| Carnegie Mellon COMPUTING SERVICES | IdM Glossary of Terms |
|---|---|
| **Author (email):** | **Mark Poepping (poepping@cmu.edu)** |
| **Revision Date** | **1/30/2009** |

## Purpose

This glossary is meant as a reference to help in understanding terms used in the IdM White Paper and other documents related to identity, authentication, and authorization. It was assembled from a number of sources to help reduce the confusion inherent in technical discussions in this area.

### References

Valued references include:

- http://www.nmi-edit.org/roadmap/draft-authn-roadmap-03/Resources/authn_roadmap_060828.pdf
- http://wiki.idcommons.net/Lexicon
- http://wiki.idcommons.net/Identipedia
- http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf
- http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
- http://www.ietf.org/rfc/rfc2828.txt
- http://www.wikipedia.org/
- http://middleware.internet2.edu/eduperson/docs/internet2-mace-dir-eduperson-200806.html
- http://middleware.internet2.edu/dir/groups/docs/internet2-mace-dir-groups-best-practices-200210.htm
- http://middleware.internet2.edu/signet/docs/internet2-mace-signet-privmgmt-recipe-02.html
- http://msdn.microsoft.com/en-us/architecture/cc836389.aspx

## IdM Terms and Definitions

**Affiliation** specifies a person's relationship(s) to the institution in broad categories; e.g. student, faculty, staff, alumni.

**Attestation** is assurance of the veracity of *Identity Attributes* asserted through an *authentication* process. *Identity Providers* attest to the quality of the identity information wrapped in *credentials* they generate. *Attestation* is only as good as the trust one places in the *Identity Provider* (*IdP*) and the *Level of Assurance (LoA)* practices the *IdP* uses to generate and protect the *credentials*.

**Authentication** is the process of validating a *credential* and associating the enclosed *identity attributes* with a *session*. A *credential* may contain a verifier (e.g. an X.509 certificate uses a cryptographic signature) or it may require collecting a verifier at runtime. A runtime verifier may be single-factor or multi-factor. Single-factor is most commonly a secret key (a password), multi-factor requires use of at least two **different** methods for verification (most commonly a smart card and a PIN). Authentication alone should not afford any *entity* access to resources, that function should be controlled through a separate *authorization* step, leveraging the *identity attributes* gained through the *authentication* step.

The most common example is where a system needs to authenticate a user. Generally the system presents a dialogue requesting a username (the *credential*) and a password (the *verifier*). The system retains username for subsequent work (the *session*), and often will leverage other sources (e.g. directories) to acquire *identity attributes* associated with that username.

A second example is where a system needs to authenticate another system (or process or service). This is increasingly important to the security of multi-component systems – which are the norm in modern computing infrastructures (ultimately characterized by SOA). Commonly, when two systems are trying to talk to each other, they should authenticate each other using *credentials* and *verifiers.* At present, this is most commonly done using X.509 *credentials* over SSL (Secure Sockets Layer), but there are many alternatives. Which alternatives are allowed and how they are used is subject to the *Identity Management* architecture for a particular organization.

**Authorization** is the technical step of allowing or denying access to resources based on business rules created by the service owner (subject to enterprise policy). The business rules are generally expressed as access control lists that leverage *identity attributes* that are defined and maintained by the enterprise. There is wide variety in the architecture and style of expressing, mixing, and optimizing *identity attributes, roles, privilege,* and access control lists (ACLs) for efficient management of *authorization* across the enterprise.

**Authorization Audit** is a process to identify and validate access capabilities in the enterprise. This could be comprehensive (all access to all resources) but is generally done in reference to a specific resource or related to a particular entity.

**Claim (Assertion)** is a statement of the value of one or more *identity attributes*; e.g. if Ben is currently a CIT freshman, the *Identity Management system* could create a *credential* containing the *claim*: "Ben's *affiliation* is student".

**Credential** is an object that is verified when presented during an authentication transaction. Credentials consist of one or two elements:

1. *Identity Attributes* (required): most often just a single identifier (e.g. username) associated with the entity being authenticated. However, in many circumstances, other identity attributes may be required (e.g. assertion of a **right to use** license for a particular resource)

2. Verifier (optional as part of the credential, may be provided separately from the *identity attributes* at authentication time).

The *identity attributes* contained in the credential are no more reliable than the *identification* and *registration* processes that precede it. The relative confidence that may be placed in the information is generally indicated by the *level of assurance* for the credential.

**Enterprise Identity Data Model** (sometimes referred to as an **Identity Profile**) is a reference for common data elements for entities associated with the enterprise (e.g. persons, objects, groups, applications). Not that it engages the entirety of useful data, but that it represents an evolvingset of common, valuable, and reliable attributes that may be trusted for accuracy and governance of semantics over time. The EDUPerson schema is commonly used as an extensible base format to house an *enterprise identity data model* for higher education institutions.

**Entitlement** indicates eligibility for a given service, and is generally assigned through evaluation of policy language to indicate the target audience(s) (e.g. students can have an email account). An *entitlement* is generally not used directly for authorization to access a resource, rather the concept of p*rivilege* is added for finer-grained control. For instance, a new employee may be *entitled* to access an email resource, but they will have access only after the resources are *provisioned* and *privilege* is granted.

**Entity** is a software representation (within an enterprise *Identity Management System*) of a physical, social, or ephemeral construct that is useful for implementing enterprise computing systems. E*ntities* generally are associated with an *entity* type (e.g. a person, process, organization, system, *persona* etc.), and contain *identity attributes* that are associated with the represented construct.

**Federation** is a negotiated agreement among *Identity Providers* and *Service Providers* that defines the terms and conditions for sharing and relying on *Identity Attributes* contained in *credentials* that are exchanged between *Federation* members.

**Group** is a software construct that manages sets of things, most commonly *entities* and/or other *groups*. *Groups* are often used to represent roles or other affinities among entities; they can greatly simply access control. For example, a *group* may hold the set of email addresses subscribed to a particular newsletter. A *group* may hold a list of users allowed to enter a building. A group may hold a list of groups that in turn hold lists of machines allowed to run a set of software applications.

**Identification** is the process by which information about a person is gathered and used to provide some assurance that the *identity attributes* are accurate to enterprise purposes. Generally, identity verification takes place within the office (e.g. Human Resources or Student Services) that first encounters the individual and creates their record within the institutional *system(s) of record*. Identification is a prerequisite to *registration* (see below).

**Identifier** (**ID**) is a label (commonly a string of text) that names an entity. Naming an entity makes it possible to refer to it. Many *entities* have multiple identifiers that are useful for different contexts. The Andrew ID (e.g. userid@andrew.cmu.edu) is a perfectly good identifier for many uses. Historically, SSN has been used inappropriately as an identifier, such misuse should be eradicated.

**[Digital] Identity** is a set of *identity attributes* retained by an *identity provider* associated with a given *entity*. We retain those attributes of personal identity that are useful to our enterprise, others are out of scope.

**Identity Attribute** is a property of an entity. Enterprises generally collect and retain relevant attributes to provide services to customers. Examples of *identity attributes* include: one or more identifiers (e.g. Andrew userID), telephone number, home address, SSN, role. Policy and process for handling particular attributes is often governed by law or norms of privacy and common business practice (e.g. vaulting SSN information).

**Identity Provider** (IdP) is an organization that builds or leverages relationships with users and *service providers* in order to broker transactions between them. An *identity provider* generally leverages *identity management* processes and software to build relevant *identity attributes* for the users it supports, and *attests* to the values of those attributes to *service providers* by placing *claims* in *credentials* for consumption by the *service providers*.

**Identity Management** is an integrated system of business processes, policies, and technologies that enable organizations to facilitate and control their users' access to online applications and resources — while protecting confidential personal and business information from unauthorized use. It represents a category of interrelated solutions that are employed to administer user authentication, access rights and restrictions, account profiles, passwords, and other attributes supportive of users' roles for one or more applications or systems.

**Identity Proofing** is the process used to map a physical person to a *digital identity*. This is often done as one aspect of the Registration stage by requiring a physical credential such as a passport or driver's license.

**Level of Assurance** (LoA) describes the degree of certainty that the user has presented a credential that accurately refers to his or her true identity. There are several emerging standards for definition and practice around LoA (e.g. NIST 800-63). In this context, level of assurance is defined as:

- the degree of confidence in the proofing process used to establish the identity of the individual to whom the credential was issued, and
- the degree of confidence that the individual who uses the credential is the individual to whom the credential was issued.

A variety of regulatory and application requirements may lead to the need for a higher assurance in the credential provided for access to an application.

**Persona** is a *Digital Identity* (i.e. group of *identity attributes)* that a user has the ability to select and use to represent themselves in a given context. For example, a staff member may be both an application administrator and a user. It may be desirable (or necessary) to separate the roles and allow choice of attribute set appropriate to a given occasion to ensure that least privilege can apply.

**Privilege** is an additional construct created to facilitate management of access and modification for resources. *Privilege* is often conceptualized as a table to map *identity attributes* to manipulation methods for particular data based on express circumstances. This is finer-grained and more flexible

than both entitlements and access control lists, and therefore is a valuable complement. *Privileges* are usually derived, audited, and maintained through combined evaluation of *entitlement* and application access policy.

For example, a *privilege* may be used to enable [role of Business Analyst] for [Computing Services Capital Account] to [issue purchase requests up to $2000], but only [during normal business hours] and [not within 3 business days of quarterly closing]. The n-tuple is often expressed as: an [*entity* with given *identity attributes*] can access [an object] using a given [method] under [certain circumstances].
A *privilege* is easily suspended or overridden temporarily, e.g. if a credentials have been compromised. *Privilege* suspension does not imply de-provisioning of resources (or revocation of *entitlement*).

**Privilege Management** is a system that enables a resource owner to define and assign specific *privileges* for applications, guided by enterprise policy and business practice. This generally includes the ability to delegate the creation of additional *privileges*.

The definition and use of a *privilege management system* requires careful coordination among application owners, enterprise policy stewards, and enterprise data designers, to ensure that the defining constructs and data models enable reasonable sharing that can be effectively leveraged to externalize and normalize access control policy for more efficient access management and audit. There is great flexibility (and ambiguity) in the definition of roles and attributes as a basis for assignment of privilege, and as such there is potential for confusion and disagreement.

**Provisioning** is the action of facilitating allocation of resources within services and applications to enable users to leverage capabilities in accordance with the entitlements they have been granted.

**Registration** (*credentialing*) is the process whereby users are given electronic credentials, leveraging the identification process to ensure that they are coupled with the correct electronic identity information. Since multiple *registration* mechanisms may leverage a single *identification* process, the two are defined separately.

**Role** is an *identity attribute* that is defined to facilitate assignment of *privilege* for the purpose of access management. Roles may also be implemented with the use of *groups*.

**RBAC** is *role* based access control. The model is based on the notion that all access provided to users is provided by assigning users to well understood business *roles*. The assignment of *roles* grants a user a related set of *entitlements* which allows for the provisioning of *authentication* and *authorization* services.

**SAML** is the Security Assertion Markup Language; an XML-based standard for exchanging *authentication*, *authorization* and attribute data, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee.

**Service Provider** is an entity that manages resources for use by customers. In this context, the *service provider* is expected to negotiate, leverage, and trust the capabilities and mechanisms of an *identity provider* to manage the *identity* protocols and policies required to operate the service in a responsible manner (specifics will depend on the nature of the service).

**Session** is a temporary exchange and caching of information between *entities* generally to support transactions between them. For the purposes of this Identity glossary, the most critical information maintained in a *session* includes *claims* for one or both communicating *entities* and time-out information for the session. Depending on the requirements for the particular session, other information may also be included, such as *LoA* of *credentials*, encryption keys for secure communication, endpoint addresses, etc. A *session* time-out value is important to limit the risk of cached *credentials*.

**Single Sign-on Authentication**, or SSO, allows users to login once and gain access to multiple applications for a defined time period without having to re-login each time: subsequent application sessions may be initiated without further user interaction or interruption. SSO is most often used to refer to "Web Single Sign-on," however it can also be implemented outside the web.

**System of Record** is the authoritative source for a given data element or *identity attribute*. While there may be several different business functions that need to access and update a specific data element, all updates should be brokered and all access granted (or cached) directly via the *system of record* for that *identity attribute*. Different *identity attributes* may have different SoR's (subject to the *Enterprise Identity Data Model*), but the key characteristic of **every** *SoR* is that it **must** be online and available to respond to requests for *identity attributes* it masters.

**Verifier** is additional information that corroborates the binding between the entity and the identifier – this is most often a password (when a user is binding to a username). However, cryptographic signatures are generally used for electronic verification of attributes between online entities (as with X.509 certificates).

## Revision Log

| **Origination Date:** | 1/30/2009 | **Author:** | Mark Poepping |
|---|---|---|---|

| Revision Date | Reason for Change | Author |
|---|---|---|
| | | |