



The Security Division of EMC

RSA Solution Brief

RSA® Federated Identity Manager

A Technical Overview



Federated identity management extends the management of digital identities for authorization and access beyond domain and corporate boundaries to externally hosted or managed applications and resources. This paper summarizes the concepts underlying federated identity, explores the standards and specifications that are emerging to support identity federation, including the Security Assertion Markup Language (SAML) and the work of the Liberty Alliance and WS-* groups, and examines the technology that underlies the RSA® Federated Identity Manager solution.

Introduction

Secure access to sensitive and confidential resources on the web has emerged as a major challenge for businesses. Organizations responded to the challenge by developing identity and access management strategies that define local security domains. These domains provide a specific set of users with access to a defined set of resources and give them digital identities with attributes that can be used by the applications and services they access.

But even as businesses have developed these local security domains, they have recognized the urgent need to expand access to remote offices, autonomous business units and business partners.

Technologies and standards have emerged to enable this sharing of access and identities. Under the collective label of “identity federation” they give organizations the ability to provide business partners access to their resources - and to make use of the resources of their partners - in relationships built on trust, mutual agreement and risk sharing.

Identity federation is evolving rapidly and the demand for solutions in the marketplace has already led to the rapid evolution of products and implementations.

RSA has been both a leader in this development and a forceful advocate for standards. It has contributed patent rights and industry leadership to the creation of the Security Assertion Markup Language (SAML), a cornerstone standard for identity federation.

RSA is a founding member of the Liberty Alliance and has helped author and review the security-related WS-* specifications.

The company’s leadership is also evident in RSA’s RSA Federated Identity Manager. The product is a result of the company’s deep understanding of the business, technology and legal issues associated with federated identity systems. Identity Manager works with identity and access management solutions and domain-based authentication authorities to enable communication across the boundaries of domains, protocols and vendor implementations:

- RSA Federated Identity Manager currently supports or will support in the near future the spectrum of federated identity protocols-SAML 1.0, SAML 1.1, SAML 2.0, and, when they are promulgated, the WS-* standards for web services security.
- RSA Federated Identity Manager is modular and extensible to insure that it will be compatible with whatever protocols and identity infrastructure products customers might use in their own domains or encounter in the domains of business partners.
- RSA Federated Identity Manager is designed to be as easy as possible to deploy and manage in complex infrastructures. It is administered from a browser-based administration GUI like that used by other RSA products.



Federated Identity Concepts

Federated identity is most easily understood (and most widely implemented) as technology to provide the benefits of single sign-on (SSO) across the boundaries of corporate security domains. Just as enterprise or web SSO allows users to access multiple web applications without having to reauthenticate within a domain, federated identity provides SSO access to applications and network resources across heterogeneous domains, each with its own authentication authority.

Identity federation works by supporting the definition and management of trust relationships between the two domains. When a user who is authenticated in one domain accesses an application in a second domain, the two domains can securely share trustworthy security information that allows the second domain to authenticate the user - without a need to re-challenge - and to grant access.

Federated identity technology goes beyond SSO to perform other enabling tasks as well:

- **Attributes.** It can communicate attributes associated with the identity of the user-information such as such as account numbers, organizational roles, account balances, location - that can be used by applications to make authorization decisions and to personalize the end user experience.
- **Identity mapping.** Because disparate security domains use and store data differently, federated identity technology must be able to map identities across domains-to translate the identity and attributes of a user in one domain into the conventions used by another. For example, an application being accessed must be able to recognize that “Johnd” is really “Jdoe” and his attribute of “Sales” should grant him access to all the application functions associated with “salesteam.” When privacy of the individual is a concern, identities can also be mapped anonymously or by group function.

- **Management.** Any federated identity solution must provide management capabilities to perform the tasks required to create, provision, manage and monitor it. These include the creation of the sophisticated business policies forged among business partners concerning acceptance and use of federated identities; configuring the system tools that perform attribute transfers and identity mapping; and logging and reporting on administrator and user activity.

A Standards-based Solution

Because federation involves different systems from different vendors operated by different enterprises, a federated identity solution must by its very nature be standards-based. The key underlying standard for federated identity is the Security Assertion Markup Language (SAML). SAML is the most mature and widely deployed identity federation protocol today and offers the highest potential for interoperability with federation partners. The latest version, SAML 2.0, marks the convergence of the SAML, Liberty ID-FF, and Shibboleth specifications into a single unified standard.

RSA is also working with Microsoft, IBM and others on a set of security specifications for federation based on web services, grouped under the name of WS-*. Included are WS-Security (which provides basic security services for SOAP messages), WS-Trust (which defines the means for establishing trust relationships among web services) and WS-Federation (which defines the creation and brokering of trust relationships within and across federations). The WS-* specifications enable web services to exchange a range of security tokens (such as passwords, digital certificates, SAML assertions or Kerberos tickets). WS-Federation is still in its infancy and has not yet been submitted for standardization or public review. RSA is actively involved in reviewing and contributing to the WS-* specifications and will support WS-Federation in the future.

Security Assertion Markup Language

The SAML standard is managed by the OASIS Security Services Technical Committee (SSTC). SAML is built on several industry-standard technologies that support other types of web services, including XML, SOAP (Simple Object Access Protocol) and XML Signature. RSA was one of the primary creators of SAML and is currently the co-chair of the SSTC. The company has donated royalty-free rights to several of its patents in order to facilitate industry-wide adoption.

SAML supports the establishment of trust relationships between security domains, so that one domain, when it receives a request for access, can evaluate that request using information provided by another domain. The domain that has the information is the “identity provider (IdP)” and the domain that is evaluating the access request is the “service provider (SP).”

The information takes the form of trusted statements (called “assertions”) about subjects (end users, web services or any other entity that can be assigned a digital identity). An assertion makes one or more statements about a subject. Every assertion includes an assertion ID (a unique identifier), an issuer identification string and a creation time stamp. Assertions may also contain additional data, such as conditions that define when the assertion is considered valid and a digital signature to ensure the integrity of the assertion data.

Figure 1 shows a simplified schematic for a SAML transaction between two domains in a trust relationship. A user who is authenticated in the source domain requests access in the destination

domain. The destination domain sends a query to the source domain where the user’s information resides and receives back an assertion that it uses to grant or deny access.

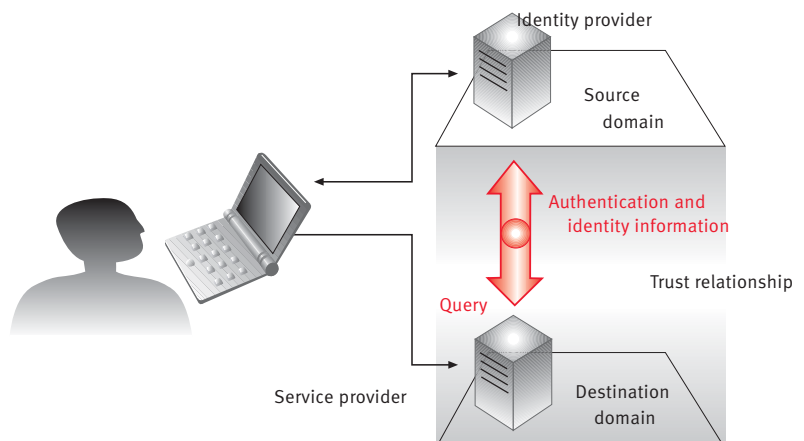
SAML defines three kinds of statements that can be carried within an assertion:

- **Authentication statement.** “This subject was authenticated by this means at this time.”
- **Attribute statement.** Provides specific details about the subject (for example, a user’s account status or membership level).
- **Authorization decision statement.** Identifies actions the subject is entitled to perform (for example, whether a user is permitted to buy a specified item).

SAML is most widely used for:

- **Web Single Sign-on.** With the support of SAML’s trust relationships between domains, authenticated web users can move between applications within their home security domain or among business partners’ environments without re-authenticating.
- **Attribute Service.** As a user moves from one web site or service to another, or accesses web services, those services can acquire authentication information as well as additional attributes of the user’s digital identity, such as their operational role, employee ID number or account balance.

Figure 1. A SAML transaction





- **Account Linking.** End-users can choose which accounts they would like to federate and then link those accounts themselves. This capability greatly empowers consumers and enables ease of access to various sites as end-users conduct business transactions with a multiple organizations.

The OASIS SSTC overseeing SAML has defined two web SSO profiles - flows of assertions and protocol messages that specify how SAML can be used to authenticate a user across domains - the Browser/Artifact Profile and the Browser/POST Profile. The Browser/Artifact Profile is a “pull” model in which a reference to the web SSO assertion (called an artifact) is sent to the service provider, which can use this reference to obtain (or pull) the assertion from the identity provider. Browser/POST Profile is a “push” model: an assertion is POSTed (using the HTTP POST command) directly to the relying party. The destination site then can make authentication and authorization decisions based on the received assertion contained within the POST message. These profiles are discussed in detail in a document available from the committee’s web site, “The SAML V2.0 Technical Overview” (<http://www.oasisopen.org/committees/download.php/13786/sstc-saml-tech-overview-2.0-draft-07-diff.pdf>).

RSA Federated Identity Manager: A Flexible Solution for Managing Identity Federation

Identity Manager is an enterprise-class identity federation solution that enables companies to securely share authentication and user identity information. It currently supports the SAML IdP/SP trust relationship and other SAML constructs. It is designed to be flexible and readily customizable, making it easy to add additional trust relationships and integrate with a customer’s existing user and attribute repositories. Identity Manager will evolve as support for other industry standards and protocols are needed. It includes many features that provide a complete system for managing federated identities using the SAML standard:

- A browser-based administration GUI and automated tools to speed deployment time,
- Enhanced security features such as digital signing and verification,
- Deployment models that accommodate virtually any architecture or vendor implementation.
- A plug-in architecture that supports integration with a company’s existing identity infrastructure and schema and provides flexible name mapping algorithms to resolve problems created by dissimilar user naming conventions.

RSA Federated Identity Manager is not an identity manager or authentication authority. Solutions such as RSA Access Manager can fill both of these roles. Rather, Identity Manager provides a way for authentication authorities - even those from different vendors - to communicate authentication and identity-related information securely to business partners: It provides the features needed to manage and use federated identities from different authorities.

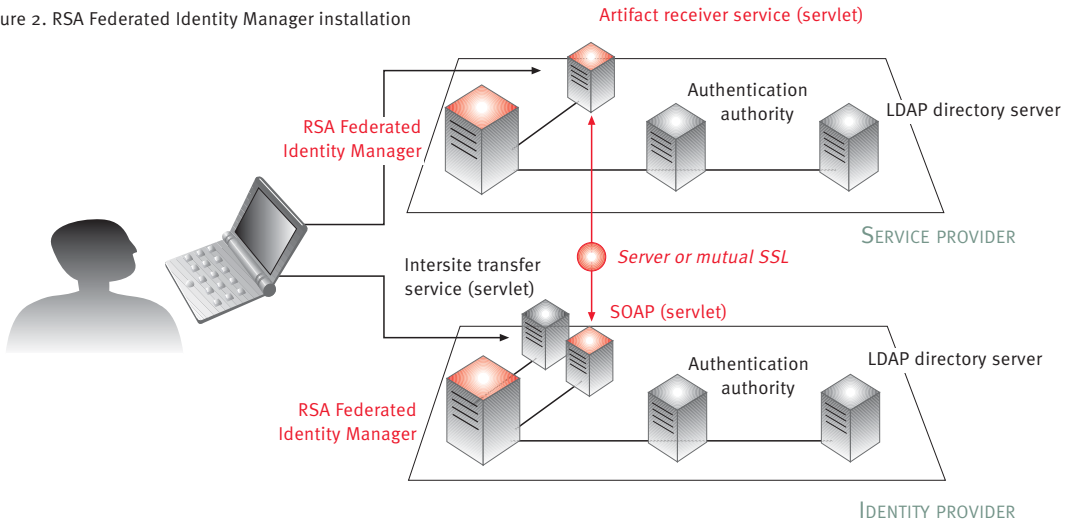
RSA Federated Identity Manager is a J2EE-compliant application. It is deployable in many different scenarios, including multi-tier environments.

In a typical deployment, Identity Manager is installed and integrated in an environment with an authentication authority. (For Identity Manager hardware requirements, see Appendix A.) See Figure 2.

When an identity provider creates a web SSO assertion, for example, RSA Federated Identity Manager interacts with the authentication and attribute authorities to acquire authentication attribute data for the user. It then packages it into a SAML protocol message and transmits it to the relying party. Identity Manager at the service provider site receives the message, unwraps the data and passes it to the local authentication and attribute authorities.

The tasks in this process that involve interaction with local site resources such as the authentication authority or attribute authority (e.g., LDAP repository) are handled by software plug-ins that communicate with the local resources and map local formats to and from SAML formats. These plug-ins are created as Java™ classes and managed through the Identity Manager administration GUI.

Figure 2. RSA Federated Identity Manager installation



RSA Federated Identity Manager makes use of four types of plug-ins:

- **Session.** When configured on an identity provider site, a ticket plug-in converts local authentication and session management mechanisms (e.g. cookies) to SAML authentication statements. Inversely, on a service provider site, this plug-in converts SAML authentication statements to local authentication mechanisms.
- **NameID mapper.** Maps local subject names to SAML-formatted subjects (for the IdP site) and SAML-formatted subjects to local subject names (for the SP site). NameID plug-ins support fixed identifiers (e.g., email address, user name), group identifiers, pseudonymous, and anonymous mapping.

- **Attribute.** Retrieves user attributes from a data repository, cookie, or HTML session context. Attributes retrieved at the identity provider site can be used by the service provider to make authorization decisions and to customize the user experience.
- **Connection.** Manages connection pooling to optimize input/output connections with data repositories that store user identities and attributes.

RSA Federated Identity Manager comes with default plug-ins designed to integrate with RSA Access Manager and RSA SecurID® products as well as any SQL and LDAP data store. However, the plug-in architecture makes it easy to configure Identity Manager to operate with other authentication authorities and user attribute repositories and to deal



Figure 3. Powerful and Intuitive Administration GUI

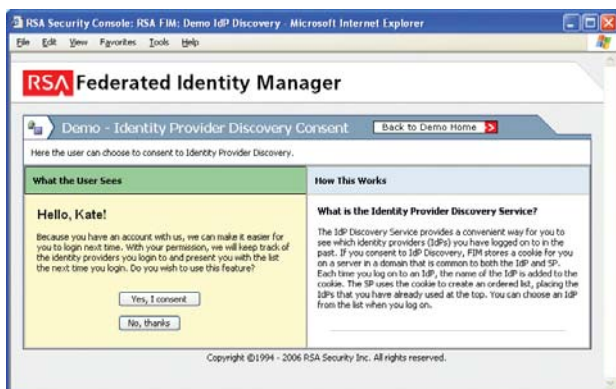


Figure 4. Integrated and Fully Customizable Testing Environment

with unique identity mapping and InterSite Transfer algorithms. The Identity Manager documentation details how to create custom plug-ins for integrating the solution with other identity management products. The default plug-in Sun® Java™ technology Java classes can be used as templates.

RSA Federated Identity Manager Support for SAML

Identity Manager has been certified by the Liberty Alliance Conformance Expert Group (CEG) and the U.S. General Services Administration (GSA) e-Authentication initiative for SAML interoperability. Identity Manager supports the most widely used SAML 2.0 use cases including IdP SSO, SP SSO, Enhanced Client and Proxy (ECP), Single Logout (SLO), account federation, nameID management, IdP discovery, and attribute queries. Identity Manager can be configured to function as an identity provider, a service provider, or both; and can exchange SAML assertions with any system that complies with the SAML standard.

RSA Federated Identity Manager supports the following types of assertions:

- **Web SSO.** Generated when a user attempts to access a remote web resource. Can contain multiple types of statements. Constructed and exchanged between parties using the SAML Browser/Artifact Profile or Browser/POST Profile.
- **Authentication.** Generated in response to an authentication query. Contains only authentication statements.
- **Attribute.** Generated in response to an attribute query. Contains only attribute statements.

RSA Federated Identity Manager supports the following types of requests:

- **Assertion artifact.** The Browser/Artifact Profile uses this request to retrieve a web SSO assertion.
- **Assertion ID reference.** Any service provider that knows an Assertion ID can use this request to ask for a specific assertion.
- **Query.** The identity provider determines the type of assertion to create based on the type of query it receives from the service provider. RSA Federated Identity Manager supports two types of SAML queries: attribute queries, which create attribute assertions, and authentication queries, which create authentication assertions.
- **NameID request.** Used by either the identity provider or the service provider to request a mapping between two accounts or to modify the opaque identifier used to by an existing mapping
- **Logout request.** The service provider issues this request when a user initiates a single logout. The identity provider forwards this request to all other service providers with which the user has an active session.

Administering RSA Federated Identity Manager

Identity Manager is configured and managed through a browser-based GUI that will be familiar to administrators of RSA® Access Manager Software and other RSA products. In this interface administrators create the trust relationships, defining identity providers and Services Providers, and configuring features such as digital signatures. The administration GUI is designed to be both powerful and easy to use. Tools such as the quick setup wizard allow new partners to be added and configured in a few short steps. Policy templates and cloning features simplify system configuration and allow specific tasks to be delegated to subject matter experts. For example, security administrators can define one or more signing and encryption policies, which can then be reused by other administrators to set up new partner connections.

The administration GUI is highly intuitive and can make even complex tasks much simpler. Wizards and context sensitive help are available to guide administrators through many tasks, and business rules are automatically applied to suggest appropriate defaults and to help debug configuration errors.

Out of the box, RSA Federated Identity Manager includes a self-contained SAML demo complete with sample web pages, identity provider and service provider configurations, plug-ins, and an integrated tutorial. The demo can be used to quickly learn about RSA Federated Identity Manager and its feature rich SAML use cases, or can be deployed immediately to verify actual SAML interoperability between domains before integrating the product with your authentication environment.

Summary

RSA Federated Identity Manager is an enterprise-ready identity federation solution that solves real-world business problems, helping organizations to unlock the true potential of business relationships while maintaining consistent and centralized control over the policies associated with users and applications.

It is designed to be fully standards-based and compatible with other systems. It is based on the latest standards for web services and security, including XML, SOAP, SAML and the Liberty Alliance specifications. These technologies that enable federation are maturing rapidly - and RSA has been in the forefront of their development.

This involvement has made RSA Federated Identity Manager a complete enterprise solution - and will continue to make it a technology leader with the ease of deployment, compatibility and security features that enterprises require.



Appendix A: RSA Federated Identity Manager Specifications

Supported Platforms

- Microsoft Windows 2003
- Sun® Solaris® 9
- Red Hat Linux AS and ES 3.0
- SuSE Linux Enterprise Server 9

Supported Directory Servers (for storing configuration data)

- Sun Java System Directory Server 5.2
- Microsoft Active Directory® 2003
- BEA WebLogic Server Embedded LDAP 8.1

Supported Browsers

- Internet Explorer 5.5 and later
- Netscape 6.2.3 and later
- Mozilla 1.7 (Linux)

Minimum Hardware Requirements

- 1 GHz CPU
- 1 GB free disk space
- 1 GB RAM

Digital Signing Requirements

If you are using digital signing, you need a Certificate Authority. RSA Federated Identity Manager is qualified to work with leading standards-based Certificate Authorities including the RSA® Certificate Manager and ships with a free limited use license for this product to issue digital certificates for use in digital signing and establishing SSL connections. For more information on this product, see your RSA sales representative.

Appendix B: Resources

For more information on RSA and RSA Federated Identity Manager, please see the following sources:

- RSA Federated Identity Home: <http://www.RSA.com/node.asp?id=1191>
- Federated Identity—Seizing Business Opportunities by Sharing Trusted Electronic Identities: http://www.RSA.com/content_library.asp
- RSA Federated Identity Manager and Return on Investment: http://www.RSA.com/content_library.asp
- RSA Access Manager : <http://www.RSA.com/node.asp?id=1186>

For more information on identity federation concepts and standards, please see the following sources:

- Security Assertion Markup Language (SAML): <http://www.oasis-open.org/committees/security>
- Liberty Alliance ID-FF: <http://www.projectliberty.org/resources/specifications.php#box1>
- WS-* identity federation standards: <http://msdn.microsoft.com>

Appendix C: Glossary of Identity Federation Terms and Concepts

These definitions are abstracted from two sources: the OASIS Security Services Technical Committee document Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0 (available at <http://www.oasis-open.org/committees/download.php/11886/oasis-sstc-saml-glossary-2.0-os.pdf>) and the Liberty Alliance Project's Liberty Technical Glossary Version: v1.3 (available at <http://www.projectliberty.org/specs/draft-liberty-glossary-1.3-errata-v1.0.pdf>).

Administrative Domain—An environment or context that is defined by some combination of one or more administrative policies, Internet Domain Name registrations, civil legal entities (for example, individuals, corporations or other formally organized entities), plus a collection of hosts, network devices and the interconnecting networks (and possibly other traits), plus (often various) network services and applications running upon them. An administrative domain may contain or define one or more security domains. An administrative domain may encompass a single site or multiple sites. The traits defining an administrative domain may and, in many cases will, evolve over time. Administrative domains may interact and enter into agreements for providing and/or consuming services across administrative domain boundaries.

Affiliation—In Liberty, an affiliation is a set of one or more entities, described by provider ID's, who may perform Liberty interactions as a member of the set. An affiliation is referenced by exactly one affiliation ID and is administered by exactly one entity identified by their provider ID. Members of an affiliation may invoke services either as a member of the affiliation (using affiliationID) or individually (using their provider ID). Affiliation and affiliation group are equivalent terms.

Affiliation ID—In Liberty, an Affiliation ID identifies an affiliation. It is schematically represented by the affiliation ID attribute of the <AffiliationDescriptor> metadata element.

Assertion—A piece of data produced by a SAML authority regarding either an act of authentication performed on a subject, attribute information about the subject or authorization permissions applying to the subject with respect to a specified resource. As used in Liberty, assertions typically concern things such as: an act of authentication performed by a Principal, attribute information about a Principal or authorization permissions applying to a Principal with respect to a specified resource.

Asserting Party—Formally, the administrative domain that hosts one or more SAML authorities. Informally, an instance of a SAML authority.

Attribute—A distinct characteristic of an object (in SAML, of a subject). An object's attributes are said to describe it. Attributes are often specified in terms of physical traits, such as size, shape, weight and color, etc., for real-world objects. Objects in cyberspace might have attributes describing size, type of encoding, network address and so on. Which attributes of an object are salient is decided by the beholder. See also XML attribute.

Attribute Authority—A system entity that produces attribute assertions.

Attribute Assertion—An assertion that conveys information about attributes of a subject.

Authentication—To confirm a system entity's asserted principal identity with a specified, or understood, level of confidence.

Authentication Assertion—An assertion that conveys information about a successful act of authentication that took place for a subject. In the Liberty specification suite, an authentication assertion contains a <lib:AuthenticationStatement>. Note that the foregoing element is defined in a Liberty namespace. Also known as Liberty authentication assertion and ID-FF authentication assertion. Liberty authentication assertions are formal XML extensions of SAML assertions.



Authentication Authority—A system entity that produces authentication assertions. In the Liberty architecture, it is typically an identity provider (synonymous with authenticating identity provider or authenticating IdP). An identity provider that authenticated a Principal

Authentication, Authorization and Accounting Services (AAA)—Three system functions that are the underpinning of a security service: authentication recognizes the user; authorization enforces access controls and delivers services; accounting tracks users' usage of system resources.

Authorization—The process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of access to a particular resource. Usually, authorization is in the context of authentication. Once a subject is authenticated, it may be authorized to perform different types of access.

Authorization Decision—The result of an act of authorization. The result may be negative: that is, it may indicate that the subject is not allowed any access to the resource.

Authorization Decision Assertion—An assertion that conveys information about an authorization decision.

Bearer token—In Liberty, a bearer token is a form of security token that connotes some attribute(s) to its holder. Typically bearer tokens connote identity and they consist essentially of credentials of some form, e.g. SAML assertions.

Binding, Protocol Binding—An instance of mapping SAML request-response message exchanges into a specific protocol. Each binding is given a name in the pattern "SAML xxx binding".

Circle of Trust (CoT)—A federation of service providers and identity providers that have business relationships based on Liberty architecture and operational agreements and with whom users can transact business in a secure and apparently seamless environment. Also known as a Trust Circle.

Discoverable—A discoverable "in principle" service is one having a service type URI assigned (this is typically in done in the specification defining the service). A discoverable "in practice" service is one that is registered in some discovery service instance. ID-WSF services are by definition discoverable "in principle" because such services are assigned a service type URI facilitating their registration in Discovery Service instances.

Discovery Service (DS)—An ID-WSF service facilitating the registration, and subsequent discovery of, ID-WSF service instances. See also discoverable.

ID-*—A shorthand designator referring to the Liberty ID-WSF, ID-FF and ID-SIS specification sets. For example, one might say that the former specification sets are all part of the Liberty ID-* specification suite.

ID-* fault message—A SOAP <S:Fault> element containing a <Status> element, with the attributes—and attribute values of both elements configured as specified herein or as specified in other specification(s) in the ID-WSF or ID-SIS specification sets.


ID-* message—Equivalent to ordinary ID-* message.

ID-FF—The Identity Federation Framework (ID-FF) is the title for a subset of the Liberty specification suite which defines largely HTTP-based protocols for web single sign-on and identity federation.

ID-PP—The "ID Personal Profile" is an ID-SIS -based service which can provide profile information regarding Principals, typically subject to policy established by those Principals.

ID-SIS—Liberty Identity Service Interface specification set. ID-SIS-based services are identity services typically built on ID-WSF.

ID-WSF—Liberty Identity Web Services Framework specification set. An ID-WSF-based service is an identity service that is at least discoverable in principle and is based on the Liberty specifications for SOAP bindings and security mechanisms.



Identifier—A representation (for example, a string) mapped to a system entity that uniquely refers to it.

Identity—The essence of an entity. One’s identity is often described by one’s characteristics, among which may be any number of identifiers.

Identity provider (IdP)—A Liberty-enabled system entity that manages identity information on behalf of Principals and provides assertions of Principal authentication to other providers.

Identity service—In Liberty, an abstract notion of a web service whose operations are indexed by identity. Such a service might maintain information about, or on behalf of, identities or perform actions on behalf of identities.

Liberty-enabled client or proxy (LECP)—A Liberty-enabled client is a client that has, or knows how to obtain, knowledge about the identity provider that the Principal wishes to use with the service provider. A Liberty-enabled proxy is an HTTP proxy (typically a WAP gateway) that emulates a Liberty-enabled client.

Liberty-enabled Provider—An umbrella term referring to any Provider offering any ID-FF-, ID-WSF- or ID-SIS-based services.

Liberty-Enabled Client and Proxy Profile—This profile specifies interactions between Liberty-enabled clients and/or proxies, service providers and identity providers [LibertyBindProf].

Liberty-enabled User Agent or Device (LUAD)—A user agent or device that has specific support for one or more profiles of the Liberty specifications. It should be noted that although a standard web browser can be used in many Liberty-specified scenarios, it does not provide specific support for the Liberty protocols and thus is not a LUAD. No particular claims of specific functionality should be implied about a system entity solely based on its definition as a LUAD. Rather, a LUAD may perform one or more Liberty system entity roles as defined by the Liberty specifications it implements. For example, a LUAD-LECP is a user agent or device that supports the Liberty LECP profile and a LUAD-DS would define a user agent or device offering a Liberty ID-WSF Discovery Service.

Markup Language—A set of XML elements and XML attributes to be applied to the structure of an XML document for a specific purpose. A markup language is typically defined by means of a set of XML schemas and accompanying documentation. For example, the Security Assertion Markup Language (SAML) is defined by two schemas and a set of normative SAML specification text.

Ordinary ID-* message—A Liberty Identity Web Services Framework (ID-WSF) or Service Interface Specification (ID-SIS) message. It is designed to be conveyed by essentially any transport or transfer protocol, notably SOAP. It is also known among the ID-* specifications as a service request or an ID-WSF (service) request or an ID-SIS (service) request.

Policy Decision Point (PDP)—A system entity that makes authorization decisions for itself or for other system entities that request such decisions. For example, a SAML PDP consumes authorization decision requests and produces authorization decision assertions in response. A PDP is an “authorization decision authority”.

Policy Enforcement Point (PEP)—A system entity that requests and subsequently enforces authorization decisions. For example, a SAML PEP sends authorization decision requests to a PDP and consumes the authorization decision assertions sent in response.

Principal—A system entity whose identity can be authenticated. In Liberty usage, Principal is usually synonymous with a “natural person”. A Principal’s identity may be federated. Examples of Principals include individual users, groups of individuals, organizational entities, e.g., corporations, or a component of the Liberty architecture.

Principal Identity—A representation of a principal’s identity, typically an identifier.

Privacy—In Liberty, proper handling of personal information throughout its life cycle, consistent with the preferences of the subject.



Profile—In SAML, a set of rules describing how to embed assertions into and extract them from a framework or protocol. Each profile is given a name in the pattern “xxx profile of SAML”. In Liberty, a profile is data comprising attributes that may be maintained on behalf of an system entity (usually a Principal), over and beyond its various identifiers. At least some of this information (for example, addresses, preferences, card numbers) is typically provided by the Principal.

Provider—A Liberty-enabled entity that performs one or more of the provider roles in the Liberty architecture—for example service provider or identity provider. Providers are identified in Liberty protocol interactions by their Provider IDs or optionally an Affiliation ID.

Relying Party—A system entity that decides to take an action based on information from another system entity. For example, a SAML relying party depends on receiving assertions from an asserting party (a SAML authority) about a subject.

Requester, SAML Requester—A system entity that utilizes the SAML protocol to request services from another system entity (a SAML authority, a responder). The term “client” for this notion is not used because many system entities simultaneously or serially act as both clients and servers. In cases where the SOAP binding for SAML is being used, the SAML requester is architecturally distinct from the initial SOAP sender.

Resource—a) Data contained in an information system (for example, in the form of files, information in memory, etc). b) A service provided by a system. SAML refers to resources by means of URI references.

Responder, SAML Responder—A system entity (a SAML authority) that utilizes the SAML protocol to respond to a request for services from another system entity (a requester). The term “server” for this notion is not used because many system entities simultaneously or serially act as both clients and servers. In cases where the SOAP binding for SAML is being used, the SAML responder is architecturally distinct from the ultimate SOAP receiver.

Rights Expression Language (REL)—In Liberty, a Rights Expression Language facilitates the expression of who are the “rights holders” for a resource, who is authorized to use a resource and their applicable permissions, and any constraints or conditions imposed on such permissions. They also may express “rights entities” and “rights transactions”.


SAML Authority—An abstract system entity in the SAML domain model that issues assertions. See also attribute authority, authentication authority, and policy decision point (PDP).

Security—A collection of safeguards that ensure the confidentiality of information, protect the systems or networks used to process it and control access to them. Security typically encompasses the concepts of secrecy, confidentiality, integrity and availability. It is intended to ensure that a system resists potentially correlated attacks.

Security Architecture—A plan and set of principles for an administrative domain and its security domains that describe the security services that a system is required to provide to meet the needs of its users, the system elements required to implement the services and the performance levels required in the elements to deal with the threat environment. A complete security architecture for a system addresses administrative security, communication security, computer security, emanations security, personnel security and physical security, and prescribes security policies for each. A complete security architecture needs to deal with both intentional, intelligent threats and accidental threats. A security architecture should explicitly evolve over time as an integral part of its administrative domain’s evolution.

Security Assertion—An assertion that is scrutinized in the context of a security architecture.

Security Assertion Markup Language, SAML—The set of specifications describing security assertions that are encoded in XML, profiles for attaching the assertions to various protocols and frameworks, the request/response protocol used to obtain the assertions and bindings of this protocol to various transfer protocols (for example, SOAP and HTTP).



Security Domain—An environment or context that is defined by security models and a security architecture, including a set of resources and set of system entities that are authorized to access the resources. One or more security domains may reside in a single administrative domain. The traits defining a given security domain typically evolve over time.

Security Policy—A set of rules and practices that specify or regulate how a system or organization provides security services to protect resources. Security policies are components of security architectures. Significant portions of security policies are implemented via security services, using security policy expressions.

Security Policy Expression—A mapping of principal identities and/or attributes thereof with allowable actions. Security policy expressions are often essentially access control lists.

Security Service—A processing or communication service that is provided by a system to give a specific kind of protection to resources, where said resources may reside with said system or reside with other systems, for example, an authentication service or a PKI-based document attribution and authentication service. A security service is a superset of authentication, authorization and accounting (AAA) services. Security services typically implement portions of security policies and are implemented via security mechanisms.

Security Token—In Liberty, a security token is a collection of security-related information that is used to represent and substantiate a claim. Outside of Liberty, the term “security token” often refers to hardware-based devices, e.g. so-called “token cards”. One should not confuse the latter and the former definitions. However, it is possible for some given authentication mechanism to employ token cards in the process of authentication.

Session—A lasting interaction between system entities, often involving a user, typified by the maintenance of some state of the interaction for the duration of the interaction.

Simple Authentication and Security Layer (SASL)—An approach to modularizing protocol design such that the security design components, e.g. authentication and security layer mechanisms, are reduced to a uniform abstract interface. This facilitates a protocol’s use of an open-ended set of security mechanisms, as well as a so-called “late binding” between implementations of the protocol and the security mechanisms’ implementations. This late binding can occur at implementation- and/or deployment-time. The SASL specification also defines how one packages authentication and security layer mechanisms to fit into the SASL framework, where they are known as SASL mechanisms, as well as register them with the Internet Assigned Numbers Authority for reuse.

Site—An informal term for an administrative domain in geographical or DNS name sense. It may refer to a particular geographical or topological portion of an administrative domain or it may encompass multiple administrative domains, as may be the case at an ASP site.

SSO Assertion, Single Sign-on Assertion—An assertion with conditions embedded that explicitly define its lifetime and include one or more statements about the authentication of a subject. Additional information about the subject, such as attributes, may also be included in the assertion.

Subject—A principal in the context of a security domain. SAML assertions make declarations about subjects.

System Entity—An active element of a computer/network system. For example, an automated process or set of processes, a subsystem, a person or group of persons that incorporates a distinct set of functionality.

Transport Layer Security Protocol (TLS)—An evolution of the SSL protocol. The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering or message forgery.



Trusted Authority—In Liberty, a Trusted Third Party (TTP) which issues and vouches for assertions, otherwise known as an identity provider.

Trusted Third Party—In general, a security authority or its agent, trusted by other entities with respect to security-related activities. In the context of Liberty, these other entities are, for example, Principals and service providers and the trusted third party is typically the identity provider(s) involved in the particular interaction of interest

Ultimate SOAP Receiver—The SOAP receiver that is a final destination of a SOAP message. It is responsible for processing the contents of the SOAP body and any SOAP header blocks targeted at it. In some circumstances, a SOAP message might not reach an ultimate SOAP receiver, for example because of a problem at a SOAP intermediary. An ultimate SOAP receiver cannot also be a SOAP intermediary for the same SOAP message.

User—A natural person who makes use of a system and its resources for any purpose.

Uniform Resource Identifier (URI)—A compact string of characters for identifying an abstract or physical resource. URIs are the universal addressing mechanism for resources on the World Wide Web. Uniform Resource Locators (URLs) are a subset of URIs that use an addressing scheme tied to the resource's primary access mechanism, for example, their network "location".

URI Reference—A URI that is allowed to have an appended number sign (#) and fragment identifier. Fragment identifiers address particular locations or regions within the identified resource.

XML—Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.

XML Attribute—An XML data structure that is embedded in the start-tag of an XML element and that has a name and a value. For example, the italicized portion below is an instance of an XML attribute:

```
<Address AddressID="A12345">...</Address>
```

See also attribute.

XML Element—An XML data structure that is hierarchically arranged among other such structures in an XML document and is indicated by either a start-tag and end-tag or an empty tag. For example:

```
<Address AddressID="A12345">  
<Street>105 Main Street</Street>  
<City>Springfield</City>  
<State Or Province>  
<Full>Massachusetts</Full>  
<Abbrev>MA</Abbrev>  
</State Or Province>  
<Post Code="567890"/>  
</Address>
```

XML Namespace—A collection of names, identified by a URI reference, which are used in XML documents as element types and attribute names. An XML namespace is often associated with an XML schema. For example, SAML defines two schemas and each has a unique XML namespace.

XML Schema—The format developed by the World Wide Web Consortium (W3C) for describing rules for a markup language to be used in a set of XML documents. In the lowercase, a "schema" or "XML schema" is an individual instance of this format. For example, SAML defines two schemas, one containing the rules for XML documents that encode security assertions and one containing the rules for XML documents that encode request/response protocol messages. Schemas define not only XML elements and XML attributes, but also data types that apply to these constructs.



RSA is your trusted partner

RSA, The Security Division of EMC, is the expert in information-centric security, enabling the protection of information throughout its lifecycle. RSA enables customers to cost-effectively secure critical information assets and online identities wherever they live and at every step of the way, and manage security information and events to ease the burden of compliance.

RSA offers industry-leading solutions in identity assurance & access control, encryption & key management, compliance & security information management and fraud protection. These solutions bring trust to millions of user identities, the transactions that they perform, and the data that is generated. For more information, please visit www.RSA.com and www.EMC.com.

©2007 RSA Security Inc. All Rights Reserved. RSA, RSA Security, and the RSA logo are either registered trademarks or trademarks of RSA Security Inc. in the United States and/or other countries. Windows and Microsoft are registered trademarks or trademarks of the Microsoft Corporation in the U.S. and/or other countries. EMC is a registered trademark of EMC Corporation. All other products and services mentioned are trademarks of their respective companies.

FIMTO SB 0707



The Security Division of EMC

RSA Security Inc.
RSA Security Ireland Limited
www.rsa.com