Active Directory

→ Active directory is essentially a database.

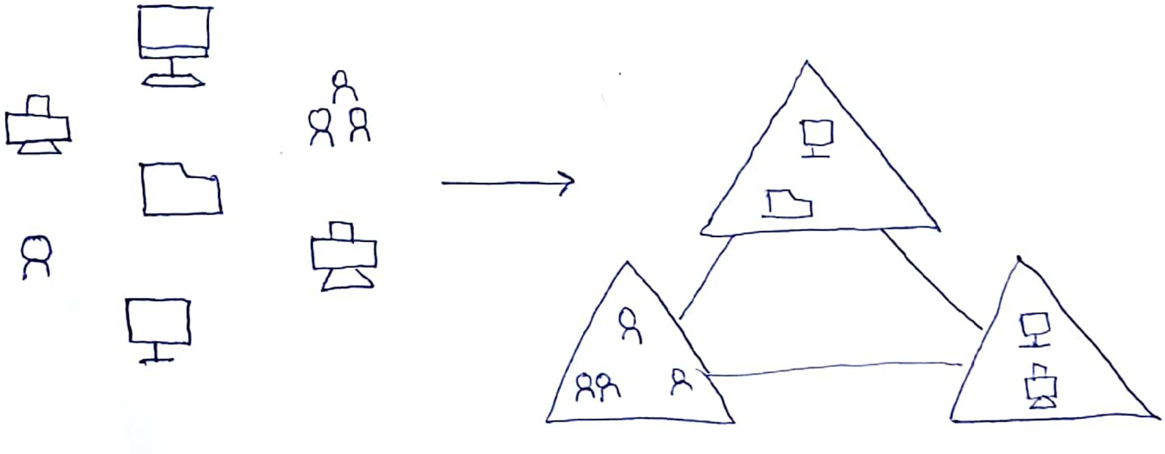
→ It is a database of user name and password (not just user name & password, but user details, computers and printers)

→ It is a directory which store information about all the objects - users, computers, resources like printers, shared file/folders - in an organization's network.

→ Based on this information it provides access and permission to objects or on objects.

→ Services like Email uses Active directory

→ Active directory stores group policy also along with objects.

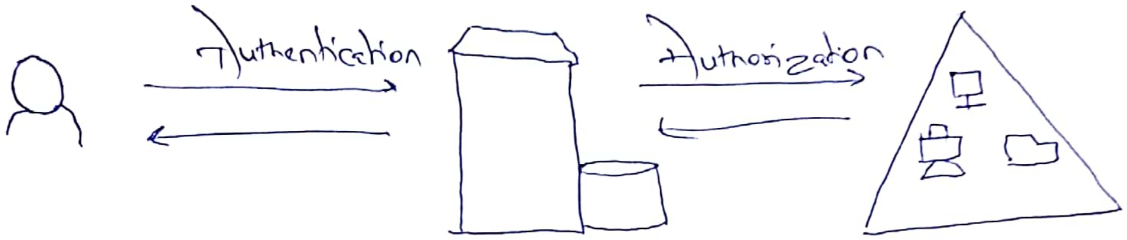


→ AD arranges all the Network's users, Computers and other objects into logical and hierarchical groupings.

⇒ Apart from storing object information, main function of AD is to authenticate & authorize users, Computer etc in a n/w

[Authentication - If username & password are correct

Authorization - On which all resources does the user has access]



AD

→ Object is a physical entity in a network. It can be described by a subset of attributes.

→ Objects in AD can be:

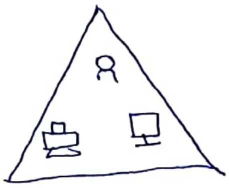
- User
- Computer
- Printer
- Group
- Shared folder
- Subnet
- Forest
- Domain
- OU- Organizational unit
- Site

→ Objects are explained by their attributes like Name, location, Department, phone no., etc.

Example.

Name: John
Location: hm/users
Department : HR

→ There is another type of object called container object which can contain other domains



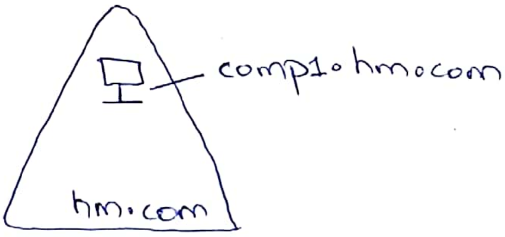
Ex. Domain, OU etc.

→ Objects which cannot contain otheo objects are called leaf objects. Ex. User, computer and printer.

Domain :

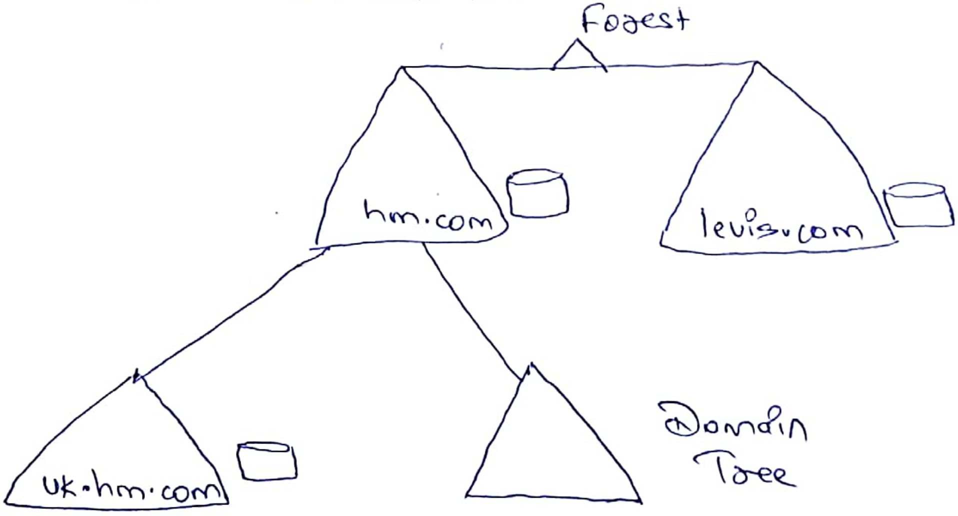
→ A domain is defined as a logical group of network objects (Computer, users, devices) that shares the same active Directory database

→ These objects also share the same namespace



Ex. for domain hm.com, computer is comp1.hm.com

→ If company has smaller subsidiaries in other location they can be made child domains. Ex.



→ Ex. if H&M buys levis, then levis can be linked to HM parent domain.

→ Forest is the highest level of classification.
It is the highest level of security boundary.

→ Forest can also be termed as complete Active Directory Instance

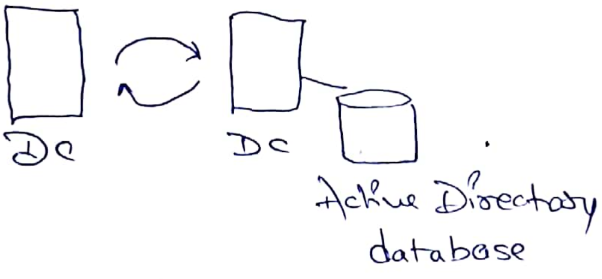
→ Forest contains objects like Domains, users, computers, Printers and other network resources.

→ Information and data exchange can only happen between objects inside a forest.
(Ex. Only employees within an organisation)

→ To communicate with objects in other forests we need to create forest level trust.

→ Forest can contain 1 or more domains or combination of domains called domain trees

⇒ Domain Controller run active directory domain services
It holds a copy of the active directory database. It replicates the changes with other domain controllers.

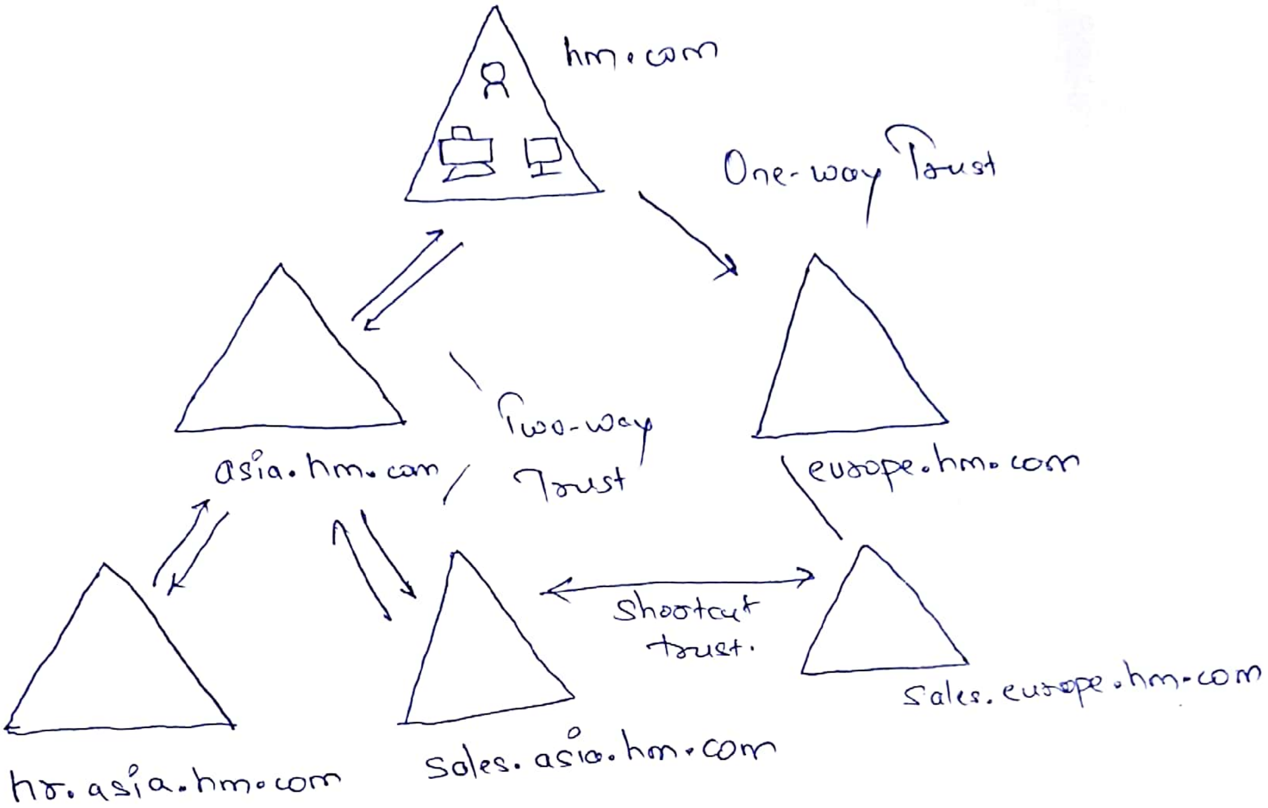


→ Domain controller is the domain's supreme authority

→ It is responsible for all authentications, authorizations, additions, deletions, edits & modifications inside a domain

→ If a user has access to a domain, he can logon from anywhere & any computer in that domain.

Domain Tree

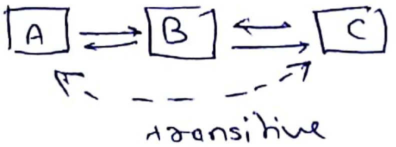


→ Domain Tree is a parent-child tree structure or Nested domains.

→ To establish communication b/w 2 domains, we create trust. There are various level of trusts

→ At root is the parent or root domain and beneath it has child domains. further a child domain can have more child domains beneath it.

→ There is a transitive trust relationship in a domain tree i.e if domain A has trust relation with domain B and domain B has trust relation with domain C, then there is a transitive trust relation b/w domain A and C.

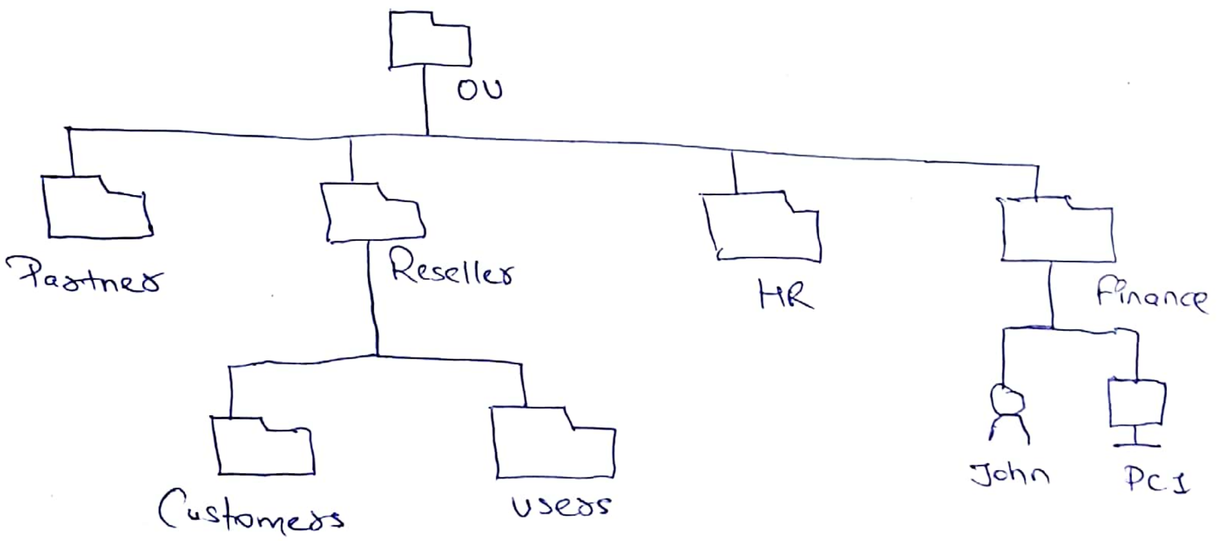


→ A domain is a very large objects and keeping objects like users, computers etc in it directly would be unorganized.
Hence domain needs another structuring or grouping of objects within it which can be done using OU (organizational unit)

→ Objects within a domain can be grouped into OU.

→ OU can provide hierarchy to a domain, ease its administration. and management.

→ OU can be used to denote a specific department, location, Team, function etc.

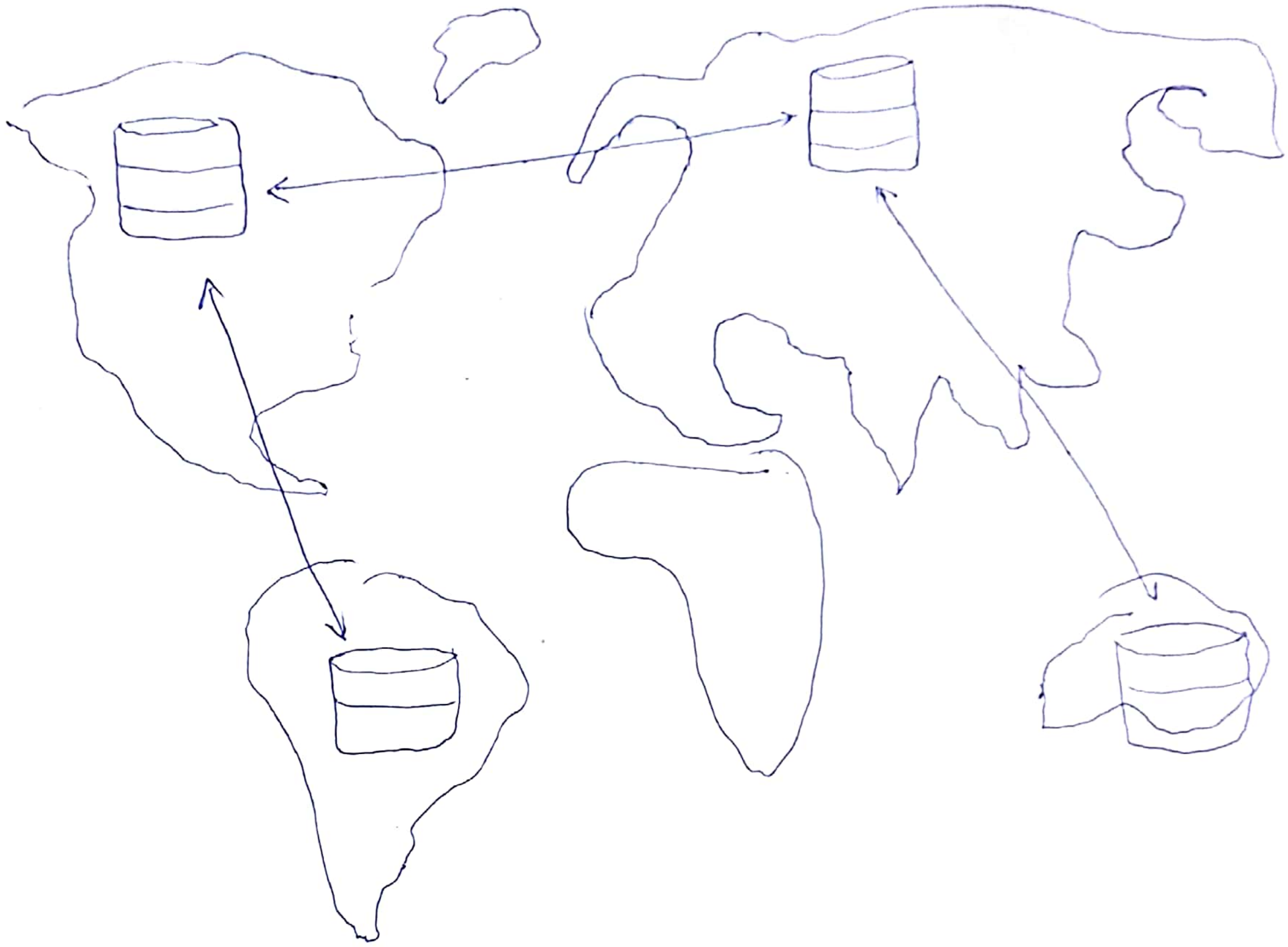


→ OUs are unique inside a domain.

→ OUs can contain other OUs inside them as well. Nested OUs have Parent-child relationship

→ All OUs inside a domain are connected

Active Directory Database



→ Active Directory is essentially a distributed database

→ Databases are stored on domain controllers and changes made on these domain controllers are replicated to other domain controllers.

→ Database is stored in NTDS. DIT file.

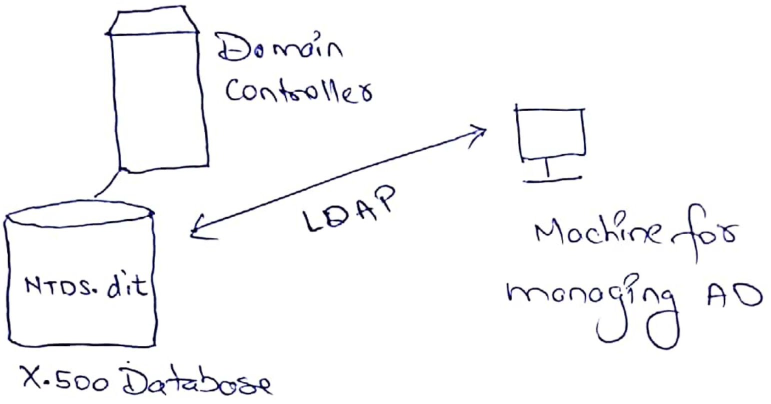
(Originally active directory was called NT directory, thats why the name NTDS)

→ NTDS. DIT database file is based on X.500 standard

(X.500 is a technique of Hierarchical distribution (organization) of entries stored/distributed across one or more servers)

LDAP - Lightweight Directory Access Protocol

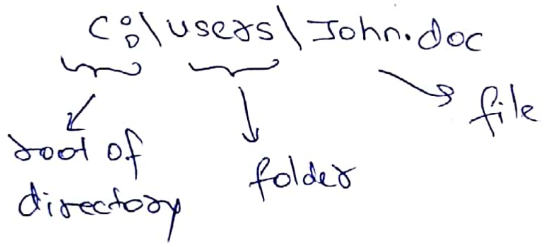
→ This is the protocol which interacts with the database to validate credentials and locate users, files, devices etc. in a n/w.



→ LDAP is an application protocol for querying and modifying items in directory services.

→ So LDAP is simply a way of accessing AD database.

→ To understand the syntax of LDAP, lets look at an example of how do we access a file on a computer



LDAP Syntax - CN - Canonical name or container name

CN is similar to file name. In AD CN represent the object.

Ex. If user is John and we want to access it

CN = John

Now since there can be many users by the name 'John' so we need to tell LDAP, where this object is located

In AD we use OU for storing objects. It is similar to folder in windows computer.

CN=John, OU=Users

→ Now we have object and its OU, we also need to mention its domain i.e where this OU's located.

CN= John, OU= users, DC= hm, DC= com

↓ canonical or container name

↘ organizational unit

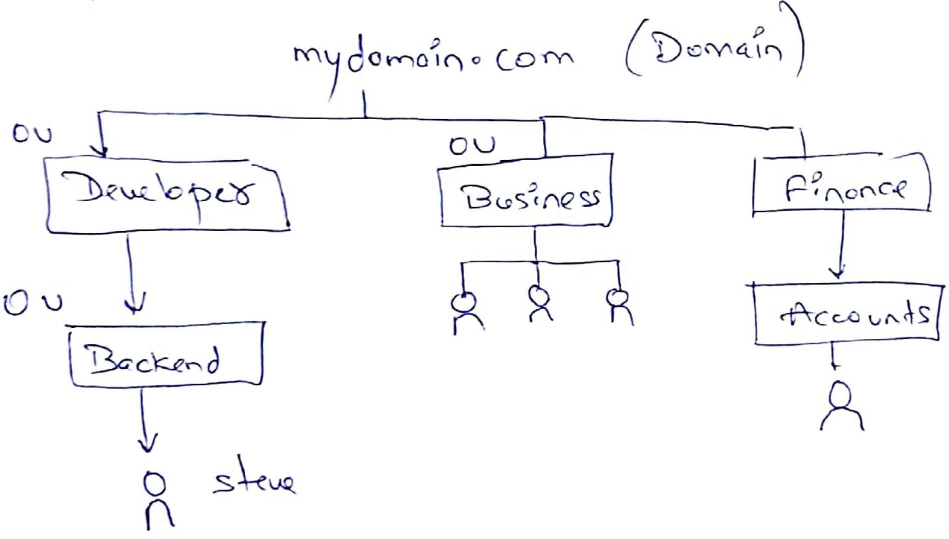
Domain component

Distinguished Name

→ Every object in AD has a distinguished name.

→ LDAP identifies every object from its distinguished Name

Another Example



To access 'Steve', LDAP syntax would be

CN= John, OU= Backend, OU= Developer, DC= mydomain, DC= com

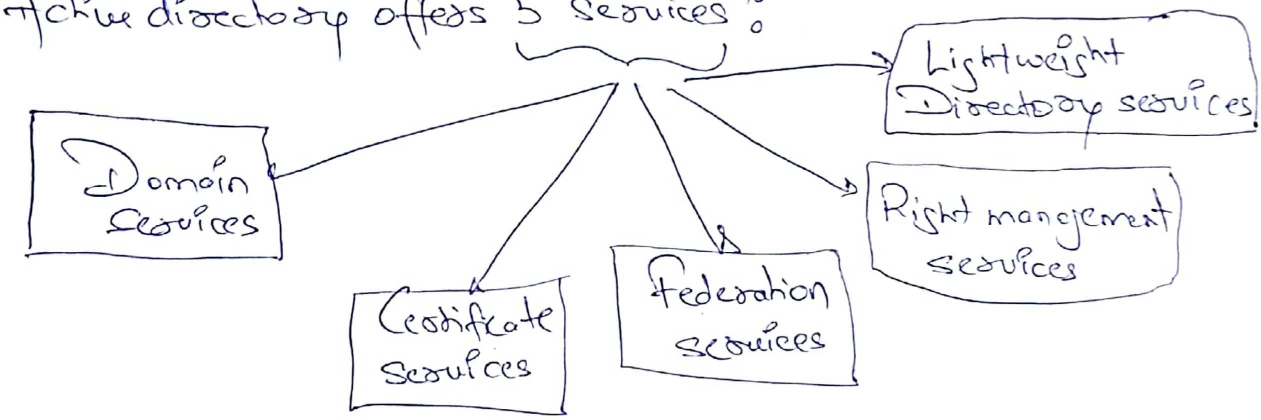
Active Directory Services

→ Whatever we've discussed till now is about Active directory domain services which is one of the service of part of Active directory.

→ Active directory is actually a collection of services ~~of~~ or suite of services (server roles and features) used to manage identity and access for and to a resource on a network.

→ Thus active directory focuses on access & Identity management

→ Active directory offers 5 services:



⇒ We've already discussed about the domain services.

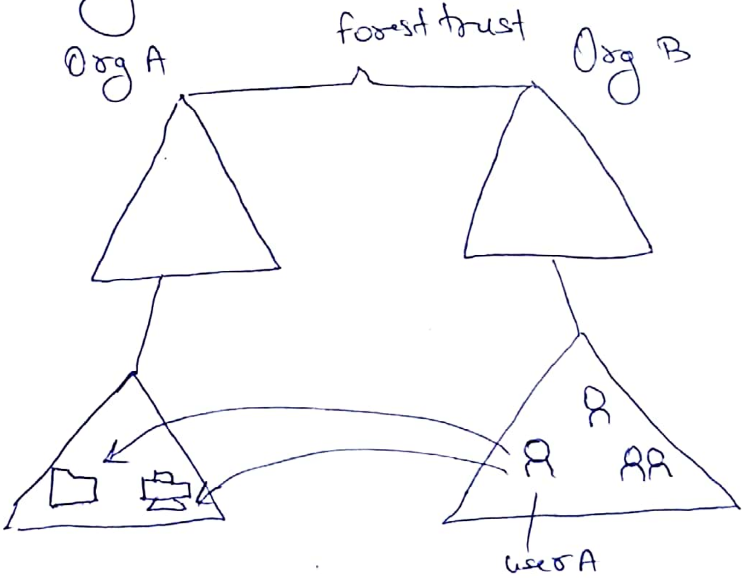
Active Directory Federation Services

→ AD FS is a service of AD and a feature of windows server that helps user to access applications and systems outside the company n/w and firewall using their own credential and without signing in again (which is also called SSO - single sign On)

→ AD FS authenticate users on third-party systems, such as another company's extranet or services hosted by a cloud provider.

→ AD FS requires a federated trust relationship between two organizations or entities.

→ Previously we learnt that to enable communication b/w two organizations we were making the forest trust. But forest trust allows full access by users of one organization to resources of another organization



[Here user A has access on both the printer as well as shared folder of organisation A because of forest trust. If company want to restrict user A access to printer and only allow access on folder That is not possible]

→ Hence we use federation services.

It allows comprehensive forest trust where organizations get to retain control over who can access resources.

Also it enables organizations to retain control of their users and group accounts

⇒ To summarise ADFS is an identity access solution that provides client computers with seamless SSO access to protected Internet-facing applications or services, even when user account and application is located in a completely diff n/w or organization

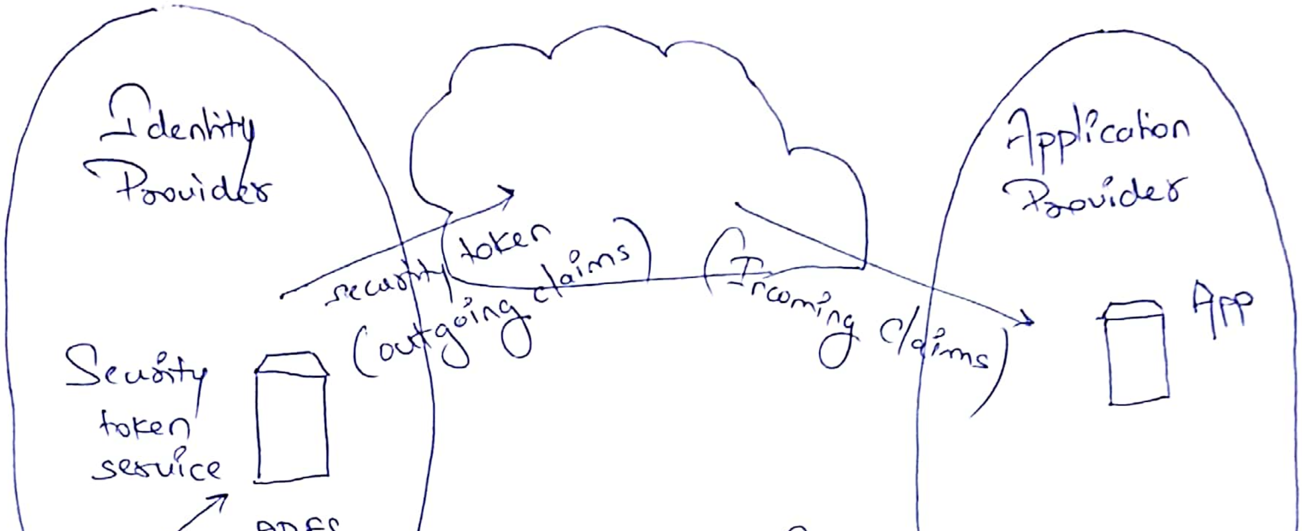
Ex. Many websites give option to login with Google plus, facebook account or LinkedIn account.

Because these companies have federated trust b/w them.

⇒ Core technology used in ADFS is 'Claim-based Identity'

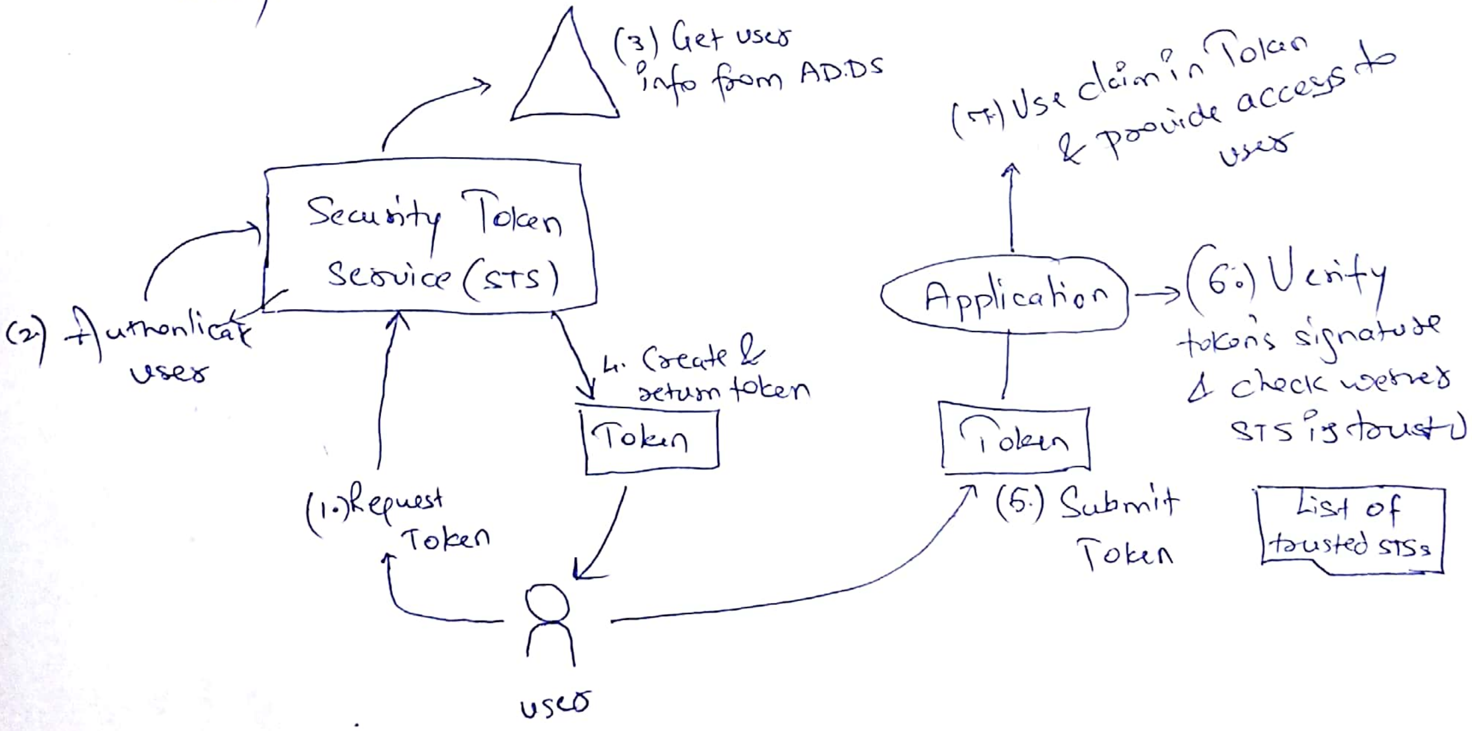
Claim based authentication requires SAML (Security Assertion Mark-up language) Tokens. These tokens are issued by the ADFS server.

[claim based identity simply means combining authentication and authorization]



claims provide info about users who the identity provider authenticates and which the application provider accepts.

These applications must be claim-aware (ex. office 365, salesforce etc)


(3) Get user info from AD.DS
(7) Use claim in Token & provide access to user
Security Token Service (STS)
(2) Authenticate user
4. Create & return token
Token
Application
(6.) Verify token's signature & check wether STS is trusted
Token
(1.) Request Token
(5.) Submit Token
List of trusted STSs
user