Federation of Identities in a Web Services World

A joint whitepaper from IBM Corporation and Microsoft Corporation

July 8, 2003, Version 1.0

Copyright Notice

Abstract

This document describes the issues around federated identity management and describes a comprehensive solution based on the Web services specifications outlined in the WS-Security roadmap and other related Web services specifications.

The approach described in this whitepaper, which will be further defined in the WS-Federation specification, introduces an identity provider as a class of security token service. As such, it uses the mechanisms of WS-Trust and WS-Federation to create and broker trust within and across federations. Additionally, mechanisms are defined for single sign-in and sign-out, sharing of attributes based on authorization and privacy policies, and integrated processing of pseudonyms (aliases used at different sites/federations).

Together, the specifications identified in this paper provide a comprehensive and integrated set of protocols for secure reliable transacted messages in and across federations by composing with other security and Web service specifications.

Executive Summary

Over the past several years Web applications have evolved from simple content delivery applications into sophisticated business productivity tools and a mechanism for application integration within and across enterprises. The growth of the Web and Web services has demonstrated the need for open and interoperable solutions to many technical problems.

In this document, we focus on a specific set of security related issues. These include:

How do we determine the right Secure Identity and Location Web services: Organizations need a standard way for service requestors (customers and partners) to *securely* find the right Web services of a given business and for business service providers to securely identify and expose the right Web Service to only authorized requestors.

How do we determine the set of credentials to enable secure invocation of Web services: A standardized means for business service requestors to securely invoke Web Services with the right set of authentication, authorization and entitlements.

How do we securely federate Web services: A standard way for allowing businesses to directly provide services for customers registered at other (partner) businesses or institutions. Within a federation of services, a business can get trusted

information about a user from the user's home organization (or information-providing service). The business doesn't need to register and maintain that user's identity, and the user is spared from having to get and remember a new login in order to interact with the business.

How do we enable Cross-Enterprise and Cross-Domain Trust: A standard way for establishing and reflecting trust between organizations. This is a key issue for Federated Services.

How do we enable Federated Identity and Attribute Mapping: Well-understood mechanisms and procedures for mapping trusted information about a foreign user (e.g., users from business partners) into authentication and authorization information usable by an enterprise's existing services.

How do we enable Secure, Reliable Transactions: A standard way to exchange messages in a secured, reliable, and transacted context. Previous specifications (e.g. WS-ReliableMessaging and WS-Transaction) provide support for reliable message exchange and business transactions. In this document we discuss federated security support.

IBM, Microsoft and our partners intend to work with customers, partners and standards bodies to evolve the specifications described here, and to ensure these specification compose well with other elements of the Web services architecture. To ensure interoperability and consistent implementation of the various proposed specifications described in this paper, IBM, Microsoft, and our partners will work closely with standards organizations, the developer community, and with industry organizations such as WS-I.org to develop interoperability profiles and tests that will provide guidance to tool vendors.

Because terminology varies between technologies, this document defines several terms that may be applied consistently across the different security formats and mechanisms. Consequently, the terminology used here may be different from other specifications and is defined so that the reader can map the terms to their preferred vocabulary. Refer to the Terminology section for a summary of these terms and their definition and usage.

1 Introduction and Motivation

Federated identity management represents a significant challenge for both individuals and businesses. This chapter explores the problem, then identifies the parties affected by it and concludes with key goals for viable solutions.

What is the Federated Identity Management problem?

A company's value network spans many organizations, systems, applications and business processes. Several different constituents make up this value network including the customers, employees, partners, suppliers, and distributors. There is no single entity or company that can purport to centrally manage or control identity information about its constituents in this end-to-end value network. Even within a single company there may exist multiple authoritative sources of identity data that need to be managed independently and autonomously within the business units.

This approach to centralization as the underlying tenet to cross-company collaboration introduces significant friction in e-business collaboration, integration and automation, resulting in high costs of identity management and reduced

efficiency. With centralization, the cost of managing the lifecycle of user identities is very high. Most businesses have to manage employee, business partner, and customer identities. In addition, the relationships between the business and these individuals change fairly frequently, and each change requires an administrative action.

In some cases, businesses would like to outsource some security functions to parties who manage identity (as they do today to credit-card companies in transactional contexts) but they cannot do this for two reasons: first, there are no third-party identity providers serving markets other than consumer financial transactions, and second, there are no business and liability models which make it safe to rely on the services of a third-party identity provider for services other than consumer financial transactions.

Other businesses want to leverage the identities they maintain to enable additional business interactions. However, establishing the trust mechanisms to allow entities to be federated across business boundaries is difficult. Also, businesses that manage identity increasingly are at risk of reputation damage or regulatory liability if their identity management actions release or use information in ways which conflict with individual privacy rights. This greatly increases the risk of managing identity.

From an individual's perspective, multiple identities exist, both personal and professional. The individual also has an identity management problem caused by the inability to re-use identities.

Federated Identity Management delivers cross-company business flexibility while enabling companies to off-load and simplify identity management costs. This enables companies to pursue business integration goals that best align with their business model, IT policy, and security and governance goals and requirements.

Who has the Federated Identity Management problem?

The main integration barrier lies in cross-company business integration due to the lack of secure communication models. The problem affects a wide range of companies including:

- Medium and large organizations that use identity information to provide services to consumers (for example, providers of Web-based travel services).
- Medium and large organizations that do business with one another and need to exchange information about individuals' identities (for example, an airline and a rental car agency, or a hospital and a health insurance provider)
- Organizations that need to integrate business applications across the enterprise and the value chain of customers and suppliers (e.g. a supply chain) and need to authorize their employees to conduct transactions on behalf of the organization.
- Organizations that outsource services, such as HR and benefits, to third parties but apply their own brands to these outsourced services (for example, an organization which provides its employees with multiple retirement plan or medical plan options via its own internal HR portal) and which must therefore share employee identity information with the service providers.
- Organizations (intermediaries, brokers, aggregators) whose business model is driven by "owning the customer experience" for reasons of disintermediation.

- Organizations that provide integrated branded full-service identity-driven business portals (financial services, insurance services, subscription, etc.) by aggregating services across multiple third-party providers.

As noted previously, individuals engaged in Web-based activities also suffer this problem in that they typically have a large number of independent identities that they must create and manage.

Goals

The primary goals of Federated Identity Services are as follows:

- Reduce the cost of identity management by reducing duplication of effort; each individual's identity is almost always already managed by a trusted organization (such as the individual's bank, employer, or physician).
- Leverage the work these existing identity managers have already done by giving other parties access (as required and with appropriate privacy protection) to the relevant identity information.
- Preserve the autonomy of all parties – an identity manager's choice of authentication technology should not impose that technology on parties who rely on its identity information. An identity manager's choice of operating system, or networking protocol, or database, should not impose the same choice on its partners.
- Respect business' pre-existing trust structures and contracts. Signing up to receive identity information from an identity provider must not require an organization to establish a trust relationship with any party other than the identity provider, and must not require adoption of any specific user authentication technology.
- Protect individuals' privacy by respecting and strongly enforcing user preferences governing the use of individually identifiable information, observing governmental and regional privacy rules, seeking the user's consent for new uses, and implementing strong recordkeeping and accountability mechanisms to ensure that privacy practices are followed.
- Build on open standards to enable secure reliable transactions for businesses and individuals.

2. Federated Single Sign On and Identity Management

The appeal of *federations* is that they are intended to allow a user to seamlessly traverse different sites within a given federation. This document focuses specifically on the issue of federated identity management and not the greater issue of general federation (beyond just security). Because of the trust relationships established between the federation participants, one participant is able to authenticate a user, and then act as an *issuing party* for that user. Other federation participants become *relying parties*. That is, they rely on information that is provided about the user by the issuing party, without the direct involvement of the user. In some cases the user may be anonymous to the relying party, for example, due to the different authentication mechanisms and use of third party authentication mechanisms.

The flexibility and appeal of the Web Services model is its building block foundation whereby companies can easily build new services to deliver innovative business models or link their value-chain network more efficiently through tight relationships

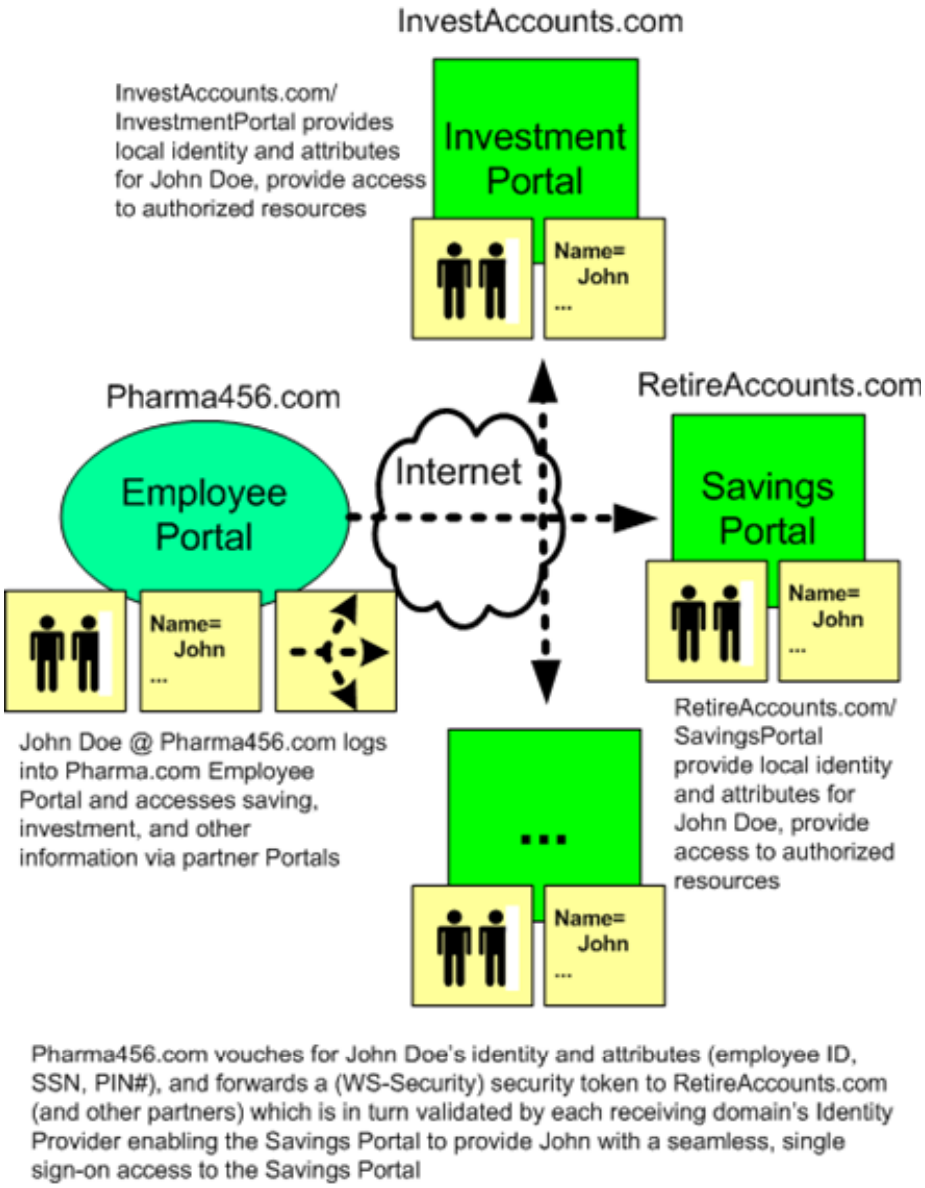
with partners, suppliers, customers and employees. Such a model can only be successful if it allows customers, partners and end-users to navigate easily between Web Sites supporting these services without having to constantly authenticate or identify themselves to the various sites, or to redundantly maintain personal information at each member within the business federation.

Unfortunately current schemes for authenticating users and getting user attribute information (e.g. credit card information) typically force the user to register with each business of interest, constantly requiring the user to identify and authenticate themselves (typically with a user name and password) and forcing the business to administer a large and rapidly changing base of identities that are not under the control of the company. Such a model is a huge impediment to the adoption of Web Services - and is a headache for both users and businesses.

Another downside of the prevalent model is that a given business may be unwilling to "give" parts of its customer information to a business partner, wishing instead to maintain and control the customer relationship.

Federations provide a simple flexible mechanism to identify and validate users from partner organizations and provide them with seamless access to Web sites within that trusted federation without requiring re-authentication with a password. In addition, Federation standards also deal with the matter of providing trusted attributes (e.g. X509 certificates, X509v3 attribute certificates, Kerberos tokens, SAML assertions) about users (e.g. including roles and group information) allowing for privacy and business-specific rules.

The concept and notion of Federated Single Sign On and Identities can be illustrated using a simple example:



User John Doe works for a pharmaceutical company called Pharma456.com; John has an account with Pharma456.com and is required to authenticate to Pharma456.com to access its resources. As an employee, John has a certain benefits with the company such as accounts and services at partner companies. In this example, the company's savings services are managed by a financial services firm called RetireAccounts.com (a service provider) and investment services are managed by a firm called InvestAccounts.com (a service provider). In order to access the resources at a partner, for example the savings portal hosted by RetireAccounts.com, a user must authenticate to RetireAccounts.com. Authentication to RetireAccounts.com requires that a user supply her Social Security Number (SSN), an Employee Identifier (a unique identifier issued by the employer – in this case Pharma456.com) and a personal PIN number (specific to RetireAccounts.com).

Without federation, John Doe has to explicitly authenticate to RetireAccounts.com site to access his savings account even though he has already authenticated to

Pharma456.com and has accessed the RetireAcounts.com Web services via the Pharma456.com employee portal.

However with federated single sign on John Doe can logon to his employee Portal, click on a portal link to access his savings account information from the partner site (RetireAccounts.com) and not have to re-authenticate or provide additional information at the partner site.

Federation also ensures that distinct identities for the same user across companies can be securely linked between two companies using federated management techniques. Federation participants can exchange identity information such that the relying company can independently validate the user's identity within their domain.

Federated single sign on between an issuing domain (Pharma456.com) and a relying domain (the federated service provider RetireAccounts.com) facilitates the secure and trusted transfer of user identifiers and other attribute-related information (such as authorization roles and group memberships, and user entitlements such as EmployeeID, SSN and PIN #). Federated Identity Management defines the process by which the relying party (RetireAccounts.com) is able to determine a locally valid identifier for the user based on the (trusted) information received from the issuing party. The details of the transfer may involve multiple messages between the business and the user's origin enterprise (and messages to ancillary services) but these are handled transparently to the user.

To enable federation we introduce identity providers, attribute services, and pseudonym services which are exemplified in the figure above.

Identity providers such as those at Pharma456.com, InvestAccounts.com, and RetireAccounts.com, provide identities which are used for local mapping/indexing (e.g. account information). By using trust and federation mechanisms, along with trust policies, these identities allow federations to share and map identities automatically.

Attribute services provide a way to federate access to authorized attributes for federated identities. In the example above, as John Doe visits each partner's portal, the portal service may access John's attributes (those to which it is authorized) in order to obtain required information and to personalize the experience. To ensure privacy, John Doe has full control over which attributes are authorized to which services. These attribute accesses can be monitored and logged to ensure compliance with stated privacy policies established with Pharma456.com employees. We don't mandate a specific type of attribute service, but instead allow different services, such as UDDI to be used.

Trust mechanisms provide a way for relying parties to associate a level of trust with an authority or identity and use that for local mappings. However, there are situations where privacy concerns wish this mapping to be opaque to the target service. That is, the target consistently knows it is "John Doe" based on John's local persona, but don't understand or know the global identity. The pseudonym services provide a mapping mechanism which can be used to facilitate this mapping of trusted identities across federations to protect privacy and identity.

Federated Identity Management (including Federated Single Sign On) is a much broader concept than Web Single Sign On solutions. While this example highlights a use case for B2C scenario, where Web SSO is an important feature, Federated Single Sign On is a broader concept that enables businesses to build a complete framework for secure B2B and B2C e-business.

3. Federated Identity Model

The *federated identity model* builds on, and integrates, the Web services infrastructure and security specifications to form a consistent and extensible security model. The federation model extends the WS-Trust model to describe how identity providers act as security token services and how attributes and pseudonyms can be integrated into the token issuance mechanism to provide federated identity mapping mechanisms.

In summary, principals sign-in and sign-out of Identity Providers (or security token services). This can be done via explicit messages or implicitly as principals request tokens. A principal requests tokens for resources/services and the issued tokens may either represent the principal's primary identity or some pseudonym appropriate for the scope. The Identity Provider (or STS) issues messages to interested (and authorized) recipients. Principals are registered with the attribute/pseudonym services and attributes and pseudonyms are added and used. Services can query attribute/pseudonym services using the provided identities (potentially anonymous which means that the party requesting the information has an opaque token, and is not aware of the real identity) to obtain authorized information about the identity.

Web Services Security Specifications

The model and approach described in this paper leverage the specifications described in the white paper, *Security in a Web Services World: A Proposed Architecture and Roadmap*. Each of the key specifications is summarized below:

- WS-Security describes how to attach signature and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.
- WS-Policy represents a set of specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries and endpoints (e.g. required security tokens, supported encryption algorithms, privacy rules) and how to associate policies with services and endpoints.
- WS-Trust describes a framework for trust models that enables Web services to securely interoperate by requesting, issuing, and exchanging security tokens.
- WS-Privacy will describe a model for how Web services and requestors state privacy preferences and organizational privacy practice statements.
- WS-SecureConversation describes how to manage and authenticate message exchanges between parties, including security context exchanges and establishing and deriving session keys.
- WS-Federation describes how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, sharing of attributes, and management of pseudonyms.
- WS-Authorization will describe how to manage authorization data and authorization policies.

Additionally, several other key Web services specifications complete the foundation layer of specifications:

- WS-Addressing describes how to specify identification and addressing information for messages.

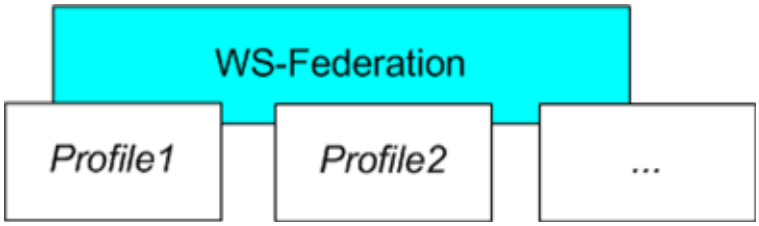
- WS-MetadataExchange describes how to exchange metadata such as WS-Policy information and WSDL between services and endpoints.
- WS-ReliableMessaging describes how to ensure reliable delivery of messages in the presence of unreliable networks.
- WS-Transactions and WS-Coordination describe how to enable transacted operations as part of Web service message exchanges.

The combination of the specifications above and interoperability profiles will enable customers to easily build interoperable secure reliable transacted Web services that integrate within and across federations by composing federation and security specifications with other Web services specifications.

Profiles

This paper describes a general model that can be used in different environments. Specifically, this model can be used in environments consisting of passive requestors such as Web browsers or active requestors such as SOAP requestors.

The federation specifications provide a general model and framework; profiles describe how the model is applied in these different environments. Additional profiles may be specified for integrating the model into other environments.



Each profile specification defines how the mechanisms in WS-Federation are applied, if at all, to a given environment such as passive or active requestors.

Consequently, the mechanisms described in this paper (identity providers, sign-in/out, security tokens, attributes, pseudonyms, …) apply to both passive requestors (such as Web browsers) and active requestors (such as Web services acting both as clients and services), as well as other profiles which may be defined in the future.

Identity Providers

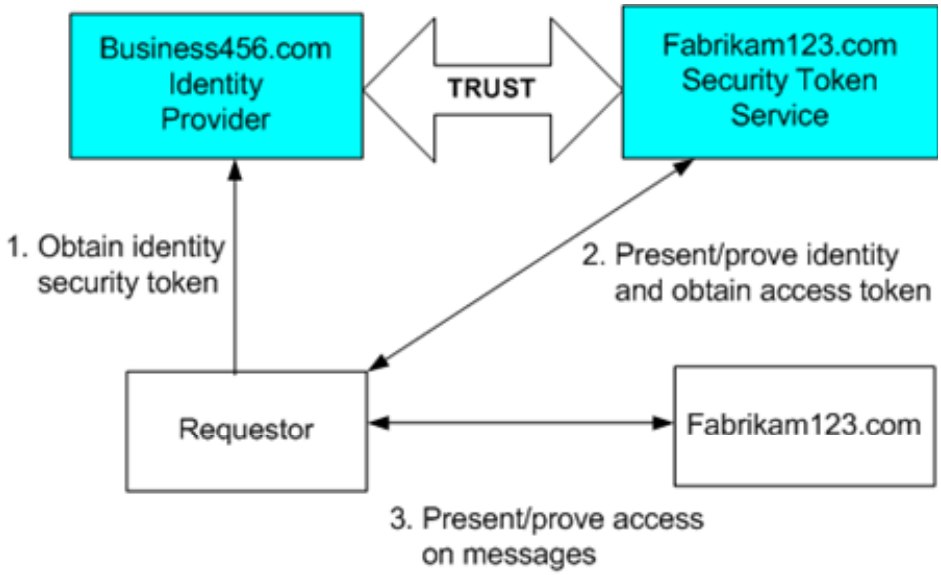
Federation starts with a notion of identity. That is, the requestor or the requestor's delegate (an identity provider who is the authoritative owner for that identity data) asserts an identity and the Identity Provider verifies this assertion. Federation then becomes a function of trust (direct, brokered, and delegated) between identity providers and those relying on the provider's determination of identity. Sometimes the relying party needs the ability to correlate the identities from multiple providers - for example, correlation of an identity on a check, on a credit card, and on a driver's license.

A security token service (STS) is a generic service that issues/exchanges security tokens using a common model and set of messages. As such, any Web service can, itself, be an STS simply by supporting the WS-Trust specification. Consequently, there are different types of security token services which provide different supplemental services. An *Identity Provider (IP)* is a special type of security token service that, at a minimum, performs peer entity authentication and can make identity or affiliation claims in issued security tokens. Note that in many cases an IP

and STS are interchangeable and many references within this document identify both.

The federation model builds on the framework defined in WS-Trust by leveraging the token issuance and exchange mechanism to include issuing and federating identities.

The following example illustrates a possible combination of an IP and STS to access a service. In this example, (1) a requestor obtains an identity security token from their IP (Business456.com) and then presents/provides the assertion (security token) to the STS for Fabrikam123. If Fabrikam123.com trusts Business456.com, and authorization is approved, the STS returns an access token to the requestor. The requestor (3) then uses the access token on requests to Fabrikam123.com.



In this example the response in step 1 is signed by Business456.com (the trusted IP) and the security token returned in the response is relative to Fabrikam123.com (e.g. they issued it).

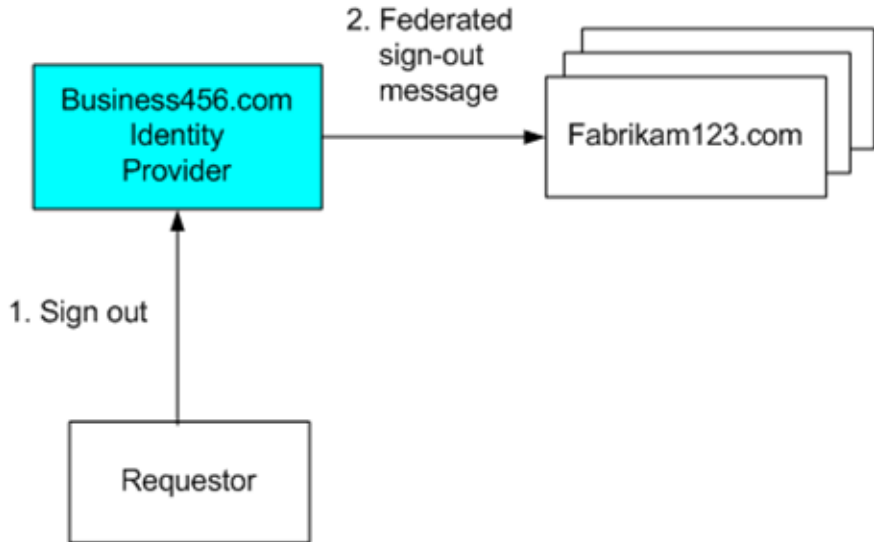
An alternative model, which is discussed below, allows the service Fabrikam123.com to register a security token with a pseudonym service and fetch this pseudonym when needed. This allows the service to manage the mapping and still provide a level of privacy for the requestor.

Single Sign On and Sign-Out

The model described in this paper and in the WS-Federation specifications allows for different notions of *user sessions* such as service-managed and requestor-managed. One example of a service-managed session would be the creation and management of cookies within a Web browser session. An example of a requestor-managed session is a Web service requestor which obtains a token and uses it for a period of time, and then discards the token prior to its expiration. The notions of *sign-out* are introduced to allow different profiles to specify how these transitions apply to each profile.

The purpose of *Single Sign On* is to establish security tokens required to access a resource within the Web of federated domain/realms. Similarly, *federated sign-out* is used to clean up any cached state and security tokens that may exist within the federation. To enable this, mechanisms are required to provide Identity Provider-

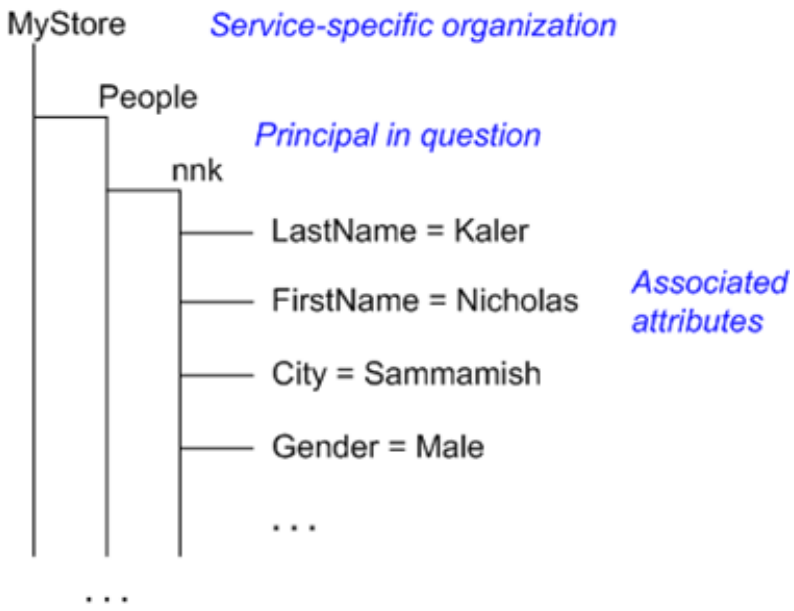
issued federated sign-in and sign-out messages to authorized parties of sign-in and sign-out actions (thereby allowing them to perform any necessary setup/cleanup actions).



Attributes and Pseudonyms

As previously discussed, there is a desire to be able to obtain information about an identity (or any federated resource) - such as for providing a "hello" greeting or obtaining the requestor's zip code to personalize an experience - and this can be provided by an attribute service. This specification allows for different types of attribute services, such as UDDI, to be used.

It is critical that such information be governed by authorization rules and privacy semantics. Similarly, it is expected that different attributes will be shared differently and have different degrees of privacy and protection (e.g. first name vs. last name). Consequently, each attribute expression should be capable of expressing its own access control policy and the policy should take into account the associated scope(s) and principals that can speak for the scope(s). For example, an end user (person) may wish to set up the following: "my services in my intranet may have access to my last name whereas other services cannot without express permission from me".



An attribute service may leverage existing repositories and may provide some level of organization or context. Within the organizational namespace, individual principals are registered and a set of attribute properties (essentially name/value pairs where the name is a string property name and the value is an XML element) represented in XML are associated with the principal.

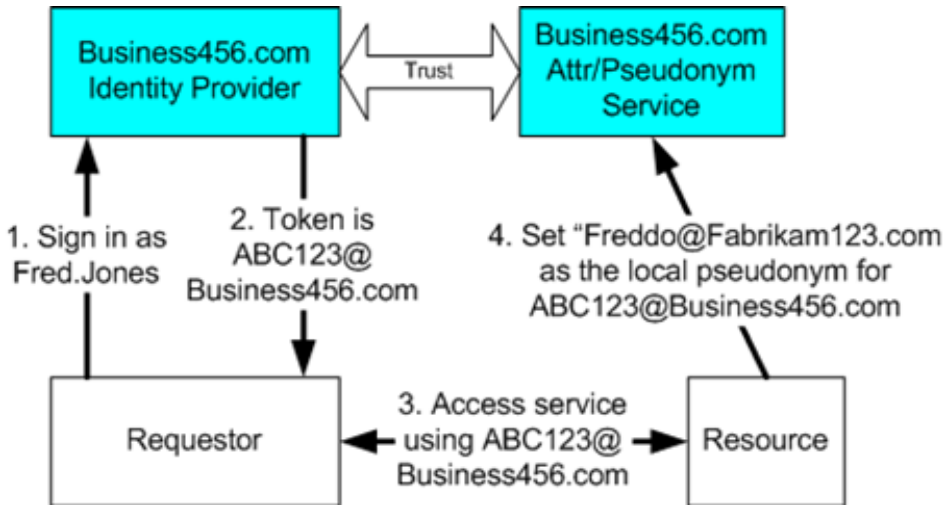
It is important to note that each attribute may have its own security authorization rules and privacy policy, allowing principals to control to whom and how information is disclosed.

Different attribute services have different capabilities which are expressed in their policy document.

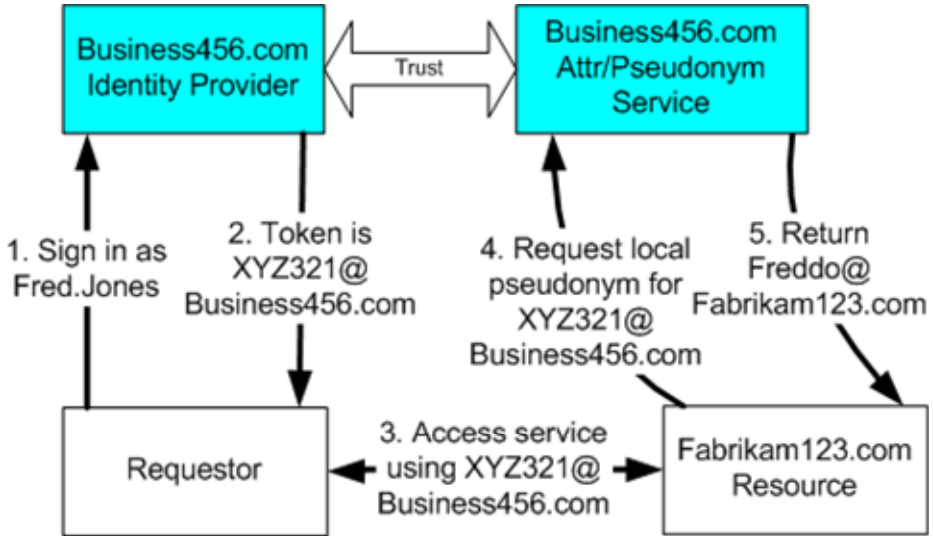
In addition to attribute services, there may also be *pseudonym services*. A pseudonym service allows a principal to have different *aliases* at different resources/services or in different domains/realms. Some identity providers use fixed identities in their security tokens. In some scenarios it is desired to ensure anonymity of the tokens; pseudonyms provide a mechanism for enabling this anonymity. There is often a trade-off of manageability that must be determined by the principal (i.e., the more identities, the greater potential for management issues).

It should be noted that in some cases the attribute and pseudonym services will be combined and in some cases they will be separate services.

For example, a requestor authenticates to Business456.com with his primary identity "Fred.Jones". However, at Fabrikam123.com, he is known as "Freddo". To preserve anonymity, Business456.com can issue a different identity whenever Fred.Jones signs in, thus appearing "anonymous" as illustrated in step 3 in the figure below. Fabrikam123.com can set "Freddo" as the local name for this requestor at Fabrikam123.com (step 4) and have the pseudonym service, which understands the anonymous name, provide the mapping to "Freddo".

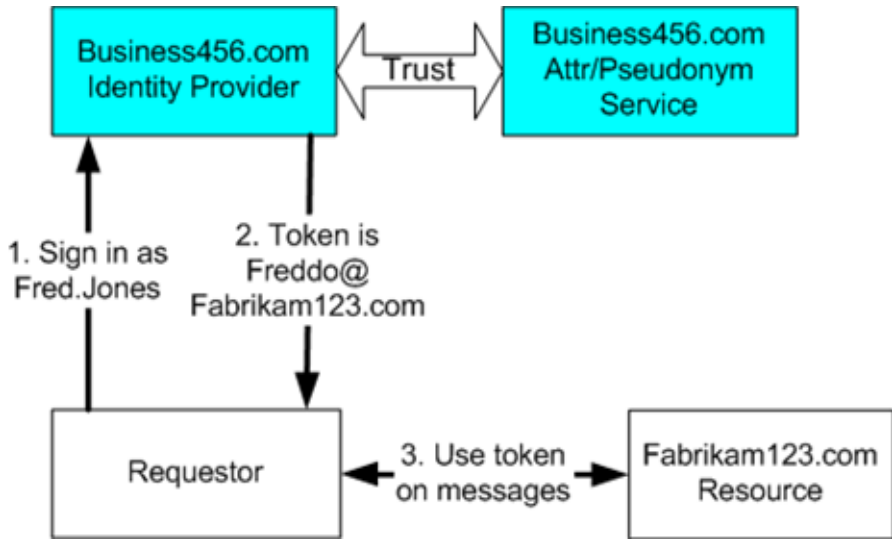


The next time the requestor signs in to Business456.com IP (step 1 below), he might be given a new identifier like "XYZ321@Business456.com" (step 2 below). Since Business456.com IP has the mapping discussed above, the Web service at Fabrikam123.com can now request a pseudonym for XYZ321@Business456.com at Fabrikam123.com (step 4 below) and get back what they previously set – in this case it is "Freddo@Fabrikam123.com" (step 5 below).



The pseudonym service is able to do this because it has the ability to back-map "XYZ321@Business456.com" into a known identity at Business456.com which has associated with it pseudonyms for different domains/realms. This back-mapping occurs based on a trust relationship between the IP and the pseudonym service. Similarly there is a trust relationship between the resource and the pseudonym service (possibly bootstrapped by the IP) that enables the resource to be authorized to get and set pseudonyms.

Alternatively, the Identity Provider (or STS) can operate hand-in-hand with the pseudonym service. That is, the requestor asks its Identity Provider (or STS) for a token to a specified trust domain/realm or resource/service. The STS looks for pseudonyms and issues a token which can be used at the specified resource/service, as illustrated below:



As illustrated, there are a number of different approaches supported in this Federated Identity model in which pseudonyms can be used to help maintain privacy. Each has different characteristics of manageability and privacy, allowing each provider and principal to choose the appropriate solution to meet its specific requirements.

4 Summary

In this document, we propose an integrated Web services federation model that enables parties to issue and rely on information from other members of federations and to broker trust and attributes across federations in a secure way that maintains individual and business privacy.

This model integrates with the Web services security and related specifications to enable secure reliable transactions between requestors and services within and across federations.

IBM and Microsoft believe that this is an important step in defining a comprehensive Web services security strategy. It reflects the challenges and solutions we have identified thus far. As we continue to work together with customers, partners and standards organizations to secure Web services, we expect that there will be additional ideas and specifications needed to make the strategy complete.

Terminology

Because terminology varies between technologies, this document defines several terms that may be applied consistently across the different security formats and mechanisms. Consequently, the terminology used here may be different from other specifications and is defined so that the reader can map the terms to their preferred vocabulary.

Passive Browser – A *passive browser* is an HTTP browser capable of broadly supported HTTP (e.g. HTTP/1.1).

Active Requestor – An *active requestor* is an application (possibly a Web browser) that is capable of issuing Web services messages such as those described in WS-Security and WS-Trust.

Profile – A *profile* is a document that describes how this model is applied to a specific class of requestor (e.g., passive, or active).

Claim – A *claim* is a declaration made by an entity (e.g. name, identity, key, group, privilege, capability, attribute, etc).

Security Token – A *security token* represents a collection of claims.

Signed Security Token – – A *signed security token* is a security token that is asserted and cryptographically signed by a specific authority (e.g. an X.509 certificate or a Kerberos ticket)

Proof-of-Possession – *Proof-of-possession* is authentication data that is provided with a message to prove that the message was sent and or created by a claimed identity.

Proof-of-Possession Token – A *proof-of-possession token* is a security token that contains data that a sending party can use to demonstrate proof-of-possession. Typically although not exclusively, the proof-of-possession information is encrypted with a key known only to the sender and recipient parties.

Digest – A *digest* is a cryptographic checksum of an octet stream.

Signature - A *signature* is a value computed with a cryptographic algorithm and bound to data in such a way that intended recipients of the data can use the signature to verify that the data has not been altered since it was signed by the signer.

Security Token Service (STS) - A *security token service* is a Web service that issues security tokens (see WS-Security and WS-Trust). That is, it makes assertions based on evidence that it trusts, to whoever trusts it. To communicate trust, a service requires proof, such as a security token or set of security tokens, and issues a security token with its own trust statement (note that for some security token formats this can just be a re-issuance or co-signature). This forms the basis of trust brokering.

Attribute Service - An *attribute service* is a Web service that maintains information (attributes) about principals within a trust realm or federation. The term principal, in this context, can be applied to any system entity, not just a person.

Pseudonym Service - A *pseudonym service* is a Web service that maintains alternate identity information about principals within a trust realm or federation. The term principal, in this context, can be applied to any system entity, not just a person.

Trust - *Trust is* the characteristic that one entity is willing to rely upon a second entity to execute a set of actions and/or to make set of assertions about a set of subjects and/or scopes.

Trust Domain - A *Trust Domain* is an administered security space in which the source and target of a request can determine and agree whether particular sets of credentials from a source satisfy the relevant security policies of the target. The target may defer the trust decision to a third party thus including the trusted third party in the Trust Domain.

Validation Service - A *validation service* is a Web service that uses the WS-Trust mechanisms to validate provided tokens and assess their level of trust (e.g. claims trusted).

Direct Trust – *Direct trust* is when a relying party accepts as true all (or some subset of) the claims in the token sent by the requestor.

Direct Brokered Trust – *Direct Brokered Trust* is when one party trusts a second party who, in turn, trusts or vouches for, the claims of a third party.

Indirect Brokered Trust – *Indirect Brokered Trust* is a variation on direct brokered trust where the second party can not immediately validate the claims of the third party to the first party and negotiates with the third party, or additional parties, to validate the claims and assess the trust of the third party.

Message Authentication – *Message authentication* is the process of verifying that the message received is the same as the one sent.

Sender Authentication – *Sender authentication* is corroborated authentication evidence possibly across Web service actors/roles indicating the sender of a Web service message (and its associated data). Note that it is possible that a message may have multiple senders if authenticated intermediaries exist. Also note that it is application-dependent (and out of scope) as to how it is determined who first created the messages as the message originator might be independent of, or hidden behind an authenticated sender.

Realm or Domain – A *realm* or *domain* represents a single unit of security administration or trust.

Federation – A *federation* is a collection of realms/domains that have established trust. The level of trust may vary, but typically includes authentication and may include authorization.

Identity Provider – *Identity Provider* is an entity that acts as a peer entity authentication service to end users and data origin authentication service to service providers (this is typically an extension of a security token service).

Single Sign On (SSO) – *Single Sign On* is an optimization of the authentication sequence to remove the burden of repeating actions placed on the end user. To facilitate SSO, an element called an Identity Provider can act as a proxy on a user's behalf to provide evidence of authentication events to 3rd parties requesting information about the user. These Identity Providers are trusted 3rd parties and need to be trusted both by the user (to maintain the user's identity information as the loss of this information can result in the compromise of the users identity) and the Web services which may grant access to valuable resources and information based upon the integrity of the identity information provided by the IP.

Identity Mapping – *Identity Mapping* is a method of creating relationships between identity properties. Some Identity Providers may make use of id mapping.

Sign-Out – A *sign-out* is the process by which security tokens are destroyed for a realm/domain or federation.

Association – *Association* is the process by which principals become associated or affiliated with a trust realm or federation.

Contributors

This document was jointly authored by IBM and Microsoft.

Key contributes include (alphabetically): Giovanni Della-Libera, Microsoft; Brendan Dixon, Microsoft; Mike Dusche, Microsoft; Don Ferguson, IBM; Praerit Garg, Microsoft; Tim Hahn, IBM; Heather Hinton, IBM; Maryann Hondo, IBM; Chris Kaler, Microsoft; Frank Leymann, IBM; Brad Lovering, Microsoft; Hiroshi Maruyama, IBM; Anthony Nadalin, IBM; Nataraj Nagaratnam, IBM; Venkat Raghavan, IBM; John Shewchuk, Microsoft

References

[Kerberos]
J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt .

[SOAP]
W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part0/
Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part1/
Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part2/.

[XML-Encrypt]
W3C Working Draft, "XML Encryption Syntax and Processing," 04 March 2002.

[XML Signature]
W3C Proposed Recommendation, "XML Signature Syntax and Processing," 20 August 2001.

[X509]
S. Santesson, et al,"Internet X.509 Public Key Infrastructure Qualified Certificates Profile,"
http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200003-I

[WS-Security Roadmap]
"Security In A Web Services World: A Proposed Architecture and Roadmap", IBM/Microsoft, April 2002

[WS-Security]
"Web Services Security Language", IBM, Microsoft, VeriSign, April 2002.
"WS-Security Addendum", IBM, Microsoft, VeriSign, August 2002.
"WS-Security XML Tokens", IBM, Microsoft, VeriSign, August 2002

[WS-Policy]
"Web Services Policy Framework", BEA, IBM, Microsoft, SAP, December 2002

[WS-PolicyAttachment]
"Web Services Policy Attachment Language", BEA, IBM, and Microsoft, SAP, December 2002

[WS-PolicyAssertions]
"Web Services Policy Assertions Language", BEA, IBM, Microsoft, SAP, December 2002

[WS-Trust]
"Web Services Trust Language", IBM, Microsoft, RSA, VeriSign, December 2002

[WS-SecureConversation]
"Web Services Secure Conversation Language", IBM, Microsoft, RSA, VeriSign, December 2002

[WS-SecurityPolicy]
"Web Services Security Policy Language", IBM, Microsoft, RSA, Verisign December 2002

[WS-ReliableMessaging]
"Web Services Reliable Messaging Protocol", BEA, IBM, Microsoft, Tibco, February 2003

[WS-Federation]
"Web Services Federation Language", BEA, IBM, Microsoft, RSA Security, VeriSign, July 2003

"Web Services Federation Language: Passive Requestor Profile", BEA, IBM, RSA Security, Microsoft, VeriSign, July 2003

"Web Services Federation Language: Active Requestor Profile", BEA, IBM, Microsoft, RSA SecurityVeriSign, July 2003

Copyright Notice

INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO THE DOCUMENT.

IN NO EVENT WILL IBM OR MICROSOFT BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.