Reducing the integration tax of cross-organizational identity-based applications with identity federation platforms. A case study of integration of JBoss Application Server in Liberty Alliance Platform

Antonio Manuel Fernández Villamor
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid
antoniofer@dit.upm.es

Juan Carlos Yelmo García
Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid
jcyelmo@dit.upm.es

Abstract

The Internet has become an incomparable communication channel to reach old and new customers and to offer innovative services. Due to the increasing interest in Internet-based services, enterprises are trying to make the best use of the advantages provided by an online presence. Moreover, they collaborate in order to provide cross-organizational identity-based services, giving an added value to their traditional services. This poses new challenges regarding identity management between domains. An option to overcome them is to integrate an identity-federation platform with that type of services, but it is a very complex task. In this paper we propose to extend the capabilities of an Open Source application server in order to make it compatible with an identity-federation platform as a basis to support cross-organizational identity-based services, reducing dramatically the integration tax.

1. Introduction

Nowadays, it is very unlikely that enterprises do not utilize websites that enable them to inform potential customers about their products, make them aware of new services, and try to attract them away from the competition. The influence of Information Technologies (IT) upon the marketplace is ubiquitous. For this reason, companies are improving their IT infrastructures to enhance relationships with customers and with other enterprises or organizations. In the last few years they have been promoting a variety of initiatives, from creating new internet and telecommunication services to core enterprise renewal, all of which are focused on one thing: to offer as many opportunities to customers as possible, in order to attract them.

An emerging type of new services is one that implies collaboration between several entities and, on many occasions, the establishment of new ways of managing their customers.

One of these is financial aggregation services, which try to aggregate information provided by different account holders into a single web application. The simplest financial aggregator shows information from account holders (bank accounts, credit cards, utility suppliers, insurance companies, on-line shops, etc.) together in one single web page, making it convenient for users. It can also process that information and show expenses sorted into categories or let the user realize transactions.

That financial aggregator could be extended with the establishment of collaborative fidelity programs between banks, airline companies, hotels and car rental enterprises. Their purpose is to provide added value over their traditional services, by means of showing enriched information (aggregated expenses, for example) or offering new possibilities (such as buying an airline ticket in one-click, displaying personalized rebates from the store, hotel + car rental offerings, etc.). As other examples we can also consider parcel tracking information services, which involve collaboration between a manufacturing enterprise and its logistics partners. The value of all these services is the sum of what they are composed of.

All of them are identity-based services as they depend on user's digital identity, which has been defined [1] as the collection of data about a subject that represents his profile, his preferences, his traits and his attributes. Moreover, they involve some kind of identity attributes sharing to accomplish their work as their business logic is distributed among several

organizations. In this paper we shall call them cross-organizational identity-based (COIB) services and applications.

One of the first examples of COIB services related to business we can find appeared in 1960s in the credit card industry in the USA [1]. Bank of America launched a successful credit card called BankAmericard but, to compete with other banks, Bank of America franchised the card. In that moment, it lost its direct relationship with its customers and had to establish new mechanisms to verify cards and customers' identity and to perform transactions. In that sense, they established what we call today an identity federation network.

Nowadays, financial institutions see on-line banking as another kind of distribution channel to reach new and existing customers. Recent studies [2] show that, while in the year 2000 there were over 30 million on-line banking customers in the world (more than 15 million in Europe), in the year 2004 that number increased to 120 million all over the world (nearly 60 million in Europe). That this increase is so dramatic in so short a period of time should prove to be a wake-up call to banks and financial institutions: the Internet should not be underestimated. However, users' perception about Internet security is a major factor inhibiting a wider adoption of such new services and so any initiative capable of making customers feel more comfortable and secure will be successful. Considering the example of financial services we can notice that in the year 2000 only 14% of customers used Internet financial services for transactions and information, while in year 2004 this number rose to 25%, which, while representing an increase, is still lower than desired [2]. One of the main problems with Internet financial services is general consumer distrust since on-line transactions account for the highest fraud rates, including identity theft. As an example: in the second quarter of year 2005 there were more than 2800 *phishing* websites detected [3].

In this paper we propose the use of an identity federation platform to manage distributed user identity on an example cross-organizational identity-based service, and define technical guidelines to integrate existing enterprise applications and services with this identity federation platform effortlessly, reducing the integration tax.

2. Real world business scenario

Nowadays, the most common situation regarding digital identity and Internet services is that every enterprise and organization has its own user database. This information is at the core of the relationship of a company with its customers and partners and is the foundation that supports its business models. They store information about users' identity, attributes and credentials across their applications and services, but with no kind of cohesion with the others'. When someone wants to book a flight he needs to log into the airline server and introduce his home address, credit card number, etc. The process is repeated when booking a room in a hotel or renting a car for a couple of days. This is quite inconvenient for an average user. Also, this private data scattering creates a distrust feeling among users.

It would be very interesting if, for example, the bank provided a COIB application to manage all these transactions from one centralized point, relieving the user from the need of logging into several services.

2.1. The problem of identities in a COIB service

Difficulties arise when COIB services are established across several organizations as they don't share, for security and privacy reasons, their user accounts. They are separate organizations, with different policies, legal requirements and security domains. "How to identify the user?", "How to share his attributes?", "How to warranty privacy?" are common questions that surround these distributed services.

An option for the bank's application would be to ask the user for his credentials in the airline, the car rental company, the hotel and the online store and store them. That way, the bank's application would be able to impersonate the user and carry the necessary transactions in the other partners' services. However, it is clear that users will feel that their private data is unprotected.

Another course of action for the bank and its partners would be to implement an identity federation platform to manage digital identities on their COIB services. This will allow them to link the users' identities across the servers of the alliance and provide cross-domain authentication across these services without violating the most basic principles of privacy.

Identity management is a specialized area of knowledge dedicated to the study of how resources (a server, an application, a person browsing the Internet…) can be identified uniquely. It also deals with access management in legal, technical and business aspects. It can help us to set up methods, processes and guidelines to manage access, security and privacy of the user and his information. The right identity management approach is a critical issue regarding sensitive business services like financial

ones and more especially if they involve identity attributes sharing like COIB ones.

There are two main approaches of identity management: centralized and decentralized. In a centralized approach, identity management is delegated to a single point inside the enterprise, as there is only one administrative domain. For example, every department of an enterprise could delegate their Authentication, Authorization and Accounting (AAA) systems to a common centralized directory. By definition, in a COIB service there are many identity domains interoperating (the bank's, the airline's, the store's, etc.). Its own requirements make it impossible to implement a centralized identity management point as it would force modification of the whole network and services architecture to enforce that unique directory.

On the contrary we have the decentralized identity management approach. It tries to solve all these issues, related to the natural isolation of identity domains that enterprises and organizations have. There is a wide array of solutions and implementations available.

3. Decentralized identity management initiatives

Latest identity management trends encourage the introduction of decentralized solutions in order to share users' identity information between different domains maintaining privacy and security. There are two decentralized identity management tendencies: user-centric and federated.

Some user-centric initiatives are OpenID (http://openid.net) and CardSpace [4], among others. OpenID is a decentralized digital identity system. Through the use of a URL (or XRI, extensible Resource Identifier), it allows the user to lay claim of a digital identity that can be verified by a Web application. It also allows the sharing of authentication information across entities. In an OpenID-enabled website users do not need to create an user account to gain access but they must handle an identifier created by an OpenID verification server known as Identity Provider (IdP). OpenID is very focused on a light-weight security for the general Web world, where security criteria is not very strict, based on the idea of "trusting someone unknown because someone else tells me to trust him". Some authors have criticized its security strength under certain attacks [5][6], although works to resolve those issues has been carried in OpenID 2.0. With somewhat wide adoption in forums and blogs, OpenID is spreading rapidly, although enterprises feel uncertain about its use in business environments.

Windows CardSpace is a Microsoft's identity technology based in .NET Framework 3.0 which improves and simplifies access to secured resources and allows the sharing of personal information over the Internet. It is based on the concept of ID cards, each of them is related to an Identity Provider, as in the real world. To become CardSpace-compatible, a website needs to specify a HTML tag that asks the user about his identity and about some data. The user selects which card he wants to use with each website. CardSpace is the one responsible of getting a security token and sending it to the website.

The other kind of decentralized identity management is the federated approach. Service providers agree to link their domains to allow users to access their services with only one authentication. There are several identity federation platforms nowadays, for example, the implementations of standards, specifications and good practices, of Shibboleth (http://shibboleth.internet2.edu) and Liberty Alliance (http://www.projectliberty.org). They promote the use of Security Assertion Markup Language (SAML) [7] and related technologies for identity federation

SAML, a standard XML-based language developed by OASIS, provides a means for exchanging security information between partners. There are two kinds of actors in a SAML exchange, one playing the role of Relying Party and other playing the role of Asserting Party. The Asserting party asserts authentication and authorization information about a subject. The Relying Party may use that information for taking security-related decisions about that subject. This way, SAML and, consequently, Shibboleth and Liberty, provides a means of exchanging identity information between different domains.

Shibboleth is an Internet2 initiative. It has created an architecture and Open Source implementation for a federated identity environment based on SAML. Shibboleth is focused on academic and university world, while Liberty Alliance is made up of many enterprise-class members.

Liberty Alliance is a consortium founded in the year 2001 to promote standards, guidelines and good practices for a framework of identity management. Their main focus is identity federation. Nowadays more than 150 enterprises and public organizations are members of Liberty Alliance: AOL, Telefónica, British Telecom, France Telecom, Hewlett-Packard, Oracle, Sun Microsystems, Ericsson, etc. It must be highlighted that the Liberty's platform is enterprise-class in terms of security, stability and scalability. Also, it can cope with hundreds of thousands of federations, very appropriate for a business environment.

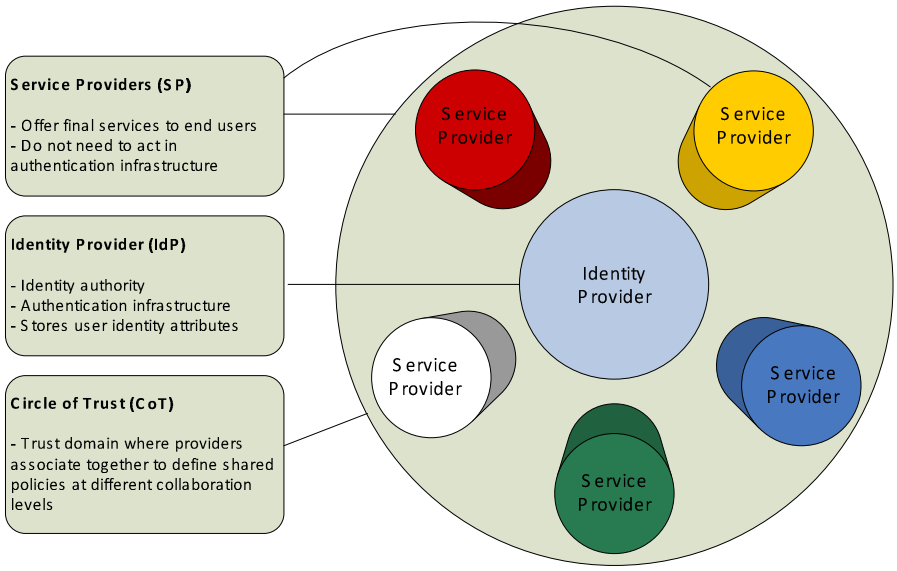

Figure 1. Liberty's Circle of Trust

Liberty is based on the concept of Circle of Trust (CoT) (Figure 1. Liberty's Circle of Trust). The CoT is a trust domain where several Service Providers (SPs) are associated together, defining collaborative business, identity, trust and quality of service policies, and governed as specified by Liberty. Users making use of services from providers inside the CoT may federate their accounts that, in general, are independent. There is also another agent inside the CoT, called Identity Provider, which is responsible for federation management. It is also the main authority for identifying and authenticating users and stores neutral user identifiers, so that each SP doesn't know usernames from the other SPs.

Liberty proposes works in three phases for technology adoption:

Phase 1 – Identity Federation Framework (ID-FF). Defines protocols for federating user accounts, for Single Sign-On and Single Log-Out and for Name Registration inside federations [8]

Phase 2 – Identity Web Services Framework (ID-WSF). It is a platform based on Web Services for the development of identity-based Web Services, such as description and discovery of services, authentication, access to shared attributes, user interaction to obtain special rights, etc. [9]

Phase 3 – Identity Services Interface Specifications (ID-SIS). Defines specific identity-based services that use the platform defined in ID-WSF. Some of the defined profiles are personal profile, employee profile, contact list, presence profile, location, etc. [10]

One of the main differences between the federated approach (Liberty, for example) and user-centric (OpenID, for example) is the nature of the CoT and the IdP's ownership. Liberty proposes a closed CoT where all agents must previously agree to participate, enforce the identity federation services and share common policies. Also, the Liberty's IdP is an entity owned by one (or several) agent of this CoT. For these reasons, trust and security is stronger in Liberty's model and so is very well focused for business environments. Of course, there are many more differences between federated and user-centric identity management, but they are out of the scope of this paper.

4. Integrating COIB services with an Identity Federation platform

It is not trivial to integrate services with an identity federation platform. They usually offer implementations or programming APIs to ease the development, but it is necessary to modify the current services and applications' code to adapt them to the identity federation platform's specifications. This could carry long and tedious development and testing processes. Such integration tax is often a serious handicap, as these are very sensitive and mission-critical services. It is an issue that impacts the whole enterprise. It is not conceivable that a bank or an airline company modifies sensitive services easily. Finally, it is also a burden for third parties to implement identity management solutions and join the CoT.

Our objective is to provide a way of integrating existing services and applications (the bank's and its partners') with an identity federation platform, Liberty. The proposal is to extend the capabilities of the application server in which the applications are running

so that is the server, and not the applications, the one in charge of the integration with the identity federation platform. This way, the business logic remains detached from the security logic. This hides the complexity of the federated authentication processes to applications and to development teams, reducing the integration tax.

SourceID ID-FF 1.2 Java Toolkit [11], an Open Source platform of Liberty Alliance's specifications, offers a SP and an IdP as web applications ready to run on JBoss Application Server. It is a good starting point to integrate existing applications in a Liberty SSO environment and supports the core profiles including Single Sign-On and Single Log-Out, Register Name Identifier, Federation Termination Notification and Identity provider Introduction. It presents some advantages:

- It is Open Source: we can study its source code and modify it to meet our needs
- Runs on JBoss, which also allows modification
- It is quite complete in terms of Liberty's specifications ID-FF, it includes a SP and a stand-alone IdP

However, there are more Liberty implementations, such as Lasso (http://lasso.entrouvert.org) and openLiberty (http://www.openliberty.org). Lasso is implemented in C (which may represent a disadvantage as Java-based applications are dominant in internet services). OpenLiberty aims to offer a programming API for ID-WSF.

JBoss Application Server is one of the most successful Java application servers (around 30% to 40% of market share [12]), although it is difficult to estimate such market share [13]. Moreover, it is Open Source, which gives us the possibility to extend it to support a new authentication method compatible with Liberty.

4.1. New JBoss authentication method

Web enterprise applications are usually divided into three tiers: presentation, business logic and data access. Security regarding user authentication (and basic authentication) is enforced at the presentation tier. It must verify the user is the one he claims to be and check if he has enough permission to access a particular resource. That information about the user is passed to the next two tiers by means of the server's applications container. Although authentication and authorization tasks can be carried out by the presentation tier itself, the container provides standard methods to do so. The application just needs to select which one to use and configure a few parameters, such as which users are allowed or which resources are protected.

JBoss includes four standard methods:

- BASIC: the user needs to introduce his username and password in a browser's pop-up window to authenticate to the application
- FORM: in this case, a more complex web page is presented to the user asking for some kind of information relative to him. Usually that information is also a username/password pair
- CLIENT-CERT: the user authenticates himself with a client certificate issued by a trusted authority
- DIGEST: similar to BASIC but the username and password are encrypted during transmission

Web applications include configuration files that may allow isolating the application logic from particular authentication configuration. In those files the developer chooses which method of authentication he wants for his application. We have developed a new authenticator that can be selected the same way the default ones. This way the application delegates the authentication and authorization tasks to the server.

JBoss has a simple mechanism that allows us to develop new authentication methods and to employ them as the standard ones included by default with the server. By reusing part of the current authentication code included in JBoss 4.2.1, it is possible to create a new authentication module that works as one of the default ones. After that, it is only necessary to register it in the server configuration (we have called it LIBERTY) to make it available to applications.

The Liberty JBoss authenticator has been developed using part of SourceID's SP code. It is divided in several parts:

- Presentation layer, mainly based in jsp pages
- Security, based in Java Servlets. Security in the original SourceID's SP application is not delegated to the JBoss server
- Java support classes that provide means to work with Liberty protocols, SAML assertions, etc.

Only the last has been of use in the creation of the new authenticator. When an application is configured to use LIBERTY (Figure 2. JBoss' authentication extension) as authentication method, every time the server receives a request to serve a resource, it first passes the request to the Liberty authenticator. If the user is already authenticated, then JBoss can serve the resource. In any other case, it redirects the user to the IdP for authentication. The IdP generates a SAML token that will be presented to the Liberty authenticator embedded in the HTTP request. The user's browser is redirected to the resource again and, in this case, the

Liberty authenticator may allow the request after analyzing this SAML assert.

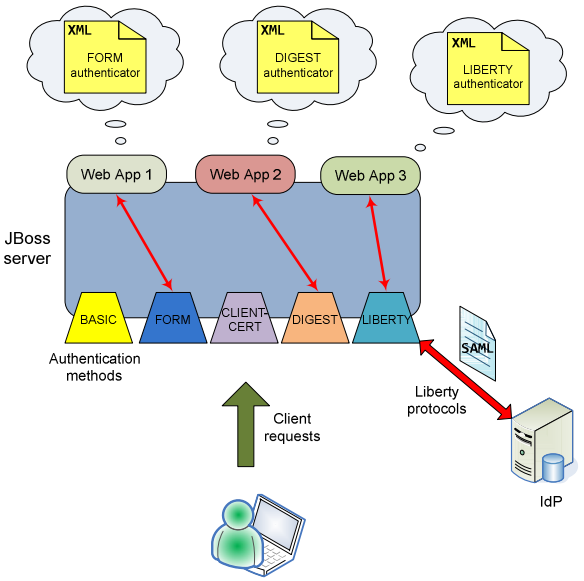

Figure 2. JBoss' authentication extension

Obviously, this new Liberty authenticator needs some extra information in order to work as a complete SP. It must know where the IdP is hosted, which Login and Logout profile to use, etc. All this information is stored in XML configuration files in JBoss' conf directory. Also, information about federations is stored in a local database.

4.2. Setting up the COIB service

The bank and its partners have their applications and services running on their servers as usual, configured to use one of the four standard authentication methods JBoss provides. The best identity management option they have in order to offer the COIB application is, as we have seen before, to integrate his services with Liberty. They should follow these steps:

- Setup an IdP. This is a low-effort task as SouceID includes a stand-alone IdP. The ownership of this IdP could be shared among the partners
- Install the Liberty JBoss authenticator in each JBoss running the services they need to "share". It is necessary to copy it to JBoss' library folder and register the authenticator as "LIBERTY" method in a XML configuration file. This task should not take more than a few minutes. Now, every JBoss server becomes a Liberty SP.
- Configure each SP and the IdP (tell the SPs where the IdP is hosted, which Liberty profiles to use, etc.)
- In a real scenario, many enterprises and public organizations store their AAA information in a Radius server. In order to take advantage of this infrastructure, the IdP could validate user credentials with the information stored in the Radius server. Also, it is possible to have a detailed accounting of the employ of COIB services. As an optional feature we have delegated IdP authentication to a Radius server

The whole process of integration of the COIB application with Liberty is very simple, needs very little effort and does not require new developments. This integration allows the user to Single Sign-On across all those services and share sensitive information inside the CoT with its partners, while maintaining privacy. Inside a Single Sign-On session the COIB bank application can access the other services without asking the user for authentication. Also, as Liberty uses neutral identifiers in the IdP, user credentials remain private. Moreover, the applications and services that are running on the partner's infrastructure and do not take part in the COIB application are not affected at all.

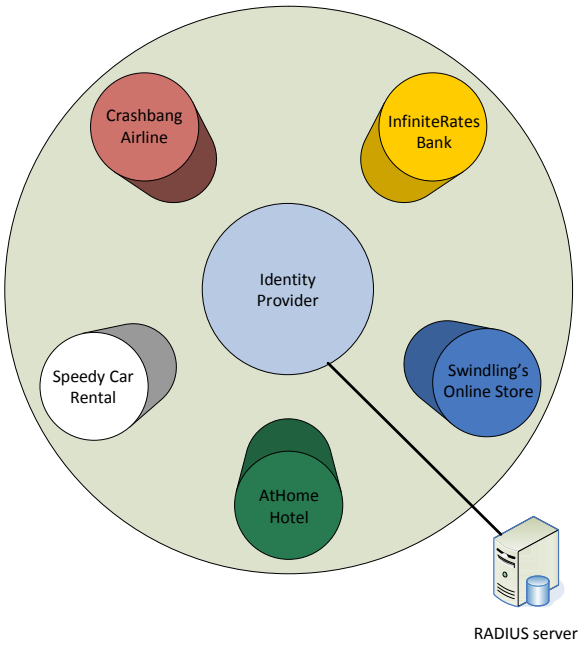

Figure 3. COIB application's CoT with Radius-delegated authentication

4.3. Authenticating to a cross-organizational identity-based applications with Liberty

Once all the partners have carried on the necessary steps and the COIB application is running inside a Liberty's CoT (Figure 3. COIB application's CoT with Radius-delegated authentication), any user connecting to the bank server would be given the opportunity to participate in this experience, with no loss of privacy and security regarding his sensitive data across the CoT. With his agreement, the user's accounts are federated (linked together) and he is given a new username in the IdP. This user federation process may be automatic or human-assisted. In our prototype, we have embedded the necessary code in the SPs to perform automatic user federation.

Every time the user wants to use this collaborative application, the JBoss Liberty authenticator checks if the user is logged on the CoT. If not, is redirected to the IdP for authentication (Figure 4. Authentication workflow of a COIB application).

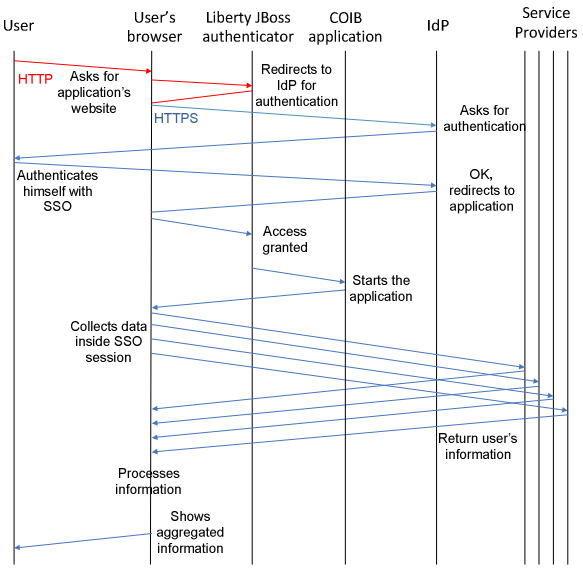

Figure 4. Authentication workflow of a COIB application

All redirections are carried out through the user's browser. When the user has authenticated to the IdP is redirected again to the application. This time, the authenticator grants access and allows the application to be run on the user's browser. It now connects to all the necessary Service Providers to collect data or execute actions. Every provider, which already has the Liberty authenticator, will grant automatically access to the application (according to the particular SP's authorization policies) as the user has initiated a SSO session. Each SP remains agnostic of what is going on with the other providers. Also, transactions are realized point-to-point from the application running on the user's browser over HTTPS, which warranties privacy of user's sensitive information. At this point all interactions occur between the application and the partners' SPs until the user logs-out from the CoT. Single Log-Out works the same way: when the user desires to finish his SSO session the application informs the IdP of that action. Then it invokes a Single Log-Out URL on each SP redirecting through user's browser.

It must be highlighted that at no time the applications and services process Liberty protocols nor identity messages. It is the servers which handle that logic. Last but not least, the authentication process does not hassle the user as he is only asked once.

4.4. Authorization in the COIB service

Each service provider may have configured authorization policies for users in their particular web applications. These policies may comprise basic rules of which group of users (usually called "roles") is allowed to access a particular URL or more complex ones closely related to business logic (for example, do not allow transfers of more than 1000€ if the user is under 18 years-old).

JBoss Liberty authenticator does not interfere with authorization tasks. Employing Liberty authenticator or another one provided by JBoss makes no difference for the applications in terms of identifying the user as they are always identified as local one. This allows applications to use the same authorization policies than usual while allowing Single Sign-On across the CoT.

5. Future work

Web Services provide a new very successful approach for creating distributed applications. They define standard communication protocols between services. On the other hand, Liberty ID-WSF defines a platform based on identity-based Web Services working with two different roles: Web Services Consumer (WSC) and Web Services Provider (WSP). Both of them are affiliated to a Liberty CoT. ID-WSF adds more entities to the CoT, for example:

- Discovery Service (DS): works as a service directory and stores user identity data and the necessary credentials for a WSC to authenticate to a WSP
- Authentication Service (AS): authenticates a WSC to the DS

Current Open Source implementations of ID-WSF allow Web Services to play the role of WSP or WSC, but they need to be adapted to the ID-WSF API. However, similar work to the one described in this paper could be carried on with ID-WSF, adding functionality to an applications container so any existing Web Service could play the role of WSC or WSP with no need to modify its code. A federated identity Web Services environment would help to provide more flexible COIB services and applications to users and enterprises. Also, it may allow setting up CoT-wide security policies.

On the other hand, it may be useful to develop a central CoT management point in order to improve and simplify the CoT's administration. This would allow to manage configuration, users, service provisioning, etc. in a centralized fashion, reducing complexity and saving costs and time.

6. Conclusions

In this paper we have highlighted the importance of Internet services as a means to maintain (or establish) new relationships between enterprises and their customers. Organizations are trying to do their best in this sense and have created innovative services. One very popular emerging kind of those services is the cross-organizational identity-based services, which allow the development of new possibilities and improve usability for customers. However, the inherent difficulties of identity management in Internet services are especially increased when trying to collaborate between isolated domains, as in COIB services.

Currently there are identity management initiatives that define guidelines, protocols and standards to ease these difficulties and provide new possibilities. For example, Liberty Alliance proposes to federate identities between different domains so that a user may access distributed services from several domains in a joint fashion maintaining private data secure.

However, the integration of an identity federation platform into the infrastructure of an enterprise to support COIB services may be an arduous work. The integration tax is very high and many enterprises may refuse to carry on such integration.

We have developed an authentication extension of a common Open Source application server (JBoss) that provides compatibility with Liberty specifications. Applications running on that server may join the federated identity environment with no modification at all. This reduces dramatically the integration tax of COIB services in a federated environment.

With our approach, Liberty-related processes and procedures are isolated from applications' business logic. The services that compound the COIB service do not handle Liberty protocols and, what is more important, they only have access to the private data they need.

7. References

[1] P.J. Windley, *Digital Identity. Unmasking Identity Management Arquitecture (IMA)*, O'Reilly Media Inc., 2005.

[2] H. Geiger, "Online Banking: Evolution and Position" SwissBanking Media Seminar, June 2005, www.swissbanking.org/mediasem8-6-05_geiger_e.pdf.

[3] N. Walters, "Gone Phishing: The Internet and Identity Theft. Research Report.", AARP Public Policy Institute, on-line. June 2005, http://www.aarp.org/research/frauds-scams/fraud/fs118_phish.html

[4] D. Chappell, "*Introducing Windows CardSpace*", April 2006, http://msdn2.microsoft.com/en-us/library/aa480189.aspx

[5] T. Anderton, "OpenID still open to abuse". IT Week, on-line. March 2007, http://www.itweek.co.uk/itweek/comment/2184695/openid-open-abuse

[6] M. Slot, "Beginner's guide to OpenID phishing", http://marcoslot.net/apps/openid/

[7] Security Assertion Markup Language (SAML) v2.0, http://www.oasis-open.org/specs/index.php#samlv2.0

[8] T.Wason, S. Cantor, J. Hodges, J. Kemp and P. Thomson, "Liberty ID-FF Architecture Overview", http://www.projectliberty.org/liberty/content/download/318/2366/file/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf

[9] Liberty Alliance ID-WSF 2.0 Specifications, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates

[10] S. Kellomäki, G. Lorenzo and R. Lockhart, "Liberty ID-SIS Directory Access Protocol Specification", http://www.projectliberty.org/liberty/content/download/3930/26256/file/liberty-id-sis-dap-v1.0.pdf

[11] SourceID ID-FF 1.2 Java Toolkit Overview, http://www.sourceid.org/projects/id-ff_1_2_java_toolkit.cfm

[12] A.Zeichik, "Zeichik's Take: Status Report on Java App Servers, IDEs", May 2007, http://www.sdtimes.com/content/article.aspx?ArticleID=30595

[13] K.Pepperdine, "Where is Open Source in the app server surveys?", Jun 2004, http://java.sys-con.com/read/45075.htm