

An Overview of Federated Identity Architecture

WHITEPAPER



Copyright © 2004 Oblix, Inc. All rights reserved.

This white paper is for informational purposes only. Oblix makes no warranties, expressed or implied, in this document. Mention of third-party products within this publication is for informational purposes only and constitutes neither an endorsement nor a recommendation.

The information contained in this document represents the current view of Oblix on the issues discussed as of the date of the publication. Because Oblix must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Oblix, and Oblix cannot guarantee the accuracy of any information presented after the date of publication.

Software and documentation copyright © 1996-2004 by Oblix, Inc. All rights reserved. Oblix, Oblix NetPoint, and the Oblix logo are registered trademarks of Oblix, Inc. NetPoint COREid System: User Manager, Group Manager, Organization Manager, IdentityXML, Certificate Processing Server (VeriSign®), COREid, COREid Server, and WebPass; NetPoint Access System: Access Manager, Access Server, WebGate, and AccessGate; SHAREid, SmartMarks, SmartWalls, SmartMaps, SAML quickCONNECT, FEDERATEDid Layer, Oblix IDLink, Associate Portal Services, NetPoint System Console, NetPoint Ready Realm, NetPoint Federation Services, NetPoint Mainframe Security Connector, NetPoint SAML Services, and their logos are trademarks of Oblix, Inc. All other company and product names are trade names, service marks, trademarks or registered trademarks of their respective companies.

Printed in the United States of America.

Printing Date: January 2004

Part Number: obx81a

Oblix, Inc.
18922 Forge Drive
Cupertino, CA 95014, USA
+1 408 861 6800

European Headquarters
Atrium Court
The Ring, Bracknell
Berkshire RG12 1BW, UK
+44 1344 393 054

www.oblix.com
info@oblix.com

What is Federated Identity?

During recent years, companies have moved more of their business processes to the Web. As a result, many organizations have a greater need to link their applications with their partners' or customers' applications. This often takes place through each company's web portal. For example, a phone company might operate a customer support portal, so that its customers can manage their accounts. The phone company outsources the bill pay application to a business partner, and when phone customers click the "Bill Pay" link in the phone company portal, they are redirected to the outsourcer's web site. The phone company wants its customers to be logged in to the bill pay application automatically, for a better user experience and reduced need for multiple passwords and Ids.

This sounds easy, but consider what's required to make this work. In most cases, the firms must synchronize identity information so that customers can move from one system to the other. If the sync fails, some customers can't get in to one or the other application. Alternatively, they might share a large centralized user database, but this brings its own operational and security issues.

Instead, federating identity data allows each company to operate independently, but cooperate for business purposes. The time and cost of connecting the applications drops dramatically, and the level of privacy compliance also grows, as there are fewer copies of users' identity data in existence. Portal users do not need to remember IDs and passwords for applications at partner sites, resulting in a better experience and increased security. At the same time, portal administrators are not required to maintain copies of user data at each partner site, decreasing replication cost and increasing compliance with privacy regulations.

Federation Architecture Overview

Consider a scenario where two business partners (Alpha Corp. and Beta Corp.) wish to link their applications, so that portal users can access external applications without additional logins.

Alpha has a customer portal so that users can login to manage their profiles, change service levels, order new products, etc. One of the functions available to these portal users is a technical specification database for all products sold by Alpha and its channel partners. This database is hosted and managed by Beta Corporation, and access is restricted according to service agreements with Beta's affiliates, such as Alpha.

When an Alpha customer logs in to the portal and clicks the "Tech Specs" link, the Beta Corporation database search application is served up via the Alpha portal. Alpha's customers don't need to maintain a separate user ID and password within Beta's environment, and neither company has to synchronize passwords, IDs, or profiles.

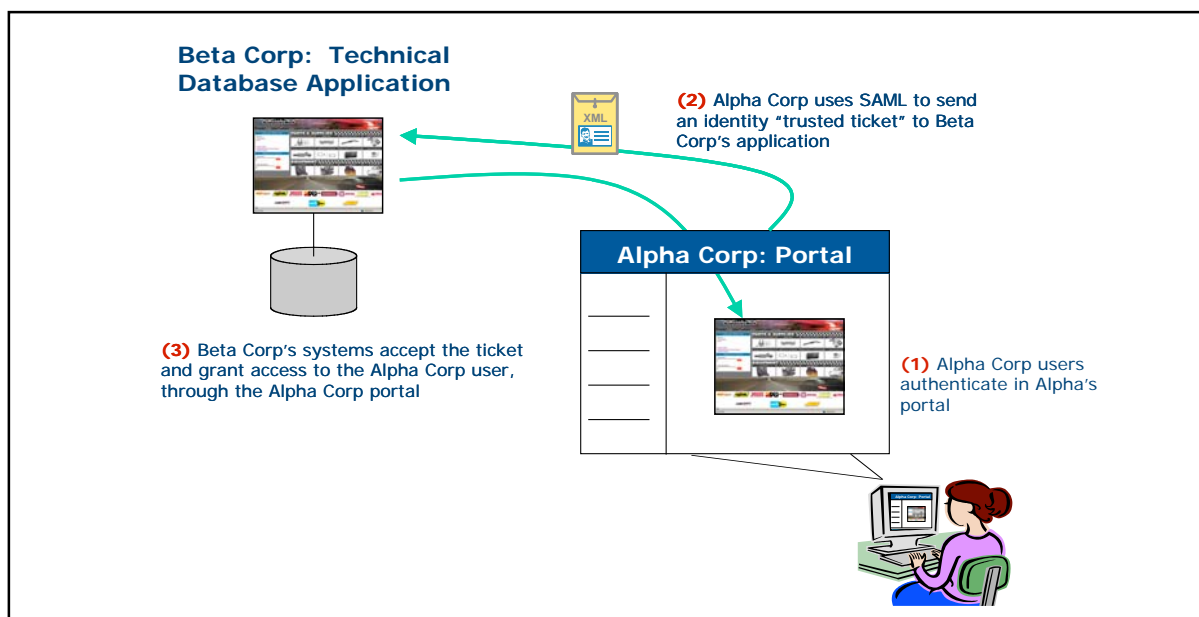
Behind the scenes, federated identity systems at both corporations transparently manage the steps required to make the above scenario possible. In our example above, Alpha and Beta use Security Assertion Markup Language (SAML), to share identity data between the two environments. Here is how this process works:

Step 1: Alpha's user logs in to the Alpha customer portal – The user provides an ID and password to authenticate against the credentials stored within Alpha's LDAP directory. After successful authentication, the portal application puts a session cookie in the user's browser.

Step 2: Alpha's user clicks the "Tech Specs" link - Alpha's federation engine uses the user's browser cookie to create a SAML assertion, digitally sign it, then redirect to Beta's federation engine.

Step 3: Beta receives the SAML assertion – Beta's own federation engine receives the SAML assertion passed from Alpha, extracts the user's identity information, and maps the Alpha user to a local Beta user (using role, email, or other information from the assertion). Beta then performs an authorization check and (if successful) redirects the user's browser to the technical specification search application.

Step 4: Alpha's user sees the Beta application in his browser – Because the Beta system successfully received and validated the Alpha user, it then placed its own cookie in the user's browser, and served up the application. The Alpha user can now navigate between applications in both domains, accessing any resources for which he has valid access rights, without additional logins.

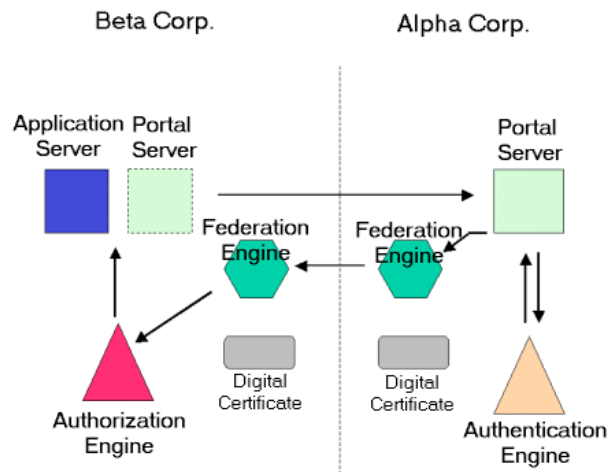


A Simple Identity Federation Example

Federated Architecture Components

Let's examine the components necessary to make this scenario work. In general, the two companies need to have these software components in place:

- **Portal Server(s).** Users typically access applications in multiple domains via a corporate portal. In this example, Alpha Corp. would have a Portal Server in place, to manage Alpha's user logins, page personalization, etc. The portal server might be homegrown logic running within an application server, or it might be a commercial product from vendors such as Microsoft, Plumtree, BEA, or others. Note that each partner might have its own portal server; Beta may serve its technical database application via a "MyBeta.com" type of portal. In that case, each company would operate its own portal server.
- **Authentication Engine.** Federation is built on trust, and in our scenario Beta Corp. trusts that Alpha Corp. has authenticated its users correctly. That is, Beta trusts that Alpha has ensured that any Alpha users being redirected to the Beta application are who they say they are. This is done via some form of authentication engine at Alpha Corp. The authentication engine may be a simple ID/password lookup in a database or LDAP directory, or it may use stronger authentication, such as ID cards, fingerprints, etc. The authentication engine might be built into Alpha's portal server, or it may be a separate product. Note that the authentication engine would not reside at Beta Corp. If users had to login again at Beta, then the companies wouldn't be providing the cross-organization single sign-on that is the goal of identity federation.



- **Digital Certificates.** Certificates are used to digitally sign the assertion at the sending end and verify the sending company's identity at the receiving end.
- **Federation Engine(s).** Once the user has authenticated at Alpha, something is required to build the SAML assertion and send it to Beta. On the Beta side, something is required to receive the SAML assertion and map it to the Beta authorization system. Each of these requirements is handled by each company's federation engine. Ideally, the federation engine can handle multiple protocols (e.g. SAML, Liberty, etc.) so that a particular company can interact with multiple partners, using each partner's preferred

protocol. The sender's federation engine prepares the information needed at the receiver, including agreed-to identity data and digital signatures to verify the sender. The federation engine may prepare this information differently depending on which standard is used. On the receiver side, the federation engine will accept the sent message, open and extract the identity data, verify the signature, and then execute logic to map this remote user to a local user, role, etc. The receiver's federation engine will work with the receiver's authorization engine to determine if this incoming user actually has rights to access the desired application.

- **Authorization System.** Beta Corp. must determine whether the incoming user is actually allowed to access the intended application, and this determination is done by Beta's authorization system. This may be custom or commercial Identity Management or Web SSO application. The authorization system contains access policies and maps users against these policies to allow or deny access to protected applications. One of the key benefits of federation is that a company that wishes to serve its application to an external firm's users need not store those users' credentials locally. The company performs authorization without needing to manage the external users' passwords, ID's, etc.
- **Application Server.** Beta Corp. likely has an application server in place to operate the technical database search application. The app server may run in conjunction with a portal (if Beta has one) or may run alone. App servers provide failover, scalability, session management, and other important web application operational functions.

Considerations for Federation

As with any technique for application deployment and integration, companies' ability to use identity federation is influenced by the state of standards, available products, and industry lessons.

State of Standards

Several identity federation standards exist today. Security Assertion Markup Language (SAML) is currently at version 1.1, and a good number of companies have deployed SAML-based federated systems. Liberty is also well-developed and covers a broader range of scenarios and functions than SAML. WS-Security is a specification driven by Microsoft and IBM that is quite broad in terms of functionality, but is further behind SAML and Liberty in terms of development and deployment. Most firms today choose SAML or Liberty for their federation deployments.

State of Products

Identity Management vendors have recently begun offering federation capabilities within their product lines. Some vendors add federation support as part of their full suite (i.e. you need to buy the whole thing) and others have created new, lightweight federation engines. While the products are relatively new, some vendors already have multiple customers in deployment and can show best practices based on production.

State of Projects

As with any technology, the early adopters show the way to those that follow. Many federation projects have successfully deployed, and are now progressing beyond simple integration to more complex federation involving dozens to hundreds of companies. These early adopters have shared their lessons with vendors and with standards bodies so that new products and standards versions will support their production needs. Companies are already saving time and money through federation, as the many case studies and magazine articles on the topic indicate.

Next Steps

Oblix solutions for centralized and federated Identity Management are in production within some of the largest portal applications in the world. To learn more about our customers' success, please contact us at:

18922 Forge Drive
Cupertino, CA 95014
+1 408 861 6800
www.oblix.com
info@oblix.com