**IDAM Terminology:**

**What is Identity?**

An identity is a virtual representation of resource user by including employees, customers, partners and vendors.

**What is I & AM or IDAM?**

“*Identity management is the set of business processes, and a supporting infrastructure, for the creation, maintenance, and use of user identities*.”

It is emerged as a critical foundation for realizing the business benefits in terms of

* Enhance security
* Enhance regulatory compliance
* Cost savings
* Auditing
* Reduced Risk & security breach
* management control
* Improve operational efficiency
* most importantly, business growth
* Reduce time to deliver applications and services

Enterprises need to manage access to information and applications scattered across internal and external application systems. Moreover, they must provide this access for a growing number of identities, both inside and outside the organization, without compromising security or exposing sensitive information. IAM comprises of people, processes and products to manage identities and access to resources of an enterprise. Additionally, the enterprise shall have to ensure the correctness of data in order for the IAM Framework to function properly. IAM components can be classified into 4 major categories: authentication, authorization, user management and central user repository (Enterprise Directory).

 “*One identity per individual*.”

The ultimate goal of IAM product is to provide

“*The right people with the right access at the right time*”

**Why we need I & AM?**

IAM Product gives glance on below:

* Control who has access to which resources
* Know what is happening in your environment
* Know what to do about it
* Have the tools necessary to take action
* I & am product address the below issues:

- Lower productivity

- Duplicate and conflicting information

- Lack of information security

- Inability to comply with audits and regulations

- Central Identity administration

The identity management system allows administrators to centrally manage multiple identities. Administrators can centrally manage both the content within the identity management system and the structural architecture of the identity management system.

**Access management:**

 You almost never see "identity management" without this term right next to it. In fact, a number of vendors and analysts are combining the two into a single concept: IAM (identity and access management). It refers to the processes and technologies used to control and monitor network access. Access management features, such as authentication, authorization, trust and security auditing, are part and parcel of the top ID management systems.

**Credential**

 An identifier employed by the user to gain access to a network. It's the user's password, public key infrastructure (PKI) certificate or biometric information (fingerprint, retinal scan).

**Provisioning**

 The process of creating identities, defining their access privileges and adding them to an ID repository& Target Systems.

**De-provisioning**

 The process of removing an identity from an ID repository and terminating access privileges.

**Reconciliation:**

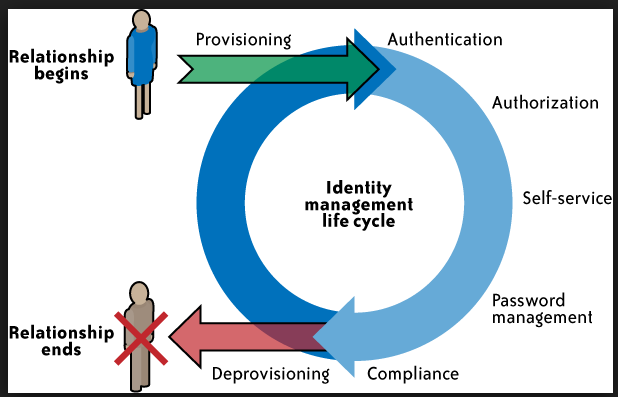
It is the process by which IDM receives information from resources.

Reconciliation is a process of pulling user data from target system into IDM to keep the entity data in consistent state between the two systems.

Process of comparing and synchronizing accounts information in target system with IDM. Data flows into IDM from External Source.  
**Type of reconciliation -** Two Types  
**a) Trusted Source Reconciliation** – External Source is **authoritative** source (eg. HR), External Source drives creation, modification and deletion of user in IDM.  
**b) Target resource reconciliation** – external Source is non-authoritative source with which user is already provisioned.

**Identity lifecycle management :**

Similar to access lifecycle management. It refers to the entire set of processes and technologies for maintaining and updating digital identities. Identity lifecycle management includes identity synchronization, provisioning, de-provisioning, and the ongoing management of user attributes, credentials and entitlements.



**Identity synchronization**

 The process of ensuring that multiple identity stores—say, the result of an acquisition—contain consistent data for a given digital ID.

**Password reset**

 In this context, it's a feature of an ID management system that allows users to re-establish their own passwords, relieving the administrators of the job and cutting support calls. The reset application is usually accessed by the user through a browser. The application asks for a secret word or a set of questions to verify the user's identity.

**Authentication (AuthN) –**Authentication is the process of validating an identity, whether it be the identity of a user or, as in the Identity of Things, a device. The classic method of validation is the username/password combination.

**Authorization (AuthZ)** –Authorization is the process of determining if a user has the right to access a service or resource, or perform an action.

**Credential –**A credential is an item, such as an ID card, or a username/password combination, used by persons or entities to prove themselves.

**Data –**Any information stored by a computer.

**Group –**In identity management, a group allows the management of multiple entities (I.e. employees or customers) within a single category. Groups are used to define roles and simplify access control.

**Log Files –**Log files are files that record either events that occur in an operating system or software, or messages occurring on communication software. For example, when a failed login to an E-mail system occurs, a log file is created to record that event.

**Logging –**the act of keeping a log for an extended period of time.

**Multifactor Authentication –**Multifactor authentication adds an additional step (or factor) to the authentication process, typically by pairing something the user knows, such as username and password, with an action, or something the user has, such as an SMS message to their phone, an email, or a token.

**OAuth –**OAuth is an open authorization standard that allows applications to autonomously access resources on behalf of a user. iOS and Android, for example, use this kind of authorization to let users choose whether or not an app can have access to certain functions and parts of the phone.

**Offboarding –**The process by which a user is removed (with access revoked) from an organization’s IAM system.

**OpenID –**A standardized, open method of decentralized authentication.

**Onboarding –**The process of adding new users to an organization’s IAM system.

**One Time Password (OTP) –**A password that is valid for use one use or session.

**Privileged User –**A user possessing  specific security privileges and entitlements.

**Role –**An identity attribute that gives users automatic privileges when assigned. Roles make take the form of groups wherein all members of a group have the same set of privileges.

**Security Administrator –**A person responsible for maintaining a list of users, their identity attributes, their passwords, security privileges, or other authentification factors.

**Single Sign-On (SSO) –**In a single sign-on (SSO) service model users log onto a single platform which gives them automatic log-in access to multiple applications for a particular period of time.  When utilizing SSO systems users only need to present one set of credentials, rather than learning or remembering separate credentials for each application

**Federation**  - Also known as federated identity management, this is a technical implementation that enables identity information to be developed and shared among several entities and across trust domains.

**Identity provider (IdP)-**  A system that validates the identity of a user in a federated system. The service provider (or SP; see below) uses the IdP to get the identity of the current user.

**Service provider (SP)-**  A system that provides a generic service to the user in a federated system. To users, a service provider is the same thing as the application they are trying to use.

**Database**

**What is Data?**

In simple words data can be facts related to any object in consideration.

For example your name, age, height, weight, etc are some data related to you.

A picture , image , file , pdf etc can also be considered data.

**What is a database?**

Database Definition A database is a systematic collection of data. Databases support storage and  manipulation of data. Databases make data management easy. Let's discuss few examples.

An online telephone directory would definitely use database to store data pertaining to people, phone numbers, other contact details, etc.

Your electricity service provider is obviously using a database to manage billing , client related issues, to handle fault data, etc.

Let's also consider the facebook. It needs to store, manipulate and present data related to members, their friends, member activities, messages, advertisements and lot more.

We can provide countless number of examples for usage of databases .

**What is SQL?**

SQL Definition: Structured Query language (SQL) pronounced as "S-Q-L" or sometimes as "See-Quel"is actually the standard language for dealing with Relational Databases.

SQL programming can be effectively used to insert, search, update, delete database records.

That doesn't mean SQL cannot do things beyond that.

In fact it can do lot of things including, but not limited to ,optimizing and maintenance of databases.

Relational databases like MySQL Database, Oracle, Ms SQL server, Sybase, etc uses SQL ! How to use sql syntaxes?

 SQL syntaxes used in these databases are almost similar, except the fact that some are using few different syntaxes and even proprietary SQL syntaxes.

SQL Example

SELECT \* FROM Members WHERE Age > 30

**LDAP**

## **What is LDAP?**

It is a protocol separately designed for accessing the directory services which can store the information of the entire organization into a central repository.

[LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol) : (known as Lightweight Directory Access Protocol) is one of the core protocols that was developed for directory services. It was developed in the early 1990s to be a lightweight version of the X.500 directory services protocols that were in use at the time. The goal of directory services at the time was to create a lightweight protocol that would allow for authentication and authorization of users to servers and applications. LDAP works in a client server format where a client following the protocol will request authentication and/or authorization information.

LDAP in Action : The server side of LDAP is a database that has a flexible schema. Not only can LDAP store username and password information, but also can store a variety of attributes including address, telephone number, group, and a variety of others. Because LDAP is an open source protocol, IT organizations can extend the database and protocol to meet their needs. LDAP has been used inside of the popular open source directory service OpenLDAP. Further, many common applications leverage the LDAP protocol to authenticate and authorize user access.

Over time, LDAP became a core directory services protocol for the Internet.

[Microsoft’s Active Directory](https://jumpcloud.com/blog/category/active-directory/) – the most widely implemented directory service in the world – built many of its underpinnings on LDAP while also extending the concept of directory services with many proprietary extensions. LDAP has most often been used for more technical situations and organizations. Because of it’s Unix and open source nature, LDAP often lends itself well for environments that are Linux heavy and leverage technical applications.

There are a handful of overarching concepts with LDAP as with most [directory services](https://jumpcloud.com/daas-product/index). The core of a directory service is the ability to add, delete, and modify records. The process of authenticating in LDAP is referred to as binding. Binding can be encapsulated within a secure, encrypted session. LDAP also provides for the ability to search the database to compare attributes – a common task done around groups. Using these basic operators within an LDAP infrastructure, IT admins can accomplish a significant amount of their directory service’s needs.

Directory Servers: (Basic Concept of LDAP):

Directory: It is a difficult structure database that stores data (user entries), through which we can search easily for special purpose.

LDAP: (Light Weight Directory Access Protocol).

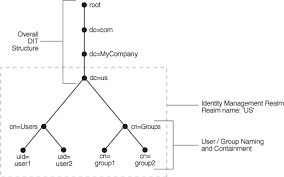
It is internet engineering task force (IETF) standard, an LDAP directory is organized in the form simple hierarchical tree known as directory information tree.

DAP: It is a protocol to access database directory.

DN (Distinguished Names):

Each user entry in an online directory is uniquely identified by distinguished name. The DN tells you exactly where the user entry resides in the directory hierarchy. This hierarchy is represented by directory information tree (DIT).

Directory Information Tree –



O = Organization

C = Country

OU = Organization Unity

CN = Common Name

SN = Sur Name

DC = Direct Component

The DN for this “smith” entry is (bottom to up). Cn=smith, Ou=server development, c=IND, O=acme.

Attributes: Each attribute consists of attribute type, attribute values.

The attribute type is the kind of information that the attribute contains.

Example: Job Title.

The attribute value is the particular occurrence of information appearing in that entry.

Example: Manager.

Attributes contain 2 kinds of information.

Application Attribute.

System Configuration Attribute.

Application Attribute: This information is maintained and retrieved by application clients or directly clients and is important to the operation of the directory or application.

Example: Telephone Number, Home address, Email Id.

System Configuration Attributes: This information pertains to the operation of the directly or application itself. Some operational information is specified by the application or directly to control the server.

Example: Time stamp for creation, Time stamp for modification, Name of the user who create or modifies user entry.

**What is LDAP Authentication?**

There are two options for LDAP authentication in LDAP v3 – simple and SASL (Simple Authentication and Security Layer).

Simple authentication allows for three possible authentication mechanisms:

Anonymous authentication: Grants client anonymous status to LDAP.

Unauthenticated authentication: For logging purposes only, should not grant access to a client.

Name/Password authentication: Grants access to the server based on the credentials supplied – simple user/pass authentication is not secure and is not suitable for authentication without confidentiality protection.

SASL authentication binds the LDAP server to another authentication mechanism, like Kerberos.

The LDAP server uses the LDAP protocol to send an LDAP message to the other authorization service. That initiates a series of challenge response messages that result in either a successful authentication or a failure to authenticate.

It’s important to note that LDAP passes all of those messages in clear text by default, so anyone with a network sniffer can read the packets. You need to add TLS encryption or similar to keep your usernames and passwords safe.

**What is an LDAP Query?**

An LDAP query is a command that asks a directory service for some information. For instance, if you’d like to see which groups a particular user is a part of, you’d submit a query that looks like this:

(&(objectClass=user)(sAMAccountName=yourUserName)

(memberof=CN=YourGroup,OU=Users,DC=YourDomain,DC=com))

Beautiful syntax, huh? Not quite as simple as typing a web address into your browser. Feels like LISP.

Luckily, in most cases, you won’t need to write LDAP queries. To maintain your sanity, you’ll perform all your directory services tasks through a point-and-click management interface like Varonis DatAdvantage or perhaps using a command line shell like PowerShell that abstracts away the details of the raw LDAP protocol.

TL;DR: LDAP is a protocol, and Active Directory is a server. LDAP authenticates Active Directory – it’s a set of guidelines to send and receive information (like usernames and passwords) to Active Directory. Want to learn more? Get a 1:1 AD demo and learn how Varonis helps protect your Active Directory environment.

**Common LDAP Attributes:**

|  |  |  |
| --- | --- | --- |
| Attribute Type | Attribute String | Description |
| Common Name | Cn | Common Name of the user entry. |
| Domain Component | Dc | The DN of the component in a domain name system (DNS). Example: dc=uk, dc=acme, dc=com |
| JPEG Photo | Jpeg Photo | Photographic image in JPEG format, this is store in binary format. |
| Organization | O | Name of the organization. |
| Organization Unit Name | Ou | Name of the unit within an organization. |
| Owner | Owner | Distinguish name of the person who owns the user entry. |
| Sur name | Sn | Last name of the person. |
| Telephone Number | Telephone Number | Telephone Number of user. |

Attribute Syntax: Attribute syntax is the format of the data that can be loaded into each attribute.

Example: Telephone Number attribute syntax contain string of numbers containing spaces and hyphens.

**Application Server**

An **application server** is a [software framework](https://en.wikipedia.org/wiki/Software_framework) that provides both facilities to create [web applications](https://en.wikipedia.org/wiki/Web_applications) and a [server](https://en.wikipedia.org/wiki/Server_(computing)) environment to run them.

Used for “Server side programming”

For Example :

WebLogic,

WebSphere,

*Tomcat,*

Jboss