Identity Management Systems
A Comparison of Current Solutions
Annu Myllyniemi
Helsinki University of Technology
annu.myllyniemi@tkk.fi

Abstract
Nowadays, there is a vast amount of work going on in the field of network identity management. Many people in many organizations want to solve the problem of users having to manage countless user accounts, each of which have their own identifier and authentication credentials for a user, as well as a set of attributes that describe the user’s identity. The problem applies to the service provider side as well: how to enable collaboration between different services by allowing the users combine two or more services using only one identity, so that the companies offering the services could profit from such collaboration? The user’s identities at the service providers either need to be linked, or the user has to have a single identity which he or she can use with various services.

In this paper, we present some of the current solutions to the identity management problem, and introduce three criteria for comparing identity management systems. The systems discussed are roughly divided into three subcategories to make some kind of comparison possible. Systems in one subcategory have more or less a common viewpoint to the identity management problem, whereas two systems in different subcategories might only have one thing in common: they attempt to solve the Web identity management problem in some way.

KEYWORDS: identity management, federated identity, single sign-on, Liberty Alliance, Shibboleth, OpenID, LID, OpenSSO, Windows Live ID, BBAuth, Tupas

1 Introduction
In today’s world where Web-based or Web-supplemented services are constantly gaining more ground, users have to manage dozens of identities. They need a username and password for reading their email online, ordering books from an online bookstore, posting to their blog, and so on. In addition to these kinds of frequently used identities, Internet users are often required to register for services they seldom use, and still remember at least their username for each of those services. Due to this problem and other issues related to it, network identity management has been the subject of active research since the beginning of the millennium. Many identity management systems have been introduced, some of them attempting to become officially accepted or de facto standards.

1.1 Selecting an identity management system
As the nature of the identity systems presented in this paper varies greatly, it is obviously not possible to point out a single "best" solution. When selecting an identity system, a person or organization (from now on referred to as "service provider") should consider some criteria, for example the ones introduced here, and decide in what way each criterion applies to their intended service.

We selected the criteria presented below from the point-of-view of a service provider. We considered what should be taken into account in order to choose the right type of system, as there are many very distinct systems all claiming to solve the identity management problem in some way.

First, the service provider has to decide if the system should be usable in an inter-organizational or inter-enterprise context or if the intra-organizational context is enough. This criterion should be quite straightforward to apply as long as the service provider has a clear plan of what it is they want to do.

Another significant criterion is the ease of deployment of the identity system. The systems available each offer a different set of features, some systems a very extensive one that covers for example policy issues, others a more concise one in order to keep the system simple and widely adoptable. The relationship between ease of deployment and extent is of course not one-to-one, so deployment should be considered based on the resources available instead of the desired number of features.

This leads to the third criterion, namely the scope of the identity management system: is the system intended for small-scale employment such as a cluster of services maintained by private individuals, or a large federation consisting of many organizations that require strong authentication and authorization, or something in between? The answer to this question indicates the required minimum set of features offered by the identity system. The features possibly offered by an identity system will be presented in section 2 along with the systems themselves.

2 Types of identity management systems
The identity systems explored in this paper can be roughly divided into three subcategories. This division is, as said, very rough, but it will serve as a quick reference when comparing systems. We made the division based on the prop-
properties of the existing identity management solutions: the intended domain and magnitude of the systems, and whether the solutions are open or owned by a company aiming to make a profit of acting as identity repository to other service providers. The categories are presented in the following subsections.

2.1 Federated identity systems

![Figure 1: Federated identity management](image)

In a federated identity management system, different organizations form federations or trust circles. Inside these trust circles there are identity providers and service providers that trust each other when it comes to dealing with user identity. A user can sign on at an identity provider, and after that he or she can use the services offered by the service providers in that federation without having to log in again. This is an example of single sign-on, defined by the Open Group as follows: "Single sign-on (SSO) is a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords. Single sign-on reduces human error, a major component of systems failure and is therefore highly desirable but difficult to implement" [26].

There are currently two major bodies working on open specifications on federated identity: the Liberty Alliance with their Identity Federation Framework (ID-FF) specifications [24] and Internet2 with the Shibboleth protocol and software [20, 6, 5].

Figure 1 illustrates a federated identity management system. The outer circle represents a "circle of trust" (in Liberty terminology) or a federation. Inside the circle of trust, the service providers and identity providers trust each other and what each one has to say about a given user. The users will be able to use the services of each service provider (SP) inside the trust circle with a single sign-on at one of the identity providers (IdPs), provided that they have an account at the IdP. In the figure, the lines connecting users to IdPs show at which IdP or IdPs the users have accounts. An IdP can be, for example, the Internet Service Provider (ISP) of the user, or, in the case of Shibboleth, the home university of a student.

Liberty Alliance and Shibboleth do not only provide pure single sign-on; both support the exchange of user attributes such as occupation, name and e-mail address, which can be used for example for authorization purposes. Liberty goes a step further here; it has a whole separate set of specifications for implementing identity-based Web services, the Liberty Identity Web Services Framework (ID-WSF) [25]. This will be discussed some more in section 3.3.

Both Liberty ID-FF and the Shibboleth protocol are based on the Security Assertion Markup Language (SAML), version 1.1 [15], a product of the OASIS Security Services Technical committee. SAML is an XML-based language that defines how an identity provider can exchange authentication and authorization data about a user with a service provider, without having to reveal security-critical information about the user (such as the user’s real identifier at the IdP). The dependencies between Liberty and Shibboleth and SAML are reversed in the latest SAML version, SAML 2.0, which is partly based on the Liberty Alliance and Shibboleth specifications. This enables interoperability of the technologies, and SAML 2.0 is supposed to supersede Liberty ID-FF as well as the Shibboleth protocol at some point.

2.2 Small-scale identity systems

While federated identity systems attempt to solve the problems of identity management in groups of collaborating organizations, there are many systems that solve the identity management problem in a lighter context. For example, an identity provider in a small-scale system might be able to tell a service provider “Yes, this user is the owner of this identifier” but not necessarily “Yes, the user owns this identifier and the identifier describes a person authorized to use this service”. These systems are not intended for use in environments where authentication and possibly authorization of real persons is needed (such as any e-commerce application). The purpose of small-scale identity solutions is generally to provide a simple mechanism for enabling users to declare their identity to service providers that have no previous knowledge of them. Single sign-on is a nice feature that comes with this functionality.

One of the systems that will be introduced here, OpenSSO, is an exception to that stated above; it involves an authorized (i.e. not client-hosted) identity server that authenticates users. Although it is quite different from the other systems classified as “small-scale” here, OpenSSO was included in this category because it is not a federated identity system (yet), but not proprietary either (anymore).

The open-source community as well as private individuals have developed their solutions to self-hosted identity management and Web single sign-on. Perhaps the best known systems are OpenID [18, 9], LID (Lightweight Identity) [11], and Sun’s recently launched OpenSSO project [17].

OpenID and LID are based on using URLs as user identi-
OpenID and LID URLs can also be digitally signed for ad-
enabled site would need e.g. an HTML form field to type
the ow of events in OpenID authentication.
and commands to allow the single sign-on procedures. Both
details of the verication process vary according to the used
security.
In the case of OpenID, a user coming to an OpenID-
bled site would find e.g. an HTML form field to type
their OpenID URL into. The service would then consult the
user’s OpenID server, and after verication procedures, the
user could use the service with their OpenID identity. The
user’s own computer) with their OpenID or LID identity.

In the case of OpenID, a user coming to an OpenID-
module interacts with the OpenSSO server, and after
management. Its main purpose is to facilitate adding SSO
management systems like it, depends on policy rather than the technology.
LID is more extensive than OpenID; it actually supports
Proprietary systems
In addition to the systems described above, there are many
proprietary identity management systems that offer mainly
single sign-on, but may support some additional functional-
ity as well. The thing that these systems have in common is
that they do not attempt to become industry-wide standards,
but merely serve as identity providers to a suitable set of ser-
vice providers.

Figure 2: Lightweight Identity (LID) architecture [11]

OpenSSO is an open-source version of Sun Java System
Access Manager. OpenSSO is being developed actively, and
it has been announced that in the near future it will imple-
ment the Liberty ID-FF [24] and ID-WSF [25] standards as
well. With this happening, OpenSSO would quickly migrate
to the federated identity systems category, but thus far, it
can be categorized as just an SSO system. As OpenSSO is
based on commercial software with a fairly long history, it is
quite a large piece of software. It differs a lot from OpenID
and LID: unlike them, OpenSSO does not drive self- identity
management. Its main purpose is to facilitate adding SSO
functionality to existing Web service applications. OpenSSO
functionality is illustrated in figure 4. The figure depicts a
client (browser), two Web servers and an OpenSSO server.
The numbers indicate the sequence of events. First, the user
goes to a Web server, where an OpenSSO module resides.
This module interacts with the OpenSSO server, and after
the user has logged in at the OpenSSO server, the Session-
Service module can automatically tell the other Web server
that the user has been authenticated.

2.3 Proprietary systems
In addition to the systems described above, there are many
proprietary identity management systems that offer mainly
single sign-on, but may support some additional functional-
ity as well. The thing that these systems have in common is
that they do not attempt to become industry-wide standards,
but merely serve as identity providers to a suitable set of ser-
vice providers.
The most well-known systems that fit in this category are Microsoft’s Windows Live ID [28] (formerly known as Passport) and Yahoo’s Browser Based Authentication (BBAuth) [29]. The latter was very recently opened for anyone to employ in their Web service as the authentication mechanism free of charge.

In addition to the internationally known systems such as Windows Live ID, the Finnish Bankers’ Association has a nationally significant identity system called Tupas [23] which is used in various commercial and governmental services to offer very strong authentication.

Like the small-scale systems, all of the proprietary systems listed here offer basically just single sign-on. Although the authentication methods are not necessarily technically stronger than in the small-scale systems in all cases, the authentication by these systems can generally be considered stronger. This is because the user identities are managed centrally by the owners of the services, and no one else has access to user account information (it would not be good for the business to get caught neglecting good authentication policies). It can be stated thus that the proprietary systems form an in-between option to the systems of the other two categories: they offer single sign-on with generally stronger authentication than in the small-scale systems, but the identities are managed centrally and cannot be federated.

3 Comparison of identity management systems

In the following subsections, the criteria for comparing different identity systems are discussed with regard to the three categories of identity systems.

3.1 Context

Context is the first criterion to consider when choosing an identity management system. If well-secured collaboration between organizations (i.e. circles of trust or federations) is needed, some solutions are ruled out. Liberty Alliance and Shibboleth enable the establishment of inter-organizational federations, the other solutions presented in this paper do not; they rely on a central, and possibly commercial, identity provider to vouch for the users. The different possible contexts include the following and their combinations:

1. Federation of organizations: e.g. companies or universities wanting to enable sharing of services
2. An e-commerce application
3. Semantic Web applications, such as blogs and wikis

For number one, only a federated identity management system would suffice. For number two, the safest bet would be a federated system or a proprietary system, and for number three, any of the three categories would be suitable, but a federated system could be considered exaggerated, as it would require out-of-band agreements for establishing the federations needed.

From a user’s point of view, using services is easier if many services can be accessed with a single sign-on. This does not imply a certain identity system, however, because the number of services accessible with a single sign-on depends on collaboration rather than implementation.

3.2 Ease of deployment and use

The ease of deployment of an identity management system depends on the level of openness of the system, the resources that a service provider has for the deployment (financial and human resources, time), the breadth of the system’s feature set, and the technologies used in the environment where identity management is to be introduced.

If a system with a wide feature set as offered by Liberty Alliance is required by, for example, an enterprise, and the enterprise does not have skilled personnel and lots of time to implement the specifications, it is always possible to buy a commercial implementation that comes with deployment support. For enterprises, this is often the most practical solution, as the Liberty Alliance specifications are in no way easy to understand, let alone to implement. Also, the open-source implementations available do not support all Liberty features yet.

If it is sufficient to have just single sign-on, one of the small-scale identity systems can be a good choice. The small-scale systems presented here are genuinely open-source and have good documentation. They are also less complex and thus easier to deploy than the larger systems. If a service provider wants easy deployment as well as reliable functioning, a commercial solution might be in place. Depending on the desired scope, a commercial solution might be a federated identity management system, or a proprietary system such as Tupas.

A client to the services, of course, does not care how much effort it took the service provider to get the identity system working. For the user, usability is what matters. All the identity management systems presented in this paper are browser-based, so there is not necessarily a lot of difference between the different solutions. The user experience depends
on how the service provider has deployed the identity service rather than on the underlying solution.

3.3 Scope of the identity management system

The scope of an identity management system can be anything from very basic single sign-on functionality to an entire framework that provides definitions of for example discovery service and user attribute exchange.

The system with the widest scope currently is the Liberty Alliance specification family. As mentioned in section 2, the Liberty Alliance specifications also include the Identity Web Services Framework (ID-WSF) [25] which builds on top of the kind of identity federation protocols and profiles specified in ID-FF to define a standard way of for example exchanging user information and finding identity service providers. In addition, there are other specification sets that compliment ID-FF and ID-WSF, for example templates defining what user data might be shared in different domains and applications.

The Liberty specifications have been developed in phases illustrated in figure 5. Phase one constituted only the ID-FF specifications, which form the core infrastructure for federated identity systems. In the first phase, ID-FF was based on SAML 1.0. The second phase introduced ID-WSF 1.0, which still relied on the ID-FF specifications. In the third and latest phase, ID-WSF 2.0 has been published. It is based on SAML 2.0 rather than the ID-FF specifications, due to the fact that SAML 2.0 will most likely replace Liberty ID-FF, as explained in the last paragraph of 2.1.

![Figure 5: The phases of the Liberty Alliance [10]](image)

Shibboleth is currently only used in the academic world, such as in the Haka system of Finnish institutions of higher education and in the American InCommon system. So far, it is only used on national level, but attempts are being made on international collaboration. Shibboleth is not quite as extensive and all-round as Liberty Alliance, but for domains in which the parties have a lot in common, like the universities in the Haka system, it is a very noteworthy alternative. We might argue that in practice, the scenarios where all of Liberty’s flexibility would come in use, are quite unlikely.

Next in the width of their scope come the proprietary systems like Windows Live ID, Yahoo! BBAuth, and Tupas of the Finnish Bankers’ society. Although they, like the small-scale systems, do not usually offer more than single sign-on, the authentication by these systems can be considered stronger because the identity providers know more about the user than just their username, and the identity information is managed by third-party service providers rather than the users themselves.

The strength of the authentication offered by an identity system varies and is also implementation-dependent. Even with a small-scale system, it is possible to get more than just the assertion “this user is the same user that logged in my service a while ago”. Of course, extending the small-scale systems requires competence to implement such extensions.

The solutions classified as small-scale systems in this paper offer the narrowest set of features, but as the systems are further developed, it might not be appropriate to describe a system as small-scale anymore. Thus, it is always necessary to review the current situation of each solution before passing judgement.

For a user, a more elaborate system might initially seem annoying with the constant asking for consent and, in the case of Liberty, the possible need for linking his or her account at the service provider with the account at the identity provider. However, the consent decisions and account linkings can usually be saved so that the user won’t have to repeat the procedures after the first time. After these initial annoyances the user will probably find the system quite convenient to use. The smaller-scale systems also require some amount of extra clicking in the beginning, so there might not be that much of a difference in user experience in comparison with federated identity systems.

4 Conclusion

With the variety of identity management systems around, it is necessary to categorize them somehow, and only then make decisions of potential deployment and usage. When selecting an identity system, the candidates have to be estimated based on the intended use of the system and the resources available for its deployment and management.

Each identity management solution has its benefits and downsides. Liberty Alliance is very versatile, but the specifications are written in collaboration of many members and therefore tend to be somewhat bloated and discursive, which makes the Liberty solutions difficult to understand and adopt. Commercial implementations of the Liberty specifications might be the only practical choice. Shibboleth on the other hand seems to be stuck in the academic world and inside national borders for now, although it could offer a noteworthy alternative for Liberty, were it to gain wider popularity.

Proprietary systems such as the Finnish Tupas might be very good when a service provider only wishes to facilitate identity management for themselves and their clients, without the costs and benefits of acting as an identity provider as well.

The small-scale identity systems are neat and easy to grasp, but they can only offer so much features and are not always extensible as such. Their goal is to solve one problem – and one problem only. This is of course a generalization, but it is necessary for the reader to realize that the small-scale systems do not even attempt to compete with Liberty Alliance or Shibboleth; their scope is intentionally different from that of the larger systems.
This paper is very much subject to early ageing, because the technologies discussed in here are all being further developed. Some of the now small-scale systems will convert into federated identity systems, or at least the feature sets offered by them will become so extensive that in many cases, the context and ease of deployment are the only criteria there is left to look at, and the scope cannot easily be compared anymore.

The technologies are not only developing, but also converging. With SAML 2.0, Shibboleth and Liberty Alliance will be interoperable. There is also a lot of collaboration going on in the open-source community and the small-scale identity systems; for example, LID currently supports OpenID as one of its features. Projects that started as small and independent are becoming parts of larger projects, which is a very good sign of progress. Collaboration is essential if we want to have working systems available.

As an important conclusion, this paper should underline the importance of the criteria introduced in section 1.1. The criteria is the essential thing to look at; finding information on the current solutions is left to the reader. The solutions introduced here serve as a starting point. If the reader wishes to deploy an identity management system, the criteria presented in this paper can be used as a guideline.

As implied above, for many reasons it is impossible to cover the whole area of identity management systems in a single paper. Section 5 as well as the references given throughout the paper act as pointers to more information. It is a good idea to visit the Web pages of the technologies presented in the paper, as they will always contain the most up-to-date information, and in many cases links to other similar projects going on at the moment.

5 Further reading

Various articles and papers have been written on different aspects of identity management.

Identity management as a concept is discussed in [3]. The article is really an introduction to the topics of an issue of IEEE Internet Computing, but it summarizes the problems of identity management well. Subenthrinan et al. [22] discuss the requirements of an identity management system from the end users’, as well as the identity and service providers’ point of view, in contrast to this paper, which emphasizes the providers’ perspective.

Gaedke et al. [7] propose a "building block" solution for identity management systems. It is written from the perspective of the federated identity model.

Goth discusses the competition situation in the identity management market, as well as the need for the different identity management specification bodies to work towards interoperability in [8]. Goth also notes in the article, very accurately, that in the end it is the users that determine if an identity solution is to become a standard or fade away.

Those interested in learning more about the definition of "digital identity" might want to look at [4].

The identity management solutions introduced in this paper are not the only ones that have been proposed. Other proposals for identity management systems have been made, some of which include the ones presented in [2], [21], and [1].

The existing identity management solutions only solve the single sign-on problem for one protocol, namely HTTP. It would obviously be very useful if single sign-on and the other aspects of identity management systems could be extended to work in other protocols, such as SMTP (Simple Mail Transfer Protocol) [16] or SIP (Session Initiation Protocol) [19], as well. Indeed, there are instances working on how to combine identity management solutions with non-HTTP protocols. For example, an Internet draft by Tschofenig et al. [27] specifies how to carry SAML assertions (see 2.1) in SIP messages.

References


