System and Network Administration

A · P · U

ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Core Services:

LDAP

Directory Services

A centralized, structured repository of configuration, authentication and other related information.

A network service that stores information about **objects** in a particular network.

- Objects can include:
  - Users
  - Groups
  - Servers
  - Clients
  - Printers
  - Network services

- Structure is based on a hierarchy of objects.
- Starts with root and branches out to other objects.

Directory Services

- A centralised, structured repository of configuration, authentication and other related information.
- A system optimised for high performance query capabilities
  - Access Control: per record
  - Optimised for read-only use (not updated during use)
  - It doesn’t have Relationships (PK <- fk)
  - It isn’t Transactional
  - It has poor modification performance
    - **SQL** allows complex update and query functions at the cost of program size and application complexity.
    - Directories use a simplified and optimized access protocol that can be used in slim and relatively simple applications.

The Role of Directory Services

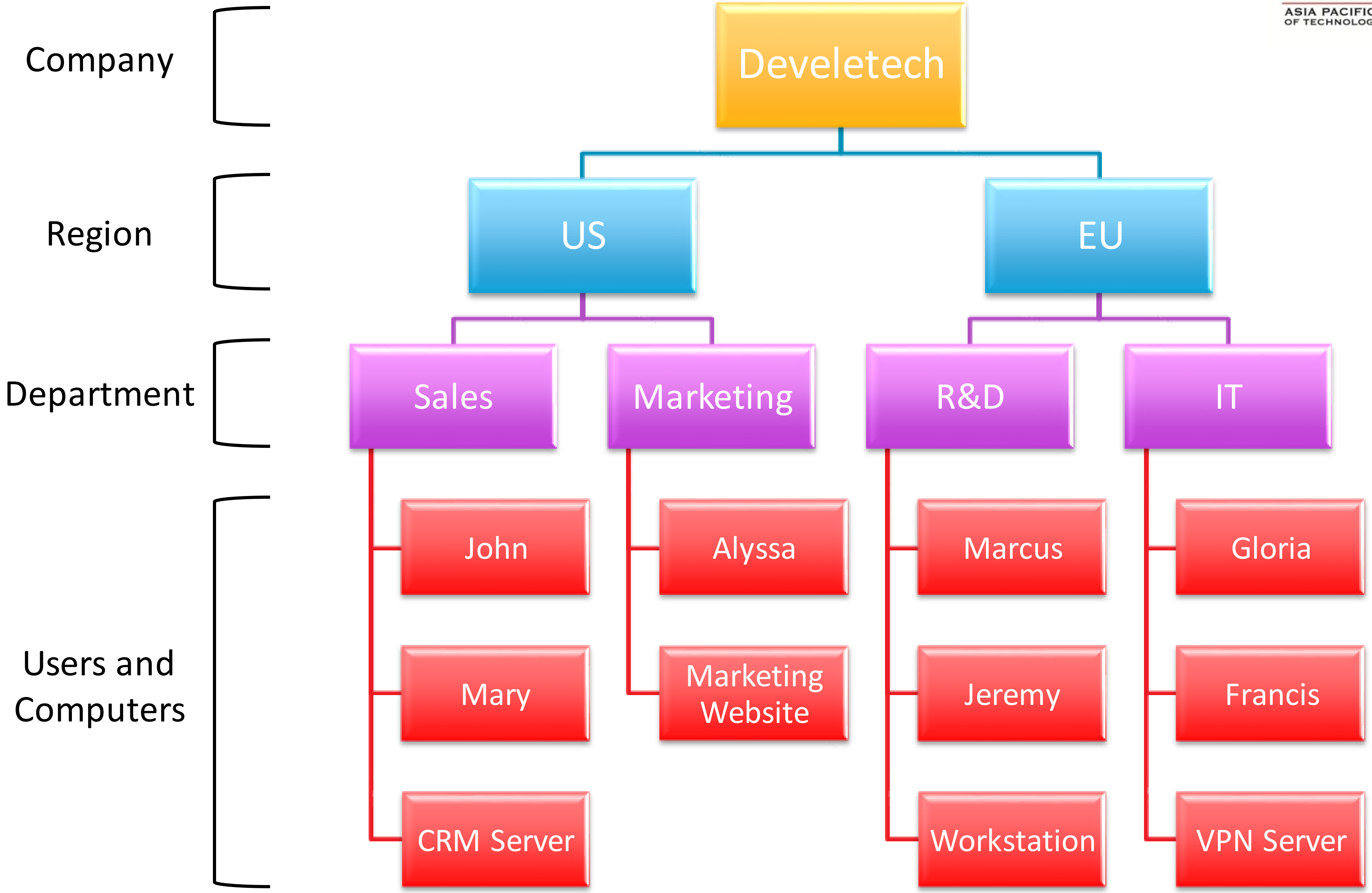
- Application settings: Allows applications to query the directory for configuration information
  - Physical location of application components
  - Version information for application components
  - Application’s object definitions
- Personal data: Allows people to find information about others
  - Users, telephone numbers, email, hosts, IP address
  - Merger of authentication sources

For example, a guard views an employee ID and types the name into a web page. The page calls a CGI program which does an LDAP query, and retrieves the employees picture file. The picture is displayed so the guard can verify the employees identity.

Objects, Attributes, and the LDAP Directory

- Real-world objects such as users and computers are represented as objects in the LDAP Directory.
- The LDAP Directory supports numerous object types
- Each object and object type is represented by a unique global identifier.
- Each object has a set of attributes that best describe it. For example, consider a user object. Each user can have attributes such as Name, Address, and Telephone number.

Directory Hierarchy

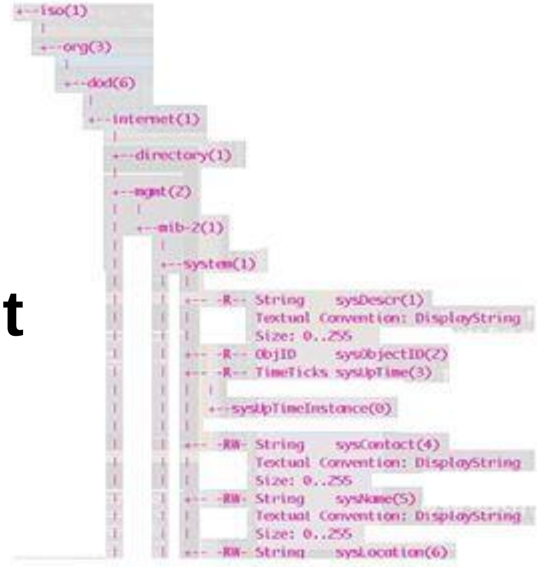


Hierarchical Directory

- Well suited to distributed environment: Directory tree may be partitioned into sub-trees with no overlap.
  - allows delegation of parts to separate hosts.
  - Cooperating groups can then manage their own data locally and share with others.
- May allow Availability and Redundancy through replication of data and service

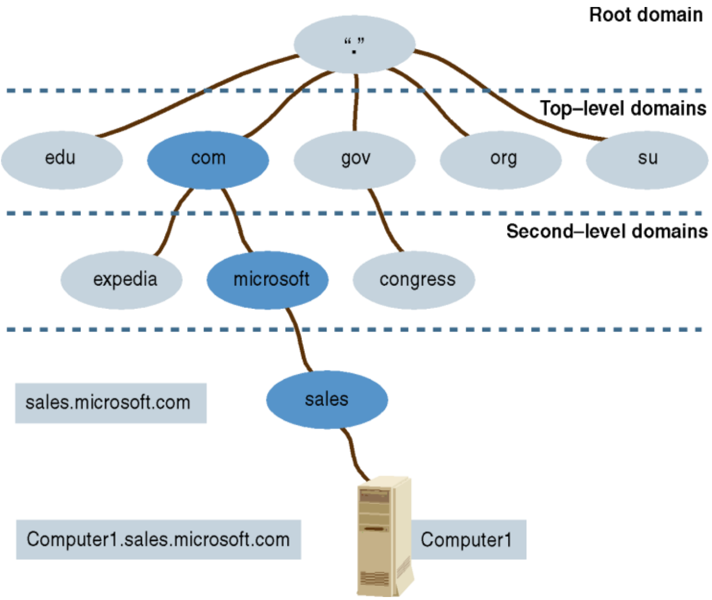
Network Directory Services

SNMP for system management information



```
+--iso(1)
   |
   +--org(3)
      |
      +--dod(6)
         |
         +--internet(1)
            |
            +--directory(1)
            |
            +--mgmt(2)
            |  |
            |  +--mib-2(1)
            |     |
            |     +--system(1)
            |     |  |
            |     |  +-- -R-- String    sysDescr(1)
            |     |  |        Textual Convention: DisplayString
            |     |  |        Size: 0..255
            |     |  +-- -R-- ObjID     sysObjectID(2)
            |     |  +-- -R-- TimeTicks sysUpTime(3)
            |     |  |  |
            |     |  |  +--sysUpTimeInstance(0)
            |     |  |
            |     |  +-- -RW- String    sysContact(4)
            |     |  |        Textual Convention: DisplayString
            |     |  |        Size: 0..255
            |     |  +-- -RW- String    sysName(5)
            |     |  |        Textual Convention: DisplayString
            |     |  |        Size: 0..255
            |     |  +-- -RW- String    sysLocation(6)
```

DNS



LDAP for organisational information

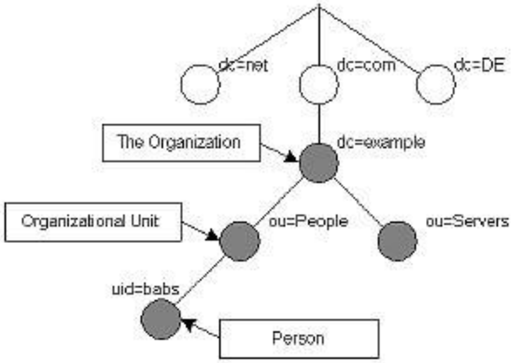


Figure 1.2: LDAP directory tree (Internet naming)

Common LDAP Directory Services

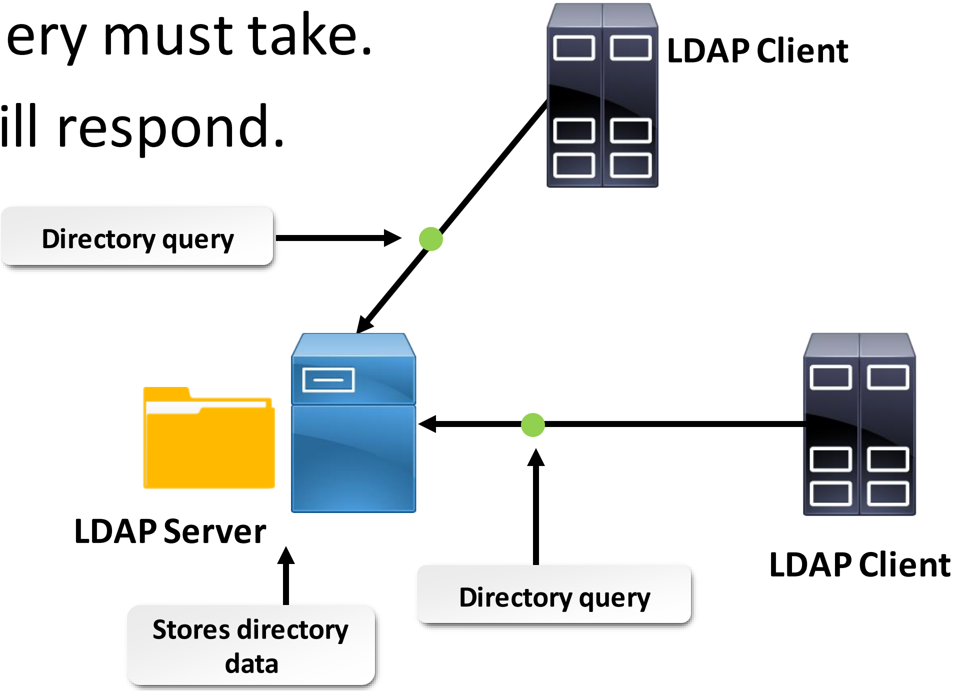
Directory Service	Description
OpenLDAP	• Open source cross-platform LDAP implementation. • Included in many Linux distros.
Microsoft Active Directory	• Compatible with simplified version of standard LDAP • Holds network object info for one or more **domains**.
Open Directory	• Apple's custom implementation of OpenLDAP for macOS. • Some compatibility with Active Directory.
Oracle Directory Server Enterprise Edition (ODSEE)	• Marketed toward large installations that require reliable scaling. • Formerly known as Sun Java System Directory Server.
OpenDJ	• Open source cross-platform directory service written in Java. • Based on Sun's OpenDS service.

Q&A
dreamstime.com

?

Lightweight Directory Access Protocol

- LDAP clients authenticate to LDAP service.
- **Schema** defines:
  - Tasks clients can and cannot perform while accessing directory.
  - The form a directory query must take.
  - How directory server will respond.
- **Schema** is extensible.



LDAP: Lightweight Directory Access Protocol

- Described in RFC 2251-2256, 2829-2830
  - Unencrypted access TCP port 389
  - LDAP-over-SSL LDAPS TCP port 686
- Your directory can contain pretty much anything you want to put in it.
  - Information describing users, applications, files, printers, and other resources accessible from a network
- The most typical use is email directories: all the usual mail programs recognize LDAP directories
- Easy lookups from scripts:
  - ldapsearch
  - Java/Python libraries
  - Web based tools: phpLDAPadmin

X.400 and X.500

- 1978: International Standards Organisation (ISO) and International Telecommunications Union (ITU)* began working on standards for network protocols, electronic mail. and directory services
  - ISO-OSI model was defined, with protocols for the lower layers. Not too popular compared to TCP/IP
- 1984: X.400 email – was popular for a while, not anymore.
- 1988: X.500 directory services – considered too cumbersome, but led to Simple Network Management Protocol (SNMP) and Lightweight Directory Access Protocol (LDAP)

** CCITT was the ITU standards committee*

X.500 Directory Terms – 1

DIT: Directory Information Tree

- Data is represented in the directory as a hierarchy of **objects**, each of which is called an **entry**. The DIT is the hierarchy of objects that make up the local directory structure.
- The top of the tree is commonly called the **root** or the **base** entry

Entry: An object stored in a directory.

- Each entry has one parent entry and zero or more child entries
- Entries are composed of a collection of **attributes** that contain information about the object
- The information is represented as a **value** for the attribute.
- Attributes and their characteristics are defined in a **schema**

X.500 Directory Terms – 2

RDN: Relative Distinguished Name

- The **OID** of an entry that is unique at its level in the hierarchy.
- DNS equivalent is a partial domain name like .com or .edu.my

DN: Distinguished Name.

- Uniquely defines an object and who is responsible for its definition
- Comprised of a series of **RDNs** that uniquely describe the complete path from the **DIT** root to the entry
- DNS equivalent is a FQDN (fully qualified domain name)

OID: Object Identifier

- **DN** represented as a string of numbers delimited by decimals, like 1.3.6.1.2.1.1 which is equivalent to

  iso(1) org(3) dod(6) internet(1) mgmt(2) mib(1) system(1)

X.500 Directory Terms – 3

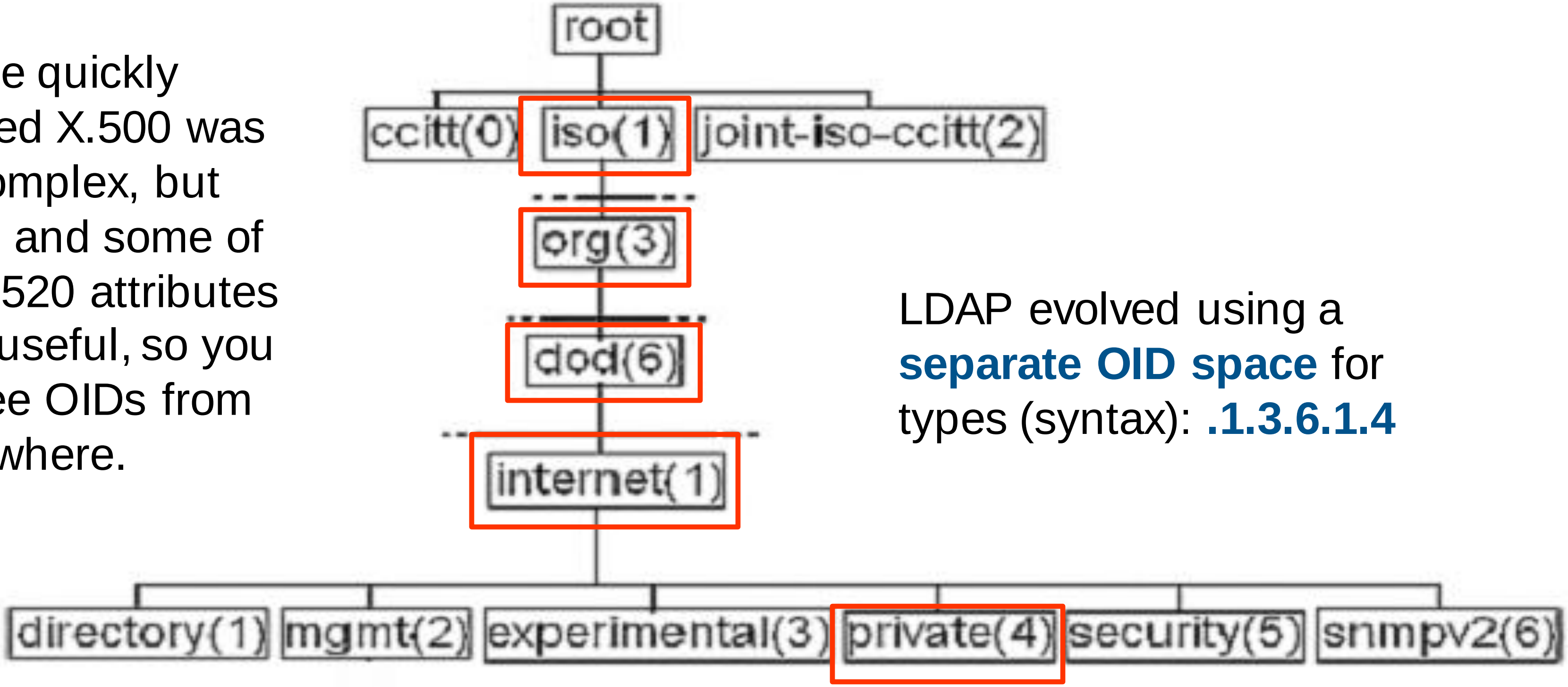
Schema: Defines objects that can be stored in the DIT

- Defines where in the DIT structure (hierarchy) objects may appear.
- Lists the attributes of each object and whether these attributes are required or optional
- Every attribute used must be defined in a schema that is known to the directory server
- The data in an entry is contained in attribute-value pairs.
- Type of an attribute is commonly used to refer to the ASN.1 SYNTAX of the attribute.
- The ASN.1 Syntax specifies what kind of values can be stored in attributes.

LDAP: Lightweight Directory Access Protocol

When X.500 standard attribute types were defined in X.520, the plan was to use **iso(1) org(3) dod(6) internet(1) directory(1)** or **.1.3.6.1.1** for general information and **.1.3.6.1.5** for X.509 certificates

People quickly decided X.500 was too complex, but X.509 and some of the X.520 attributes were useful, so you will see OIDs from everywhere.



LDAP evolved using a **separate OID space** for types (syntax): **.1.3.6.1.4**

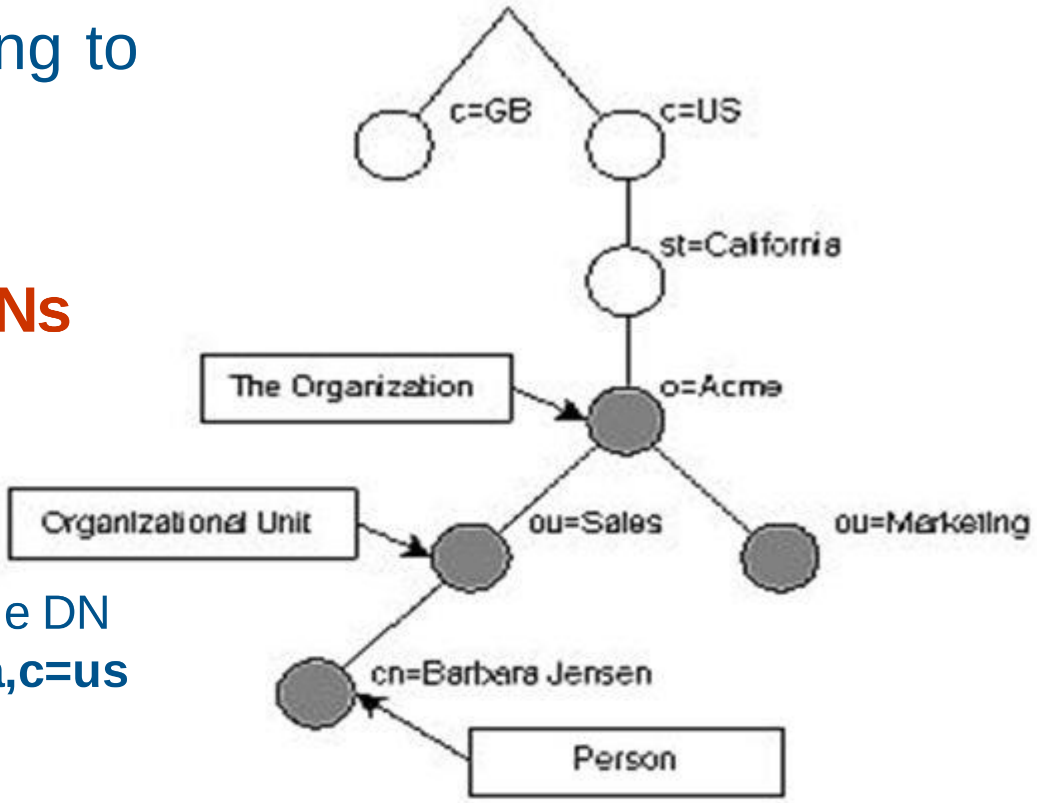
SYNTAX 1.3.6.1.4.1.1466.115.121.1

LDAP Naming Model

Entries are named according to their position in the DIT

A RDN is a name=value

A **DN** is a sequence of **RDNs** separated by commas

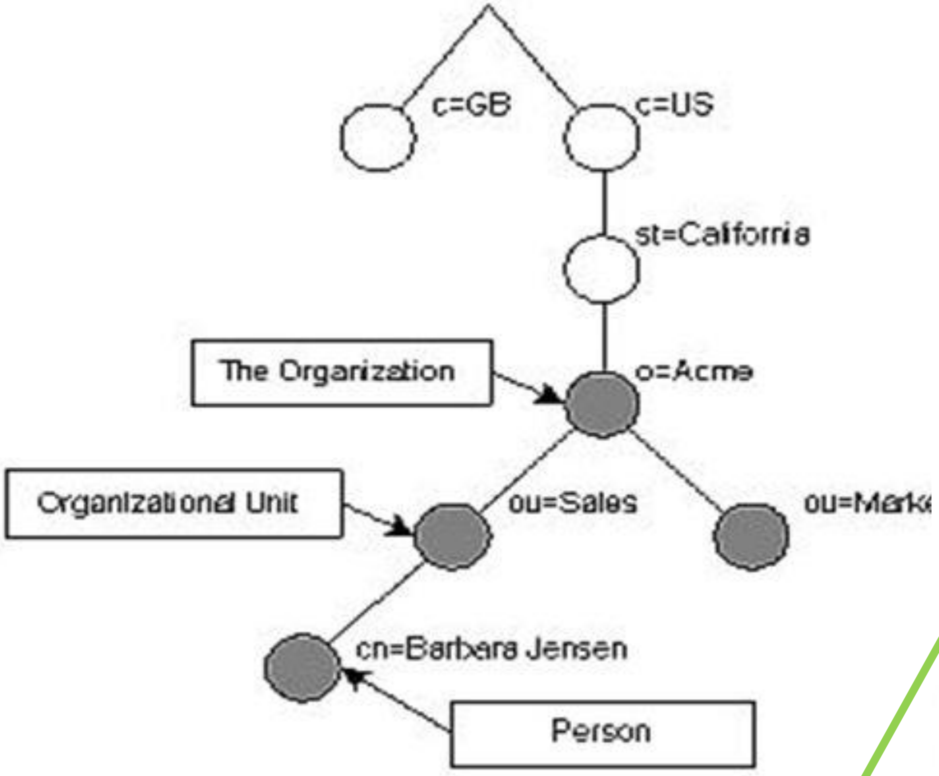


The Organizational Unit **sales** has the DN **ou=sales,o=Acme,st=California,c=us**

The entry at the bottom for Person has the DN of

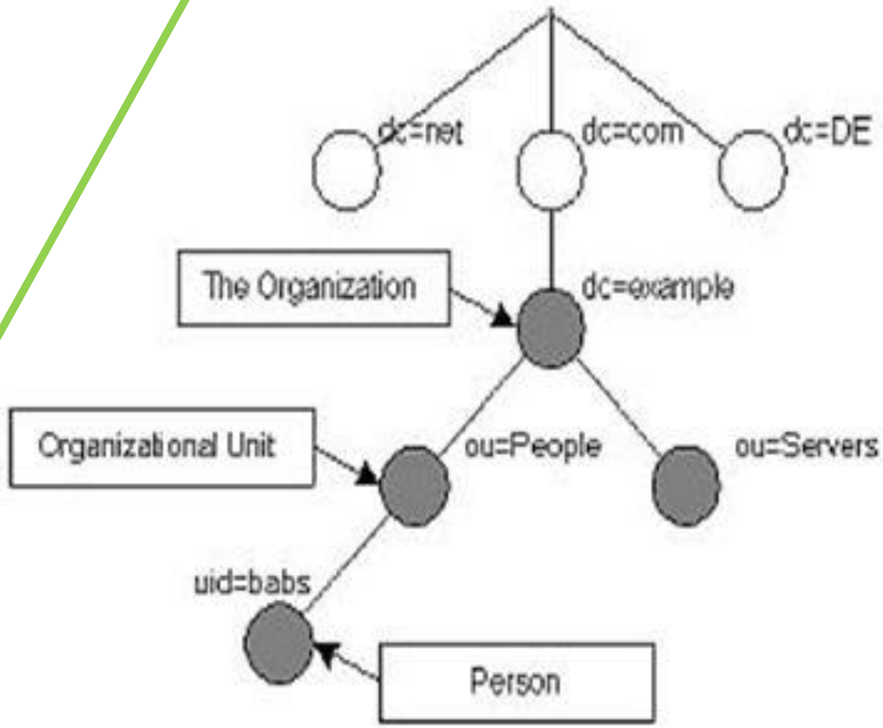
cn=Barbara Jensen,ou=sales,o=Acme,st=California,c=us

Typical LDAP DITs



Traditional

Internet Domain



LDAP Functional Model

General interaction between an LDAP client and server:

1. Binding: establishes a session

    The client specifies the host name or IP address and port number where the LDAP server is listening.

2. Authentication

- Authenticated User: client supplies a **DN** identifying itself along with a simple **clear-text password**..
- Anonymous User: without username and password.

While **plaintext authentication with TLS to encrypt the session is quite common**, use of strong authentication is encouraged:

- Simple Authentication and Security Layer (SASL) makes several different authentication methods available

LDAP Functional Model

General interaction between an LDAP client and server:

3. Perform Operations:
   - Search for entries meeting user-specified criteria
   - Add / Update / Delete an entry
   - Modify the DN or RDN of an entry (move)

4. Unbind
   - Close the session with the server

LDAP Queries

The most common operation is search.

To perform a search, the following parameters **must** be specified:

1. **Base:** A DN that defines the starting point, called the base object, of the search. The base object is a node within the DIT.
2. **Search Filter:** Specifies the criteria an entry must match to be returned from a search.
   - The search filter is a Boolean combination of attribute value assertions.

LDAP Queries

The following parameters **may** be specified:

3. **Scope:** Specifies how deep within the DIT to search from the base object.
   - **baseObject**: Only the base object is examined
   - **singleLevel**: Only the immediate children of the base object are examined, but not the base object itself
   - **wholeSubtree**: The base object and all of its descendants are examined
4. **Attributes to Return:** from entries that match the search criteria.
5. **Limit:** restricts the number of entries returned from the search.

OpenLDAP Database Tools

Slapadd is used to add entries specified in LDIF to an OpenLDAP database when the slapd daemon is running. Note that it does not perform user and system schema checks – in particular, it does not verify that superior entries exist before adding an entry.

ldapadd is used to add entries specified in LDIF to an openLDAP database when the slapd daemon ***is not*** running. **It does verification and schema checks.**

The -c option to continue in case of a syntax error is highly recommended

To initialise the LDAP database, these tools can be easier to use than web-based tools

OpenLDAP Database Tools

ldapsearch opens a connection to an LDAP server, binds, and performs a search using specified parameters.

- Useful for debugging and in special cases
- Generally, other tools are better
  - phpLDAPadmin

ldapmodify: allows you to batch changes in an LDIF file

- The LDIF format can have a changetype: field in the record

  add, modify, delete, modrdn
- Every change record requires a unique DN
- Useful in special cases

Common LDAP Directory Services

Directory Service	Description
OpenLDAP	• Open source cross-platform LDAP implementation. • Included in many Linux distros.
Microsoft Active Directory	• Compatible with simplified version of standard LDAP • Holds network object info for one or more **domains**.
Open Directory	• Apple's custom implementation of OpenLDAP for macOS. • Some compatibility with Active Directory.
Oracle Directory Server Enterprise Edition (ODSEE)	• Marketed toward large installations that require reliable scaling. • Formerly known as Sun Java System Directory Server.
OpenDJ	• Open source cross-platform directory service written in Java. • Based on Sun's OpenDS service.

What’s in slapd.conf

OpenLDAP configuration is done via a file called:

/etc/openldap/slapd.conf

This is where you specify:

- Authentication and TLS
- Schema and ObjectClass definitions

```
include  /etc/openldap/schema/core.schema
include  /etc/openldap/schema/cosine.schema
include  /etc/openldap/schema/inetorgperson.schema
```

- Backend Database options, including indexes and default search base

I’ve forgotten the ‘admin’ password

oops!

1. Change the password in /etc/openldap/slapd.conf

   rootpw <admin password>

2. Restart slapd

   /etc/rc.d/rc.slapd restart

Q&A

AAA is an architectural framework for configuring a set of three independent security functions consistently

- **Authentication: the method of identifying users,** including login and password dialog, challenge and response, messaging support, and, depending on the security protocol selected, possibly encryption.
- **Authorization: the method for access control,** including one-time authorization or authorization for each service, per-user account list and profile, user groups, and protocols
- **Accounting: the method for collecting and sending information used for billing, auditing, and reporting,** such as user identities, start and stop times, executed commands, number of packets, and number of bytes.

Guidelines for Placing ACS in the Network 17 Jan 2006
www.cisco.com/en/US/products/sw/secursw/ps2086/products_white_paper09186a0080092567.shtml

SASL and PAM

Simple Authentication and Security Layer (SASL) is a method for adding authentication to connection-based protocols

- used when a simple user/password authentication is not enough
- protocols supporting SASL include: SMTP, IMAP, POP, LDAPv3 (but not v2)

Pluggable Authentication Modules (PAM) is a method for adding authentication to applications and services

- "Plug in" different authentication methods
- Different services can have different authentication policies

Most linux distributions are "pam-aware" (compiled with the /usr/lib/libpam.a library) – but ours is not

SASL

- Usually used for authentication of systems
- For example:
  - MUA to MSA (webmail to mailserver)
  - MSA to MTA (mailserver to mailserver)
  - MTA to MDA (mailserver to IMAP server)
  - MUA to Directory (webmail to LDAP server)

SASL

- A SASL capable protocol includes configuration directives for authenticating a client to a server and for optionally negotiating a security layer
- Servers can be configured to negotiate and use possibly nonstandard and/or customized mechanisms for authentication
- A challenge/response system is used to get the needed information from the client, up to the point the authentication is either successful or fails.
- Usually used for authentication of systems: sasl fills one small gap in our security policy - port 587 authentication

SSH doesn't use SASL because it predates it, and there has not been interest making something that uses SASL and TLS

PAM

There are four groups of PAM modules for independent management:

- **Account modules** check that the specified account is a valid authentication target under current conditions. This may include conditions like account expiration, time of day, and that the user has access to the requested service.
- **Authentication modules** verify the user's identity, for example by requesting and checking a password or other secret. They may also pass authentication information on to other systems or modules.
- **Password modules** are responsible for updating passwords, and are generally coupled to modules employed in the authentication step. They may also be used to enforce strong passwords.
- **Session modules** define actions that are performed at the beginning and end of sessions. A session starts after the user has successfully authenticated.

PAM

- Which authentication module is to be used depends on the local system setup and is at the discretion of the local system administrator.

- Modules are *stacked* (order is important)

Sample PAM configuration in /etc/pam.d

```
interface  control flag   module name
auth       required       pam_nologin.so
auth       required       pam_securetty.so
auth       sufficient     pam_unix.so
auth       required       pam_ldap.so
```

lots of modules are available
http://www.linux-pam.org/modules.html

PAM

Standard Warning:

- You can lock yourself out of your own system if you are not careful!

- (also applies to SELinux) (next week)

PAM, LDAP and Network Accounts

- Local accounts are unique to each host. Changes to an account (e.g., new password) on one host do not affect similar accounts on other hosts
- A networked account is a single user identity shared by many hosts. Changes to the account globally affect all other hosts
- PAM is required to use a LDAP server to provide user information rather than the local database of users (/etc/passwd and /etc/shadow)

Note:
We see this in system descriptions, so we need to know what it is even if we don't use it

Scenario

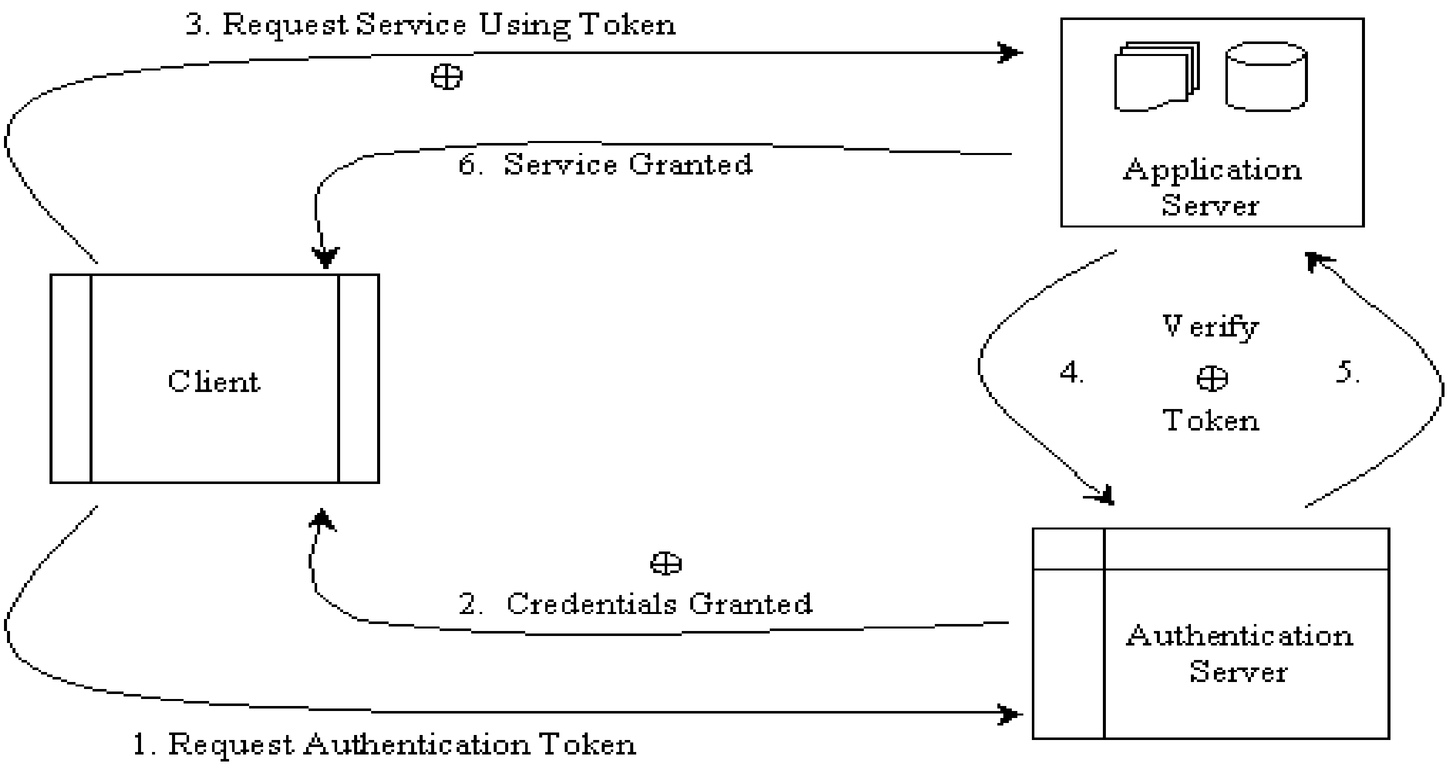
ssh to server, network account

- Caught by iptables, source & destination allowed <or>

  Caught by tcpwrap, host & service allowed (redundant?)
- Caught by PAM, request passed to LDAP client application
- LDAP client contacts LDAP server
- LDAP server requires SASL authentication before bind
- Authentication successful, LDAP server returns username and (encrypted) password
- LDAP client passes these back to PAM
- Next PAM module verifies credentials
- ssh connection is allowed (whew!)

AAA: Kerberos (Microsoft Active Directory)

Kerberos



Chapter 11 Diagram 1
Electronic Commerce: Economics, Management, Marketing, and Technology
By Thomas O'Daniel
© Inkblot Sdn Bhd 1999

AAA: RADIUS

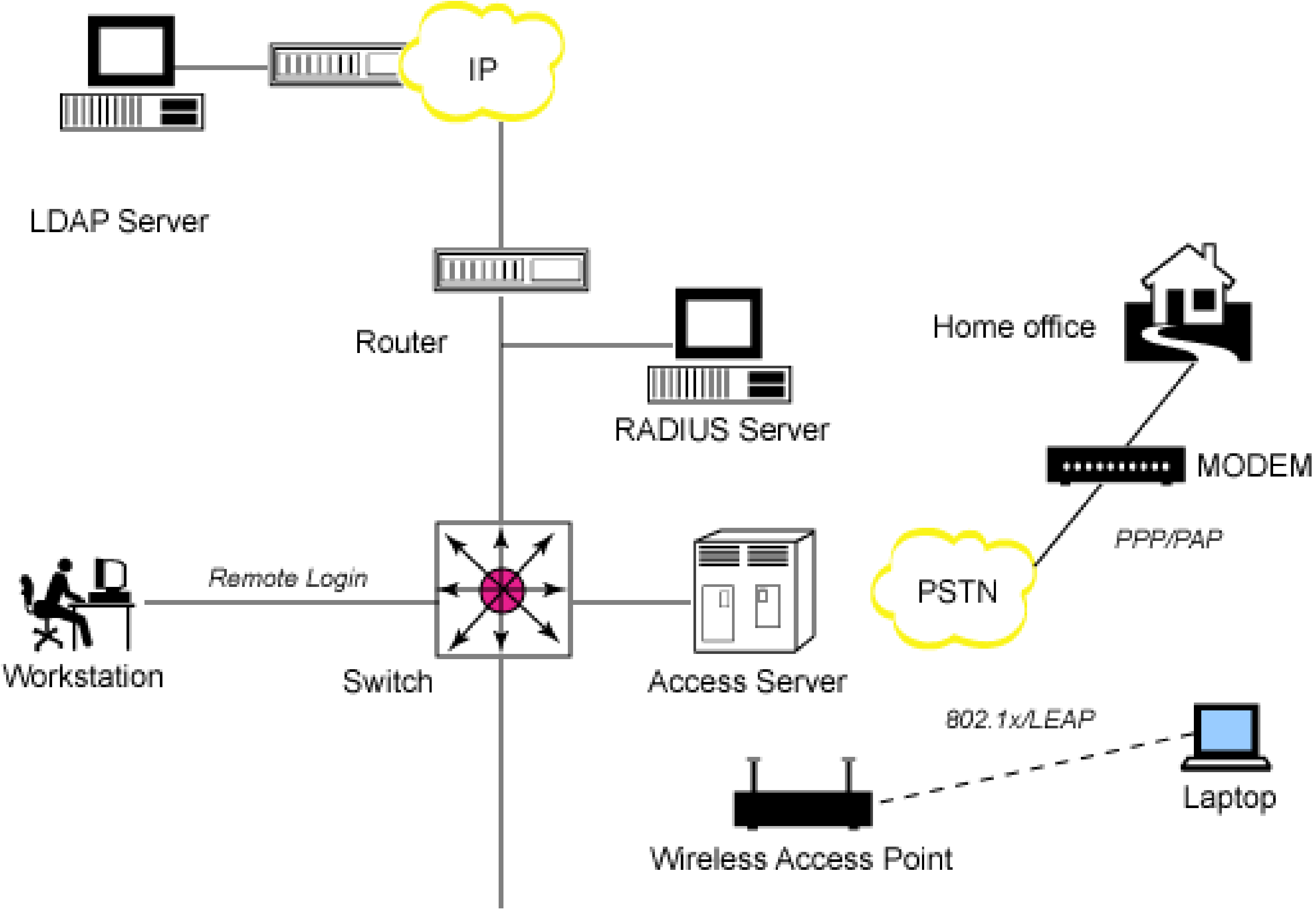
Remote Authentication Dial-In User Service (RADUIS)

- Industry standard for AAA support, developed at Livingston (now Lucent)
- The IETF RADIUS specification (RFC2865) and RADIUS accounting standard (RFC2866) along with RFC2868 (RADIUS Attributes for Tunnel Protocol Support).
- Centralised database of user information:
  - authentication & configuration information
  - type of service permitted

AAA: RADIUS

- RADIUS provides authentication and authorization in a single step.
- When the user logs into the network, the Network Access Server (NAS) prompts the user for a username and a password.
- The NAS will then send the request to the RADIUS server. The NAS may include a request for access restrictions or per-user configuration information.
- The RADIUS server returns a single response with authentication approval status and any related access information available.

Figure 1. Authentication via RADIUS and LDAP



http://www.ibm.com/developerworks/library/l-radius/

Q&A

OpenLDAP: Ground up

1. If you use iptables, open port 389 (636 for ldaps)

```
iptables -A INPUT -s 198.168.0.0 -p tcp --destination-port 389 -j ACCEPT
iptables -A INPUT -s 198.168.0.0 -p udp --destination-port 389 -j ACCEPT
```

2. Configure /etc/openldap/slapd.conf
3. Build an initial data base
   - Design a Schema
     - Find out which apps require what
   - Design a Tree (DIT)
     - First you build the base (tags like “C”, “DC”, “O”, and “OU”)
     - then the branches (the individual “DN” or distinguished names)
     - then the rest (tags like “address”, “title”, “mail”)
4. Create an LDIF data file, and populate the LDAP server using the OpenLDAP database tools
5. Start the `slapd` service on your machine

LDAP attributetype Definition

```
#  Sspecifies the favourite drink of an object
#  (or person). [cosine schema]

attributetype ( 0.9.2342.19200300.100.1.5
   NAME ( 'drink' 'favouriteDrink' )
   DESC 'RFC1274: favorite drink'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

OID (Object Identifier)
Always begins with a string of numbers delimited by decimals.

Every attribute has one or more NAMEs

In LDAP names are preferred for queries rather than OIDs (opposite of SNMP)

Matching rules: define methods of comparison, e.g., case sensitive or case insensitive. Matching rules are typically built-in to the LDAP server and do not need to be defined explicitly. ***See OpenLDAP v2.3 Admin.Guide section 9.2***

SYNTAX: specifies the data type, e.g., string, number etc.

Every attribute has a data type – ***See OpenLDAP v2.3 Admin.Guide section 9.2***

Typical LDAP Schema (1)

```
attributetype ( 2.5.4.41 NAME 'name'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )

attributetype ( 2.5.4.6 NAME ( 'c' 'countryName' )
  DESC 'RFC2256: ISO-3166 country 2-letter code'
  SUP name SINGLE-VALUE )

attributetype ( 2.5.4.8 NAME ( 'st' 'stateOrProvinceName' )
  DESC 'RFC2256: state or province which this object resides in'
  SUP name )

attributetype ( 2.5.4.10 NAME ( 'o' 'organizationName' )
  DESC 'RFC2256: organization this object belongs to'
  SUP name )

attributetype ( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
  DESC 'RFC2256: organizational unit this object belongs to'
  SUP name )

attributetype ( 2.5.4.3 NAME ( 'cn' 'commonName' )
  DESC 'RFC2256: common name(s) for which the entity is known by'
  SUP name )
```

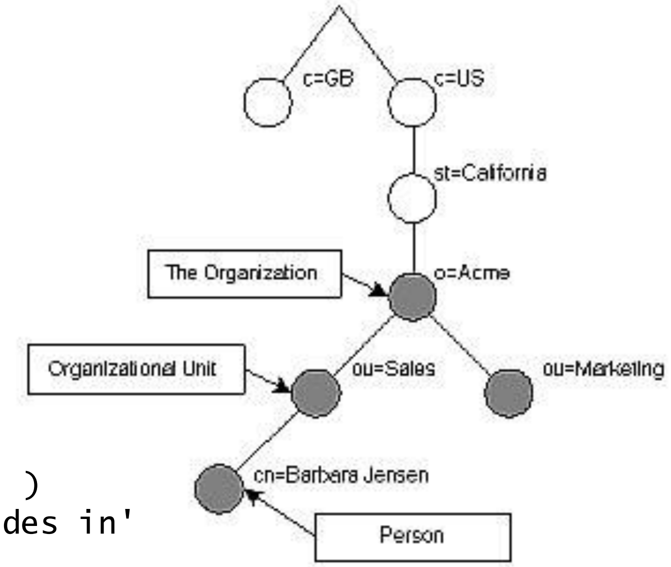


Figure 1.1: LDAP directory tree (traditional naming)

supports inheritance: **SUP**

Typical LDAP Schema (2)

```
attributetype ( 0.9.2342.19200300.100.1.25
  NAME ( 'dc' 'domainComponent' )
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

attributetype ( 2.5.4.10
  NAME ( 'o' 'organizationName' )
  DESC 'RFC2256: organization this object belongs to'
  SUP name )

attributetype ( 2.5.4.11
  NAME ( 'ou' 'organizationalUnitName' )
  DESC 'RFC2256: organizational unit this object belongs to'
  SUP name )

attributetype ( 0.9.2342.19200300.100.1.1
  NAME ( 'uid' 'userid' )
  DESC 'RFC1274: user identifier'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

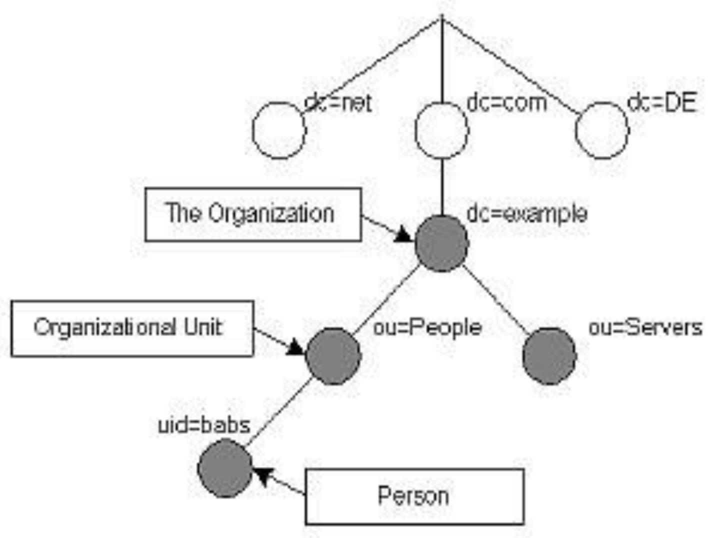


Figure 1.2: LDAP directory tree (Internet naming)

Several standard schemas:

- core
- cosine
- inetorgperson

LDAP objectClass

- One or more **objectClass**(es) must be present in an LDAP entry.
- Denotes the type of object being represented by a directory entry or record – **a set of attributes**
  - listed as must contain (mandatory) and may contain (optional)
- An attribute defined in one schema can be used by an objectClass defined in another schema.
  - An attribute may be optional in one objectClass and mandatory in another
- There are a confusing number of pre-defined objectClasses, each of which contains bucket-loads of attributes for almost all common applications. But of course the one you NEED is never defined

LDAP objectClass

An object class is declared as **abstract**, **structural**, or **auxiliary**.

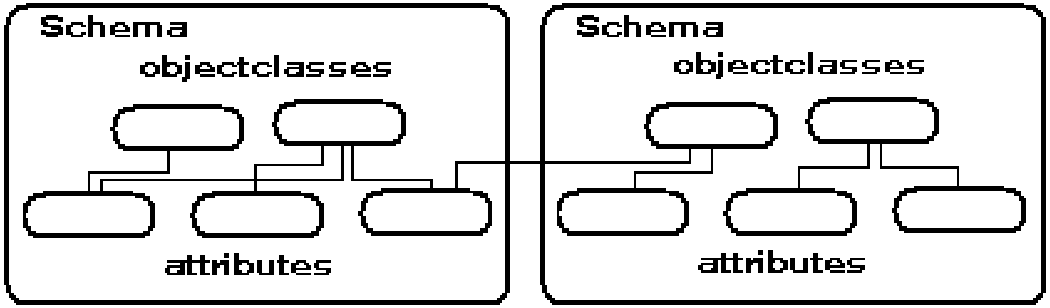
- Directory entries are instantiated from **structural** object classes.
- An **abstract** objectClass is used as a template for creating other object classes.
  - A directory entry cannot be instantiated from an abstract object class.
- **Auxiliary** object classes provide a method for extending structural object classes without having to change the schema definition of a structural class.
  - An auxiliary object class cannot be instantiated by itself as a directory entry;

➢ There are some objectClasses and attributes defined as **operational** which are embedded in the LDAP server software and do not need definition.

LDAP objectClass Definition

```
objectclass ( 0.9.2342.19200300.100.4.13 NAME 'domain'
  SUP top STRUCTURAL
  MUST domainComponent
  MAY ( associatedName $ organizationName $ description $ businessCategory $ seeAlso
      $ searchGuide $ userPassword $ localityName $ stateOrProvinceName $ streetAddress
      $ physicalDeliveryOfficeName $ postalAddress $ postalCode $ postOfficeBox
      $ streetAddress $ facsimileTelephoneNumber $ internationalISDNNumber
      $ telephoneNumber $ teletexTerminalIdentifier $ telexNumber
      $ preferredDeliveryMethod $ destinationIndicator $ registeredAddress
      $ x121Address ) )

objectclass ( 0.9.2342.19200300.100.4.15 NAME 'dNSDomain'
    SUP 'domain' STRUCTURAL
    MAY ( ARecord $ MDRecord $ MXRecord $ NSRecord $ SOARecord $ CNAMERecord ) )
```



OpenLDAP and LDIF

- The LDIF format is unusual, it makes use of tags, but is also position sensitive
- Each line begins with a tag, then a full colon, then ONE space, then data
- A space in column one denotes a continuation of the previous line
- Tabs are forbidden, and be very careful about errors like alphas in numeric fields

- **OpenLDAP is obnoxiously unforgiving of syntax errors**
  - Error messages are sparse and cryptic, if you get them
  - Beware of trailing spaces ...

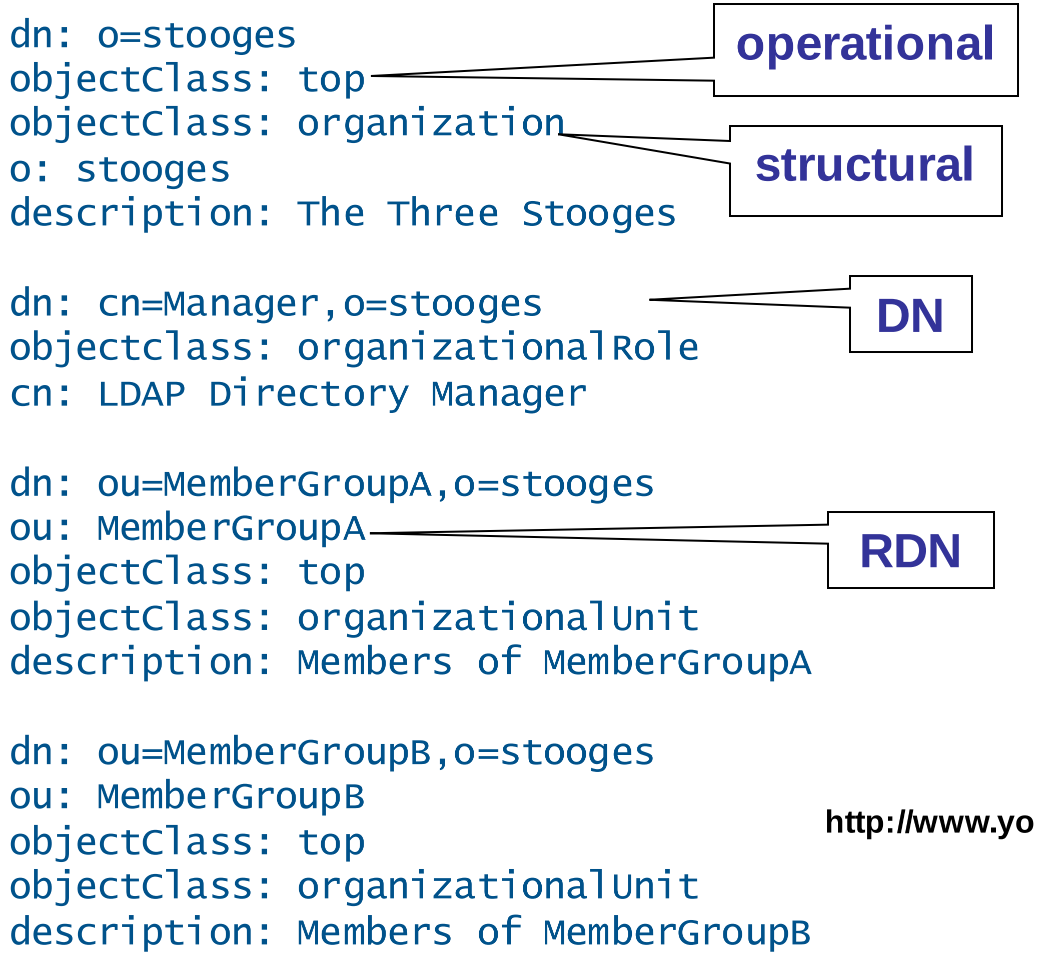
Build the database: LDIF

- LDIF (LDAP Data Interchange Format)
  - RFC 2849 documents the format and fields

- LDIF files are simple text files
  - Can be created and edited with any suitable text editor.

- LDIF files are used in five general cases:
  - To initially construct the DIT structure.
  - To add (import) bulk records into a directory.
  - To restore (import) a directory.
  - To archive (export) a directory.
  - To apply bulk edits to a directory.

LDIF: Begin with base entries



```
dn: o=stooges
objectClass: top
objectClass: organization
o: stooges
description: The Three Stooges

dn: cn=Manager,o=stooges
objectclass: organizationalRole
cn: LDAP Directory Manager

dn: ou=MemberGroupA,o=stooges
ou: MemberGroupA
objectClass: top
objectClass: organizationalUnit
description: Members of MemberGroupA

dn: ou=MemberGroupB,o=stooges
ou: MemberGroupB
objectClass: top
objectClass: organizationalUnit
description: Members of MemberGroupB
```

operational

structural

DN

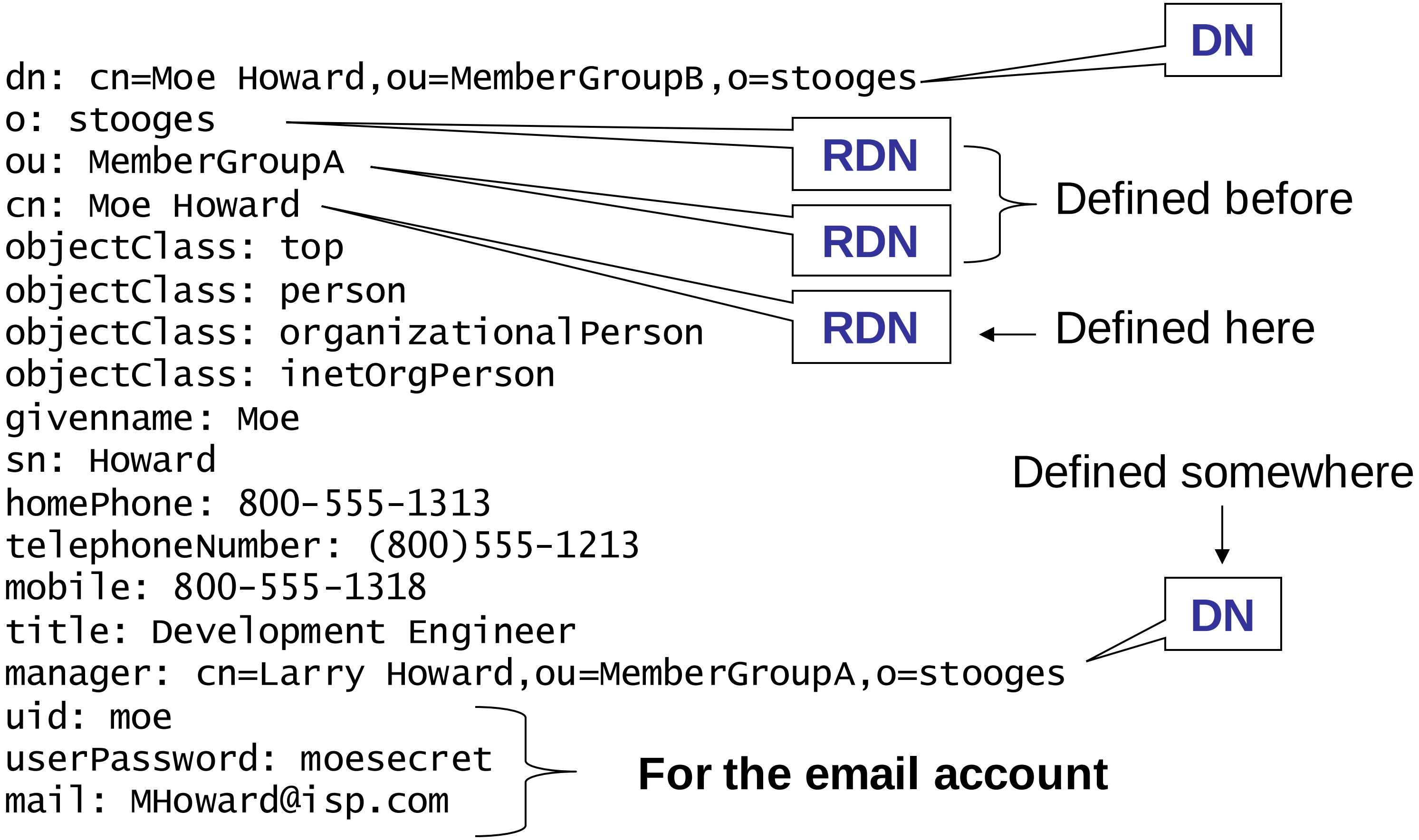
RDN

Each LDIF record begins with a unique DN (usually a "common name" + the Parent DN), followed by several data elements.

`objectClass: top`
is not defined in the schemas, but everything else is

http://www.yolinux.com/TUTORIALS/LinuxTutorialLDAP.html

LDIF: Continue with detail entries



```
dn: cn=Moe Howard,ou=MemberGroupB,o=stooges
o: stooges
ou: MemberGroupA
cn: Moe Howard
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
givenname: Moe
sn: Howard
homePhone: 800-555-1313
telephoneNumber: (800)555-1213
mobile: 800-555-1318
title: Development Engineer
manager: cn=Larry Howard,ou=MemberGroupA,o=stooges
uid: moe
userPassword: moesecret
mail: MHoward@isp.com
```

For the email account

http://www.yolinux.com/TUTORIALS/LinuxTutorialLDAP.html