doing federated identity management, and this paper has presented some of the most important ones currently available. SAML is a generic framework that provides the basic mechanisms for achieving SSO and identity federation in different environments. The Liberty Alliance framework and Shibboleth are both based on SAML. They extend the SAML specifications and provide specific guidelines and practices for establishing a federation between organisations. WS-Federation is a more recent specification proposed by Microsoft and IBM, along with some other companies, to enable federation capabilities as a part of their overall family of specifications that extend the basic Web services standards.

All of the solutions covered in this paper are quite large in their scope and functionality, and they also have many similarities both in their architecture and the offered functionality. Some convergence has already happened between SAML, Shibboleth and the Liberty framework, and with the advent of SAML 2.0, its expected that this convergence will continue to happen. In addition, achieving interoperability between the SAML-based solutions and WS-Federation seems to be a goal in the future development of the specifications [4]. After all, federated identity management is all about interoperability and collaboration in the first place.

Finally, various other identity federation solutions also exist that were left out of the scope of this paper, such as OpenID [10]. It will be interesting to see what the direction of their future development will be, and how widely each of the currently existing solutions will be adopted into use. In any case, it is safe to say that federated identity management will continue to be an active area of research and development in the coming years, and we will likely see more and more organisations taking federation systems of various scope into use.

References

[1] D. C. Chou and K. Yurov. Security development in web services environment. *Computer Standards & Interfaces*, 27(3):233–240, March 2005.

[2] C. Geuer-Pollmann and J. Claessens. Web services and web service security standards. *Information Security Technical Report*, 10(1):15–24, 2005.

[3] M. Goodner et al. Understanding WS-Federation. Technical report, IBM and Microsoft, May 2007.

[4] G. Goth. Identity management, access specs are rolling along. *IEEE Internet Computing*, 9(1):9–11, January 2005.

[5] C. Kaler et al. Web services security (WS-Security) version 1.0. Technical report, April 2002.

[6] R. Morgan et al. Federated security: The Shibboleth approach. *EDUCAUSE Quarterly*, 4:12–17, 2004.

[7] A. Nadalin et al. Web services federation language (WS-Federation) version 1.1. Technical report, December 2006.

[8] A. Nadalin et al. WS-Trust 1.3. Technical report, OASIS, March 2007.

[9] N. Ragouzis et al. Security assertion markup language (SAML) v2.0 technical overview. Technical report, OASIS Security Services Technical Committee, 2006.

[10] D. Recordon et al. OpenID authentication 2.0. Technical report, OpenID Foundation, 2007.

[11] T. Scavo and S. Cantor. Shibboleth architecture technical overview. Technical report, Internet2, 2005.

[12] J. Schlimmer et al. Web services policy framework (WS-Policy). Technical report, September 2004.

[13] S. S. Shim, G. Bhalla, and V. Pendyala. Federated identity management. *Computer*, 38(12):120–122, December 2005.

[14] J. Tourzan and Y. Koga. Liberty ID-WSF web services framework overview. Technical report, Liberty Alliance Project, 2005.

[15] T. Wason et al. Liberty ID-FF architecture overview. Technical report, Liberty Alliance Project, 2005.