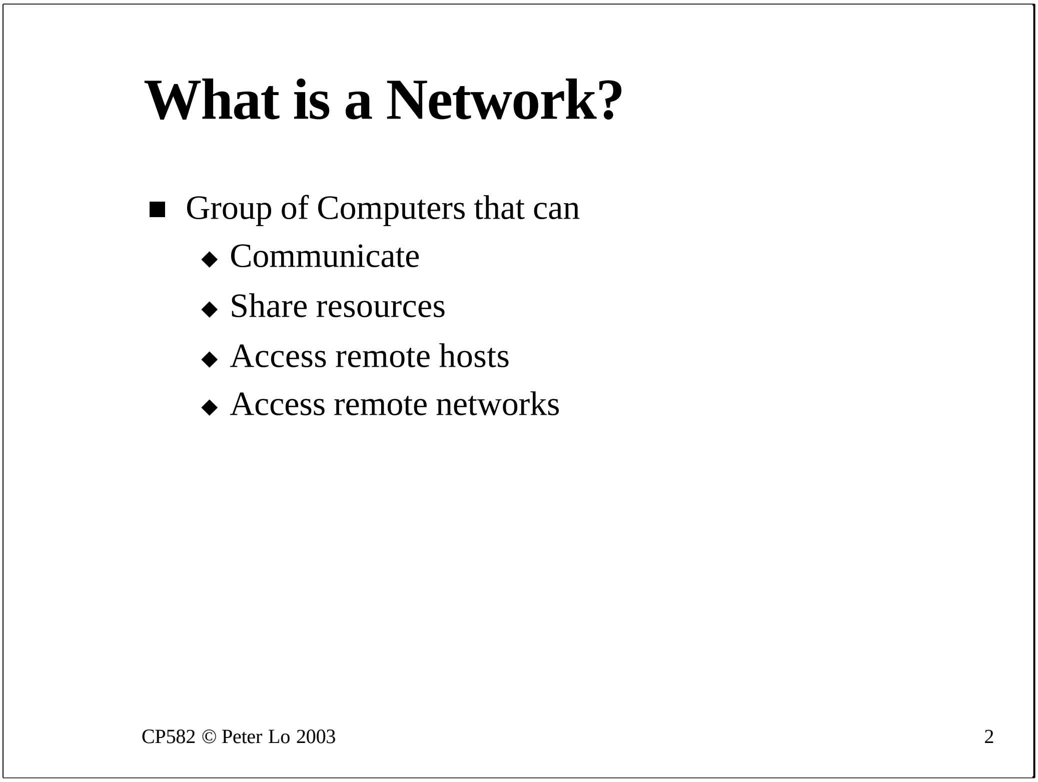
Novell Directory Services
NDS

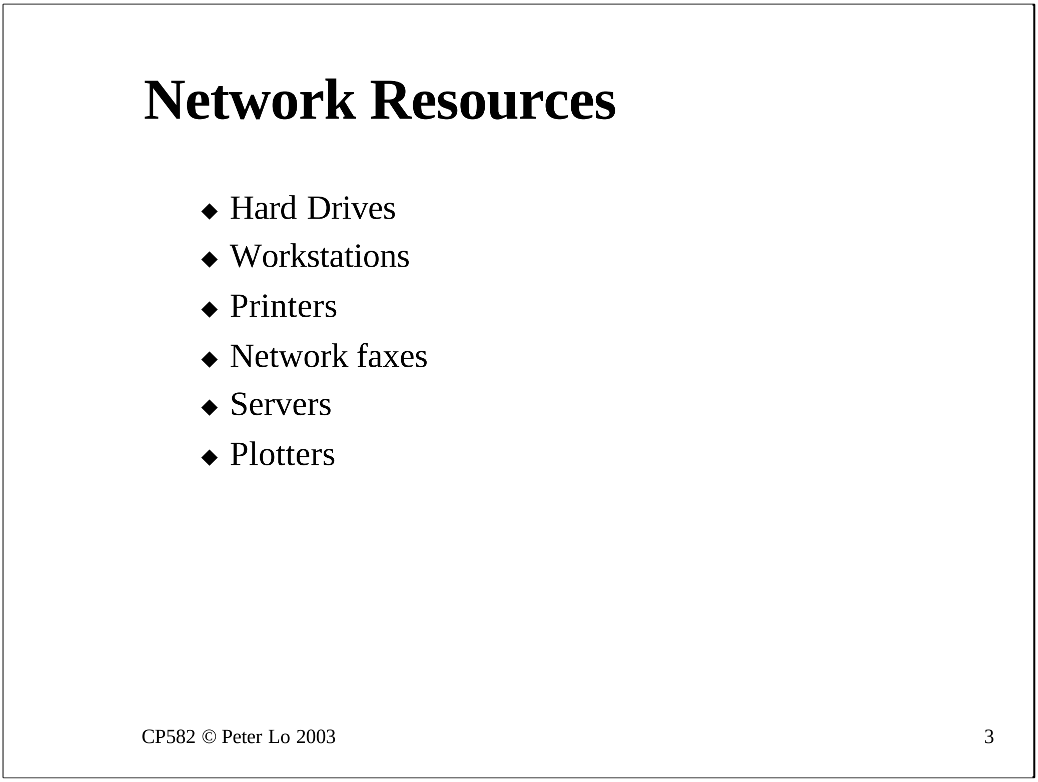
Peter Lo



What is a Network?

- Group of Computers that can
  - Communicate
  - Share resources
  - Access remote hosts
  - Access remote networks

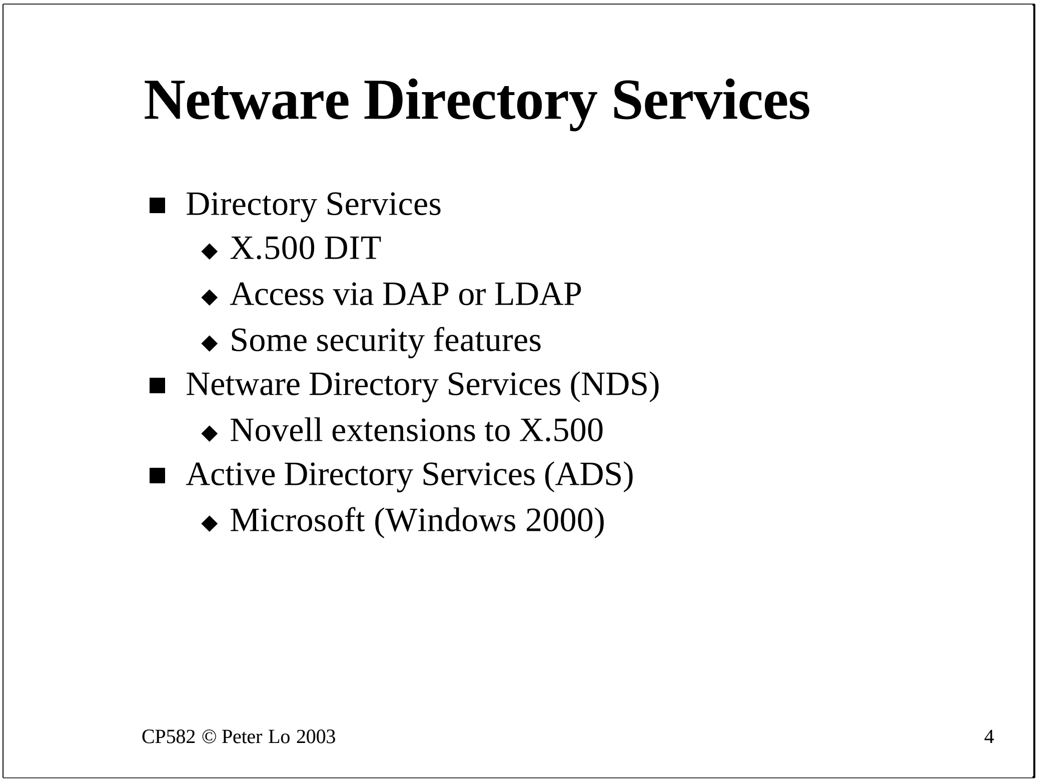
Foundation stuff - just a brief reminder of what we are setting out to accomplish with a network - the sharing of resources and data between a number of computers without regard to location.



Network Resources

- Hard Drives
- Workstations
- Printers
- Network faxes
- Servers
- Plotters

The resources that might be shared fall into many categories and the ones shown here are not an exhaustive list. The point of this slide is to prepare you for the idea that there are many different **types of object** that might be shared through a network and that the requirements of sharing one type of object might be different from the those of sharing another.



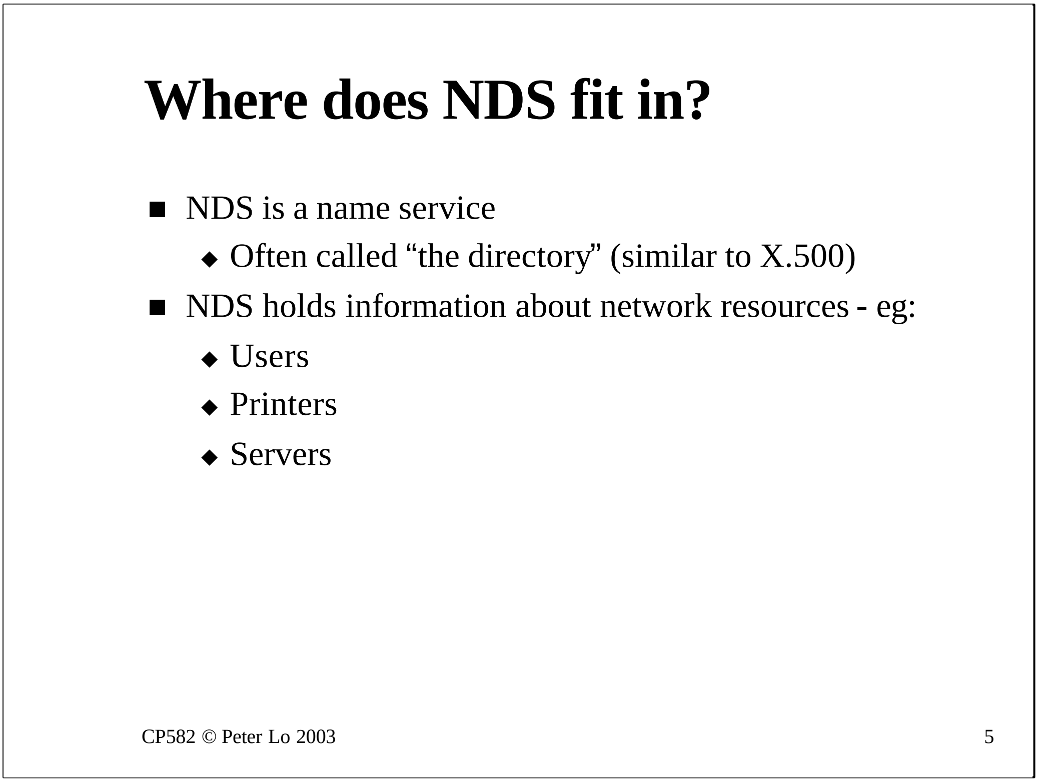
Netware Directory Services

- Directory Services
  - X.500 DIT
  - Access via DAP or LDAP
  - Some security features
- Netware Directory Services (NDS)
  - Novell extensions to X.500
- Active Directory Services (ADS)
  - Microsoft (Windows 2000)

The idea of a Directory Service, as we looked at last week in the form of X.500 accessed via LDAP, is such a good one that Novell decided in the late 80's to base their new operating system's user account management of X.500.

We will see later the extent to which X.500 was extended - the Novell extensions are significant but the basic model of X.500 has been preserved which means, for example, that an NDS/X.500 gateway is has quite a simple job to do.

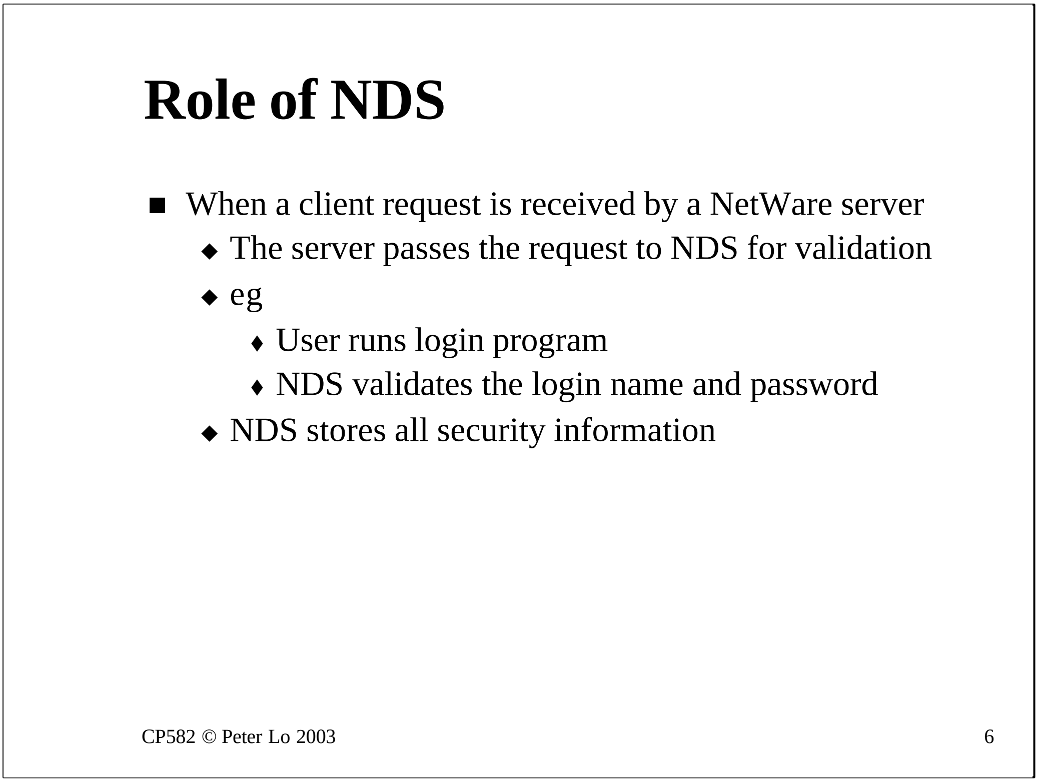
Microsoft are also about to come out with an X.500 based directory in the form of ADS which is part of Windows 2000.



Where does NDS fit in?

- NDS is a name service
  - Often called “the directory” (similar to X.500)
- NDS holds information about network resources - eg:
  - Users
  - Printers
  - Servers

NDS, in a Novell network, is the database in which all information about objects involved in network sharing is stored.



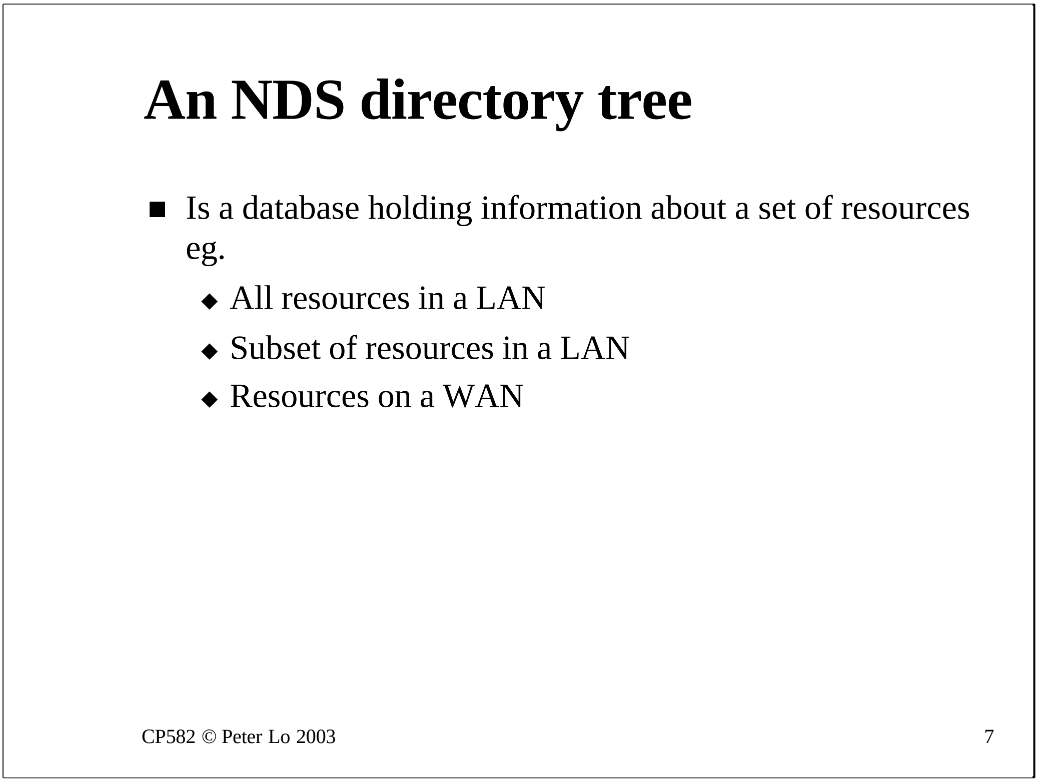
Role of NDS

- When a client request is received by a NetWare server
  - The server passes the request to NDS for validation
  - eg
    - User runs login program
    - NDS validates the login name and password
  - NDS stores all security information

X.500 is a Directory Service through which details about objects can be looked up.

NDS is much more involved in the detailed functioning of a Novell network. Whenever a Novell server needs to check that a particular activity is permitted it will refer to NDS to check on the security settings.

As a directory service NDS is far more active than X.500 - if you were to look at the packets on a Novell network you would find that there was a constant stream of NDS packets from all workstations because NDS is the basis for all network security.



An NDS directory tree

- Is a database holding information about a set of resources eg.
  - All resources in a LAN
  - Subset of resources in a LAN
  - Resources on a WAN

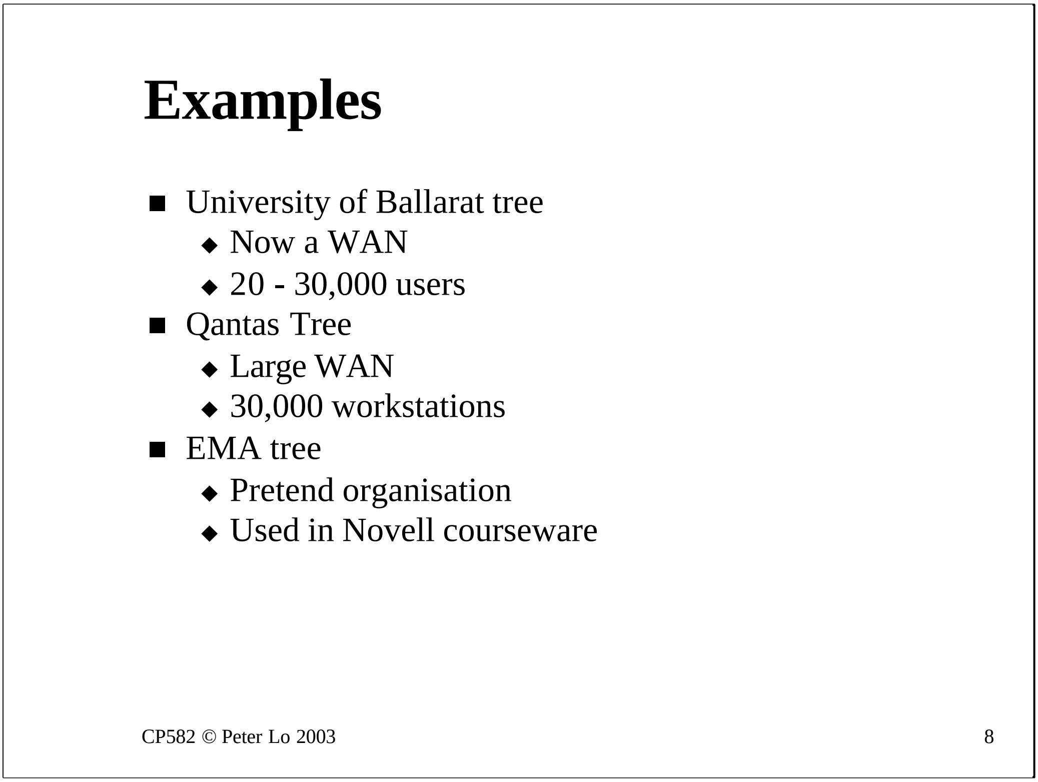
The scope of the DIT in X.500 was the whole world whereas NDS trees are applied to subsections of the global network (although there are some attempts to set up a worldwide NDS tree)

An NDS tree could define resources within a:

• **WAN**
A company with branches throughout the world could use a single NDS tree to store all information from all those branches

• **LAN**
The most common example of an NDS tree would be one that stored all the information relating to a single site.

• **Part of a LAN**
Some companies might find reason to setup multiple NDS trees within a single LAN. It is also possible that a LAN (or MAN) might be shared by several companies in which case there would probably be multiple NDS trees.



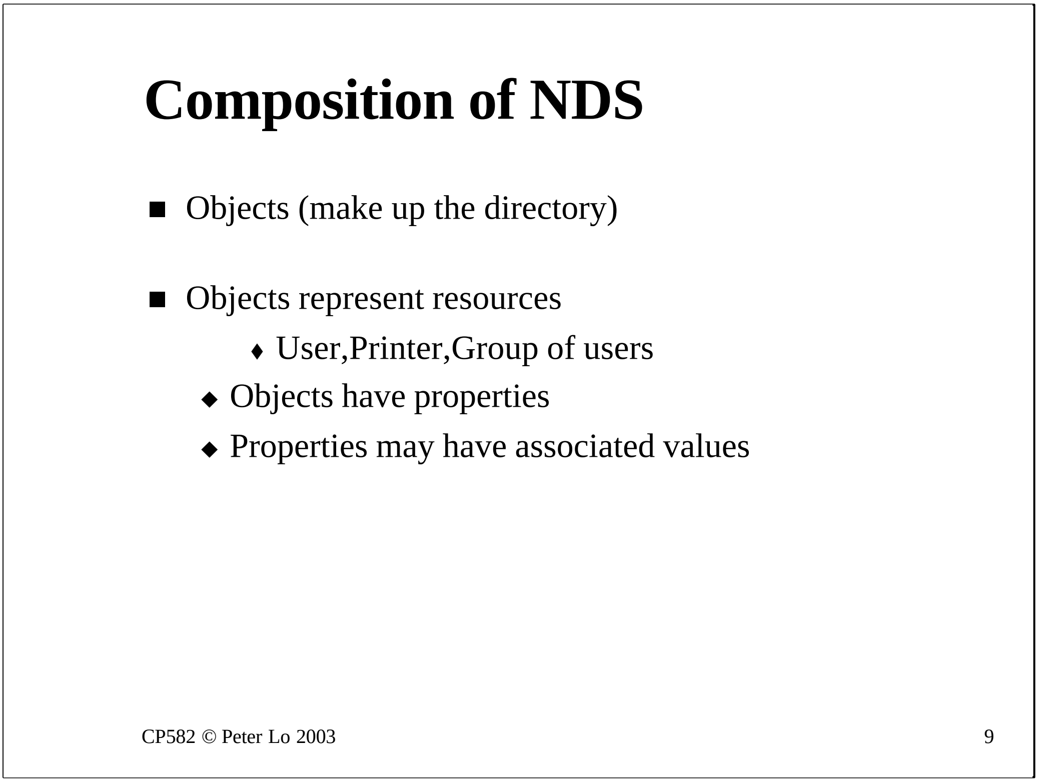
Examples

- University of Ballarat tree
  - Now a WAN
  - 20 - 30,000 users
- Qantas Tree
  - Large WAN
  - 30,000 workstations
- EMA tree
  - Pretend organisation
  - Used in Novell courseware

The NDS tree that you have already made extensive use of is the University of Ballarat tree. This tree was setup in 1994 for the University LAN but since the amalgamation of the Uni with SMB and Wimmera TAFEs the tree is being extended to cover the WAN.

Qantas are one of the largest users of Netware around. Their tree spans a very large WAN and contains many users and many workstations. When you board a Qantas plane the automatic ticket checking machine is connected into the Qantas NDS tree.

For the purposes of learning about Novell networks Novell use a fictional company EMA. The EMA NDS tree spans a worldwide organisation.



Composition of NDS

- Objects (make up the directory)
- Objects represent resources
    - User,Printer,Group of users
  - Objects have properties
  - Properties may have associated values

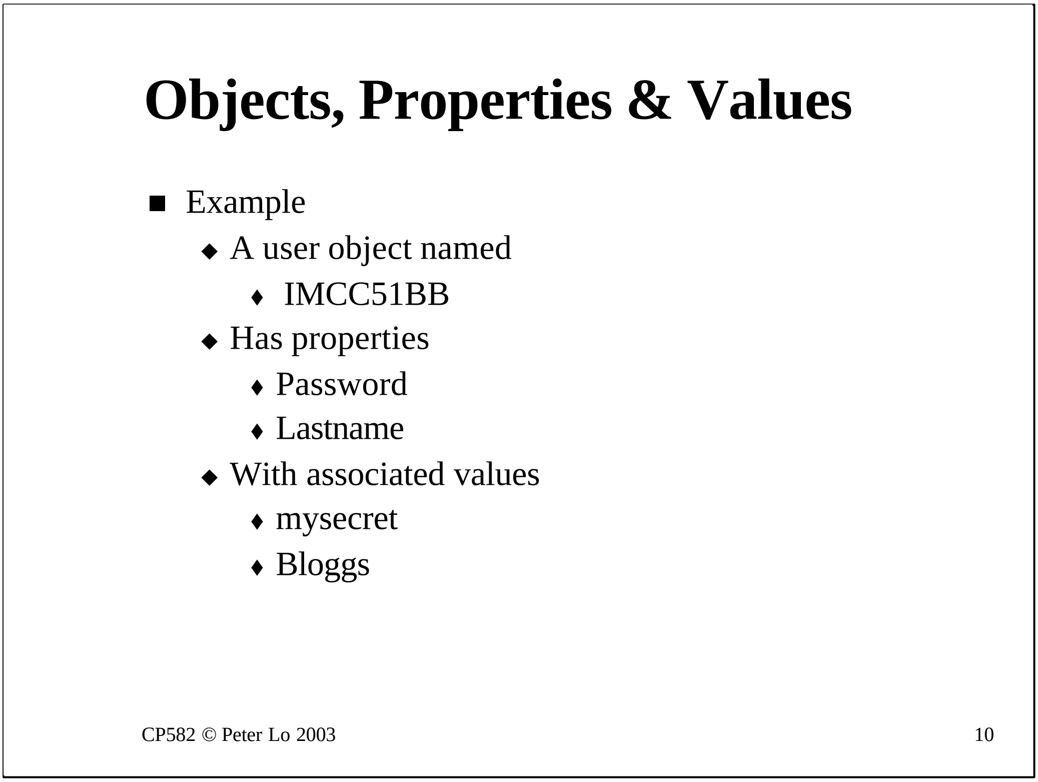
What is stored in NDS?

Objects, Properties and Values are the components of the NDS database.

The **objects** tie the tree structure together and represent real world entities such as Organisational Units, Users, Computers, Groups (of users).

Each different type of object has a set of **properties** that it may, or may not, have.

Each property of each object, if it has been set, can have a **value**



Objects, Properties & Values

- Example
  - A user object named
    - IMCC51BB
  - Has properties
    - Password
    - Lastname
  - With associated values
    - mysecret
    - Bloggs

Here we see an example of the Object/Property/Value hierarchy in the case of a User object.

Every single object has a Common Name property (just like X.500) which in this case is IMCC51BB but since this is a User object there are other properties that can be defined.

Lastname

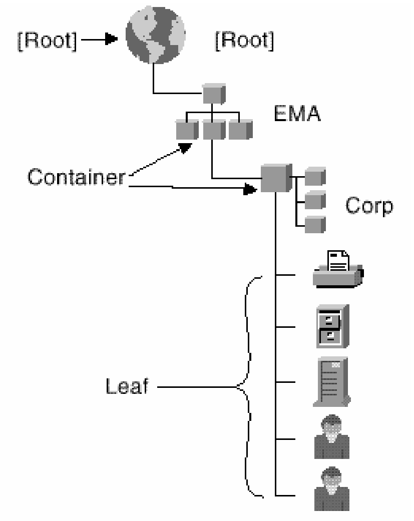
The Lastname property of a User has been defined as **mandatory -**it is not possible to create a user without a last name - even if the user is “Guest” you need a last name (I usually use “user” as the last name of IDs like this)

Password

Password is an optional property and only therefore exists if one is created.

Types of Objects

- [Root]
- Container
- Leaf



The big picture division amongst objects is simple:

[root]

Always written like this (with square brackets) [root] is the very top of the NDS tree. In a given tree you cannot have a context higher than [root] and all other contexts are descendants of [root] (just like a hard disk).

[root] itself is anonymous - there is no other name that can be set for it although seen from the outside the NDS tree does have a name which can be set. This is like the name that can be set for your C: drive in W95 - whatever you name the drive the root of the drive is still \ (backslash).

Container

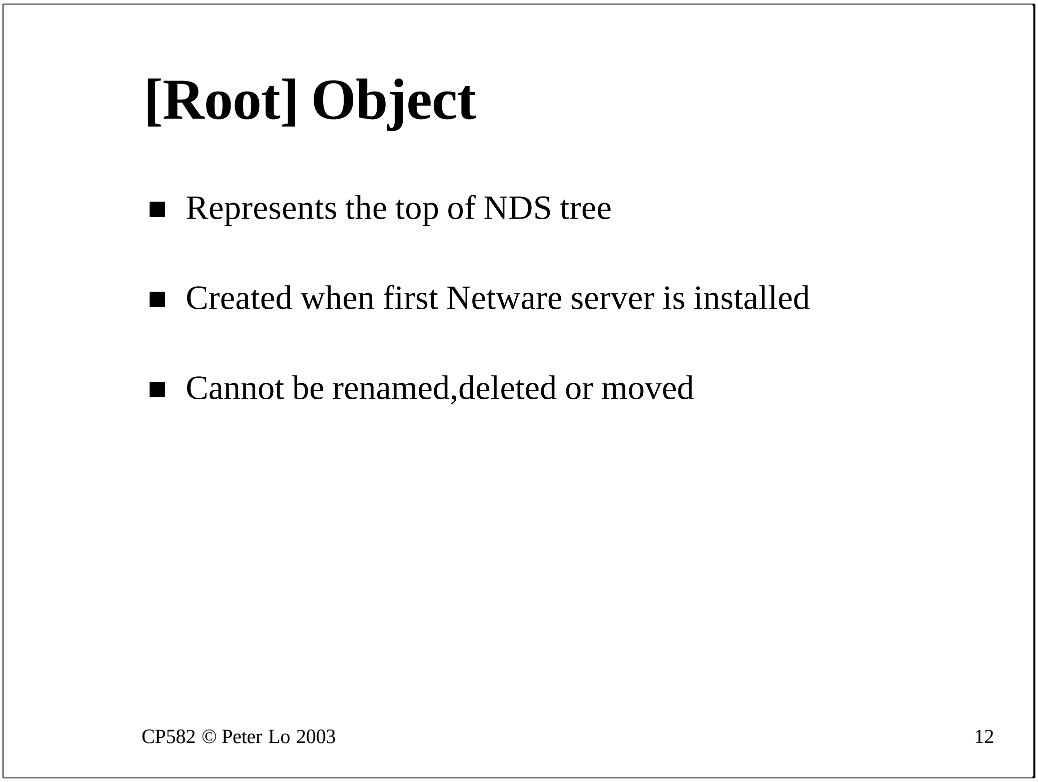
If an object can contain other objects it is termed a **container** The containers that we saw in X.500 were Country, Organisation & Organisational Unit. In the X.500 browsing you may have also come across Locality containers.

A container in NDS is like a directory in the file system.

Leaf

An NDS object that cannot contain other NDS objects is termed a **leaf** object. This is like a file in the file system.

It is important to understand that leaf objects (like files) can contain all kinds of other data - what make them leaf objects is that they cannot **contain** other NDS (or in the case of files - file system) objects.



[Root] Object

- Represents the top of NDS tree
- Created when first Netware server is installed
- Cannot be renamed,deleted or moved

The [root] object is created when the first Netware server is installed in a given NDS tree.

The [root] object cannot be renamed ([root] is the only name it will ever have), deleted (only by deleting the whole tree) or moved (it is at the top and will always be at the top of the tree)

Container Objects

- Country
  - 2 letter designator
  - Optional
  - Can only exist in [root]
- Organisation
  - Usually identifies Company, Division
  - Can only exist in [root] or a Country

Considering the Container objects in turn:

Country

As you saw in X.500 country objects have two letter identifiers such as US, AU or SA. The latter designator refers to “Saudi Arabia” and it was bug in the X.500 browser that caused it to identify “South Australia” (a Locality within Australia as Saudi Arabia)

The only place where Country objects can be created is [root] - they are not legal further down the tree - but they are optional in NDS. Most NDS trees are local to a company or organisation and do not require Country objects.

Organisation

Usually identifies a company or division.

There are placement rules for this object as well - it has to be near the top of the tree - either in [root] or in a Country object under [root]

Container Objects

- Organisational Unit
  - Identifies a smaller business unit
    - School of IT
    - Sales Dept

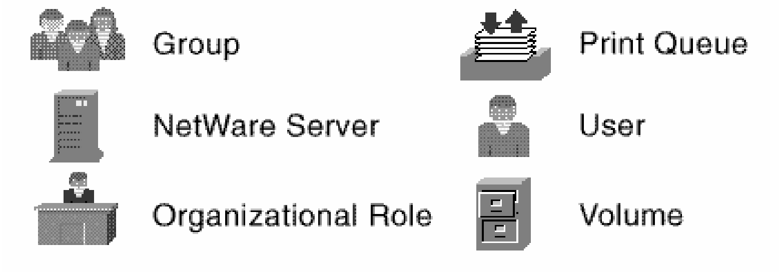
The Organisation Unit is the workhorse container in NDS.

It usually represents a sub-division within an Organisation - a section or division - but can be nested to many levels. OU’s can contain OU’s which can contain OU’s and so on.

An OU cannot be created in [root] or in a Country object though - you need an Organisation to set the OU ball rolling!

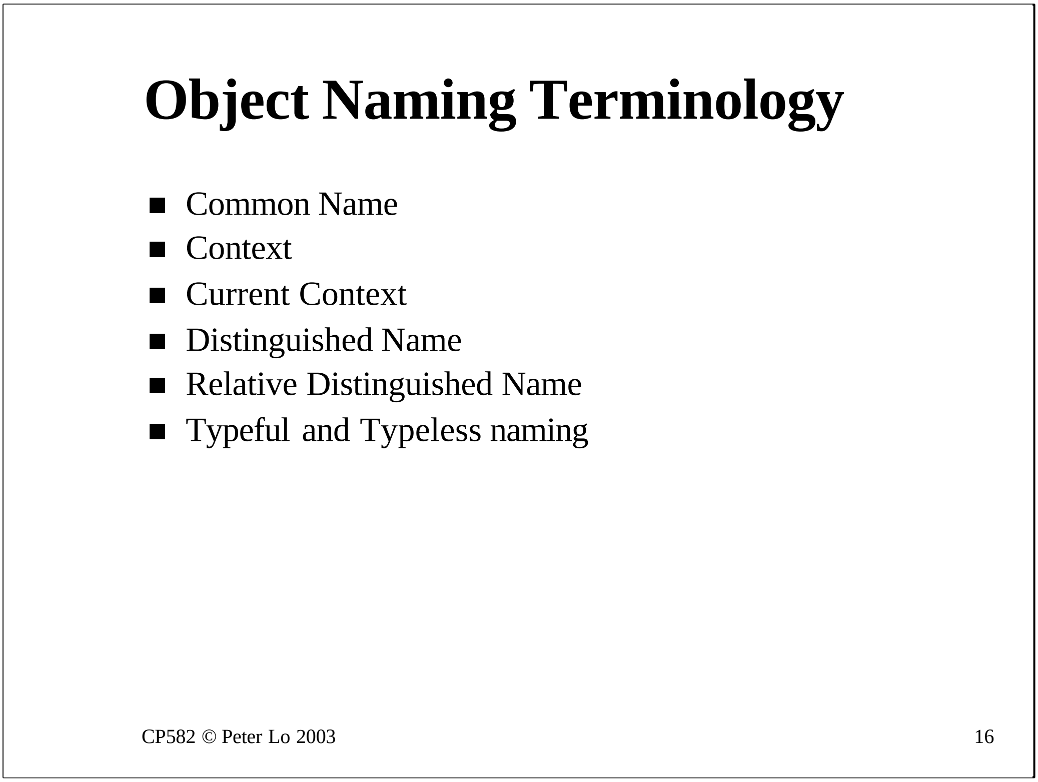
Leaf Objects

- Represent Resources
  - eg. Users,Groups,Servers,Volumes



Leaf objects are where the real data of NDS is stored and leaf objects are only allowed to be placed in, or below, and Organisation.

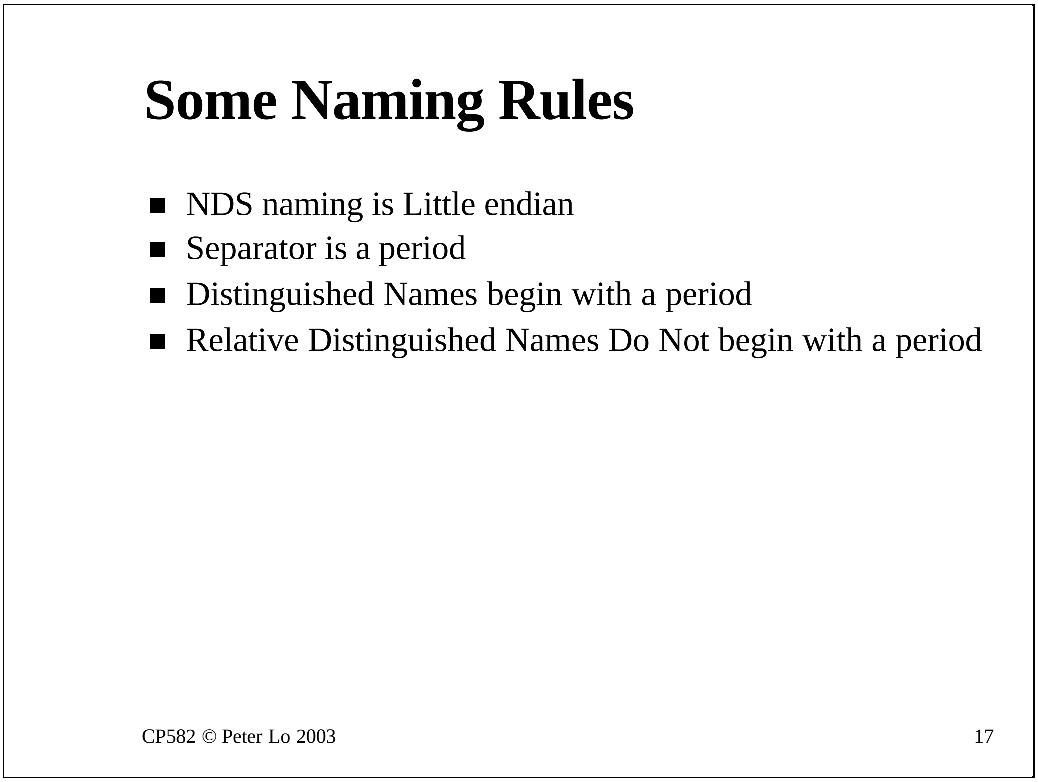
In terms of last week’s lab, if X.500 had to obey the NDS containment rules Leigh Hume could not exist where he does



Object Naming Terminology

- Common Name
- Context
- Current Context
- Distinguished Name
- Relative Distinguished Name
- Typeful and Typeless naming

These terms, which are used with NDS, should all make sense to you from the X.500 point of view. They all have exactly the same significance within NDS.



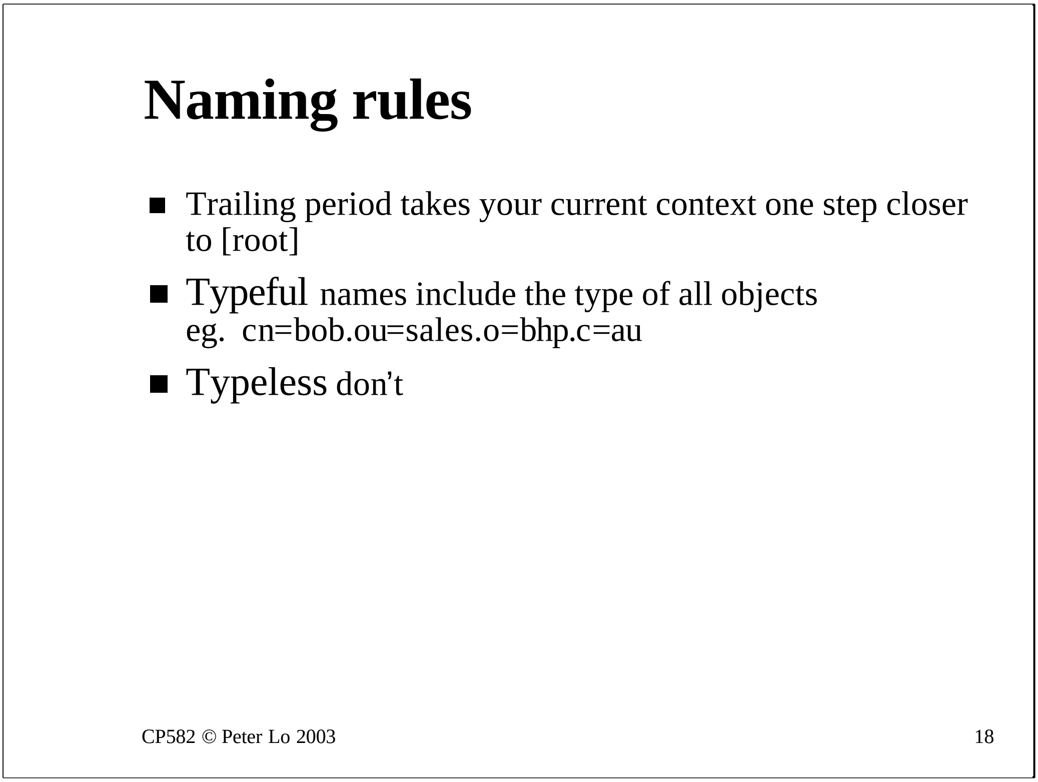
Some Naming Rules

- NDS naming is Little endian
- Separator is a period
- Distinguished Names begin with a period
- Relative Distinguished Names Do Not begin with a period

The syntax of NDS names is similar to, but different from (!) X.500

In common with X.500 NDS is little endian.

Unlike X.500 NDS uses a period as a separator - and as a complete name tag.

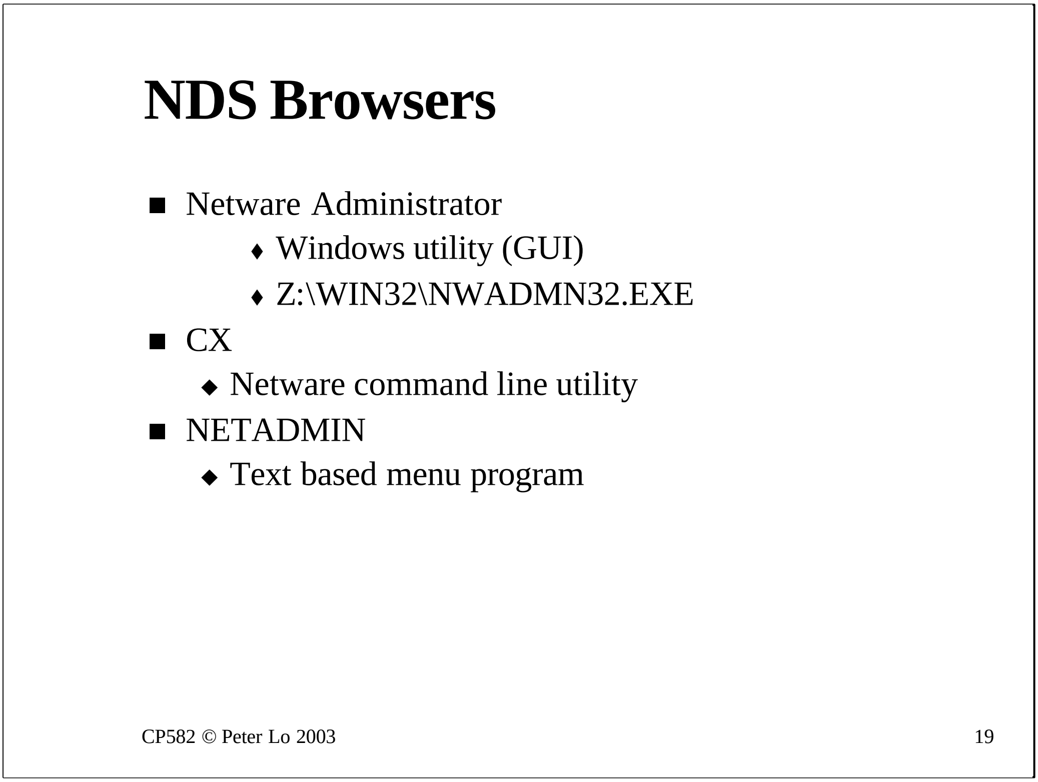


Naming rules

- Trailing period takes your current context one step closer to [root]
- Typeful names include the type of all objects
  eg. cn=bob.ou=sales.o=bhp.c=au
- Typeless don't

The "move up a level" indicator is a trailing period - at the end of the name.

Typeful has the same meaning as in X.500 - the term "typeless" is used to mean the reverse.



NDS Browsers

- Netware Administrator
  - ♦ Windows utility (GUI)
  - ♦ Z:\WIN32\NWADMN32.EXE
- CX
  - ◆ Netware command line utility
- NETADMIN
  - ◆ Text based menu program

You can view the NDS tree with a variety of tools:

Netware Administrator

This is a Windows program and the preferred tool for administrating NDS. It has the ability to perform almost any action required within NDS because as NDS is extended and added to the program grows to keep track by the addition of a **snap-in** for each new type of NDS object.

CX

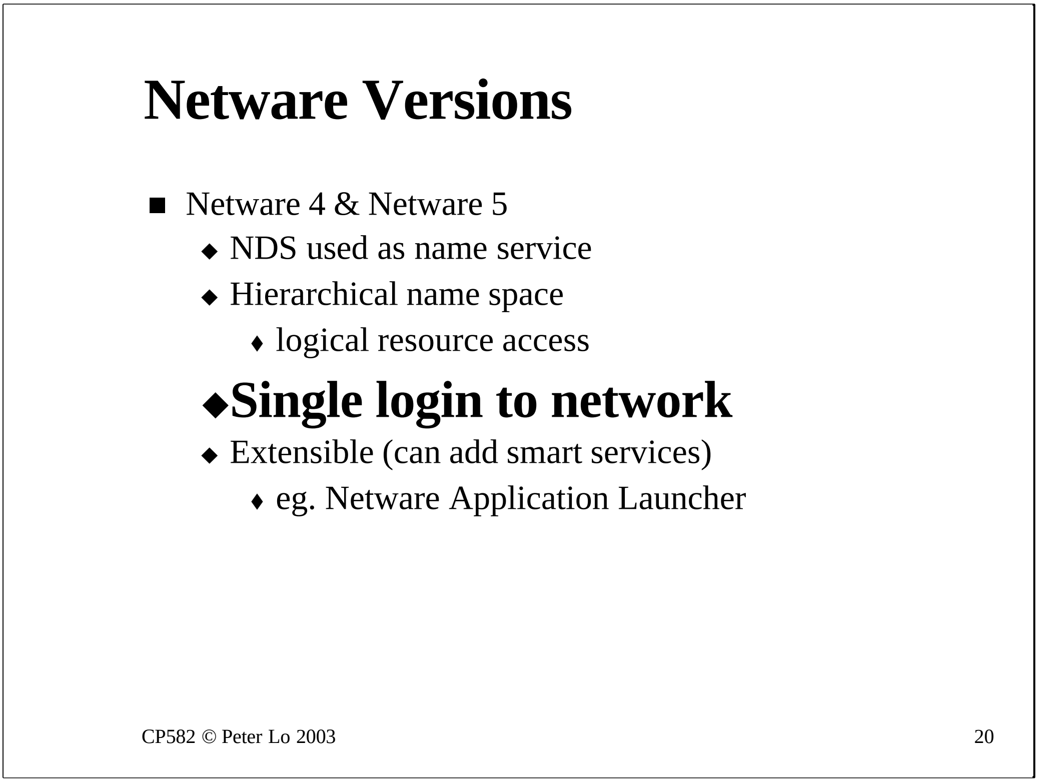
This DOS command line utility can view, but not change the contenys of the NDS tree.

It will allow you to see any information that you are permitted to see but the display is a raw dump of data to the screen which in many caases may be meaningless.

NETADMIN

This is a character-based menu utility (I often refer to these programs as "blue and yellow" - you will see why)

It is quick to load but is not keeping track of all the changes in NDS. A lot of types of object are now not supported by NETADMIN.



Netware Versions

- Netware 4 & Netware 5
  - NDS used as name service
  - Hierarchical name space
    - logical resource access
  - **Single login to network**
  - Extensible (can add smart services)
    - eg. Netware Application Launcher

NDS was developed for Netware 4 and since then has been the database through which all account creation is done.

The biggest single advantage of NDS is that it is (like X.500) a distributed database that all the servers in the tree share between them. This means that a single login to the network is all that is needed for access to any server. You see this with you student accounts that give you access to MFS2 and FS2 with a single login.

In earlier versions of Netware you would have had to login separately to each server.

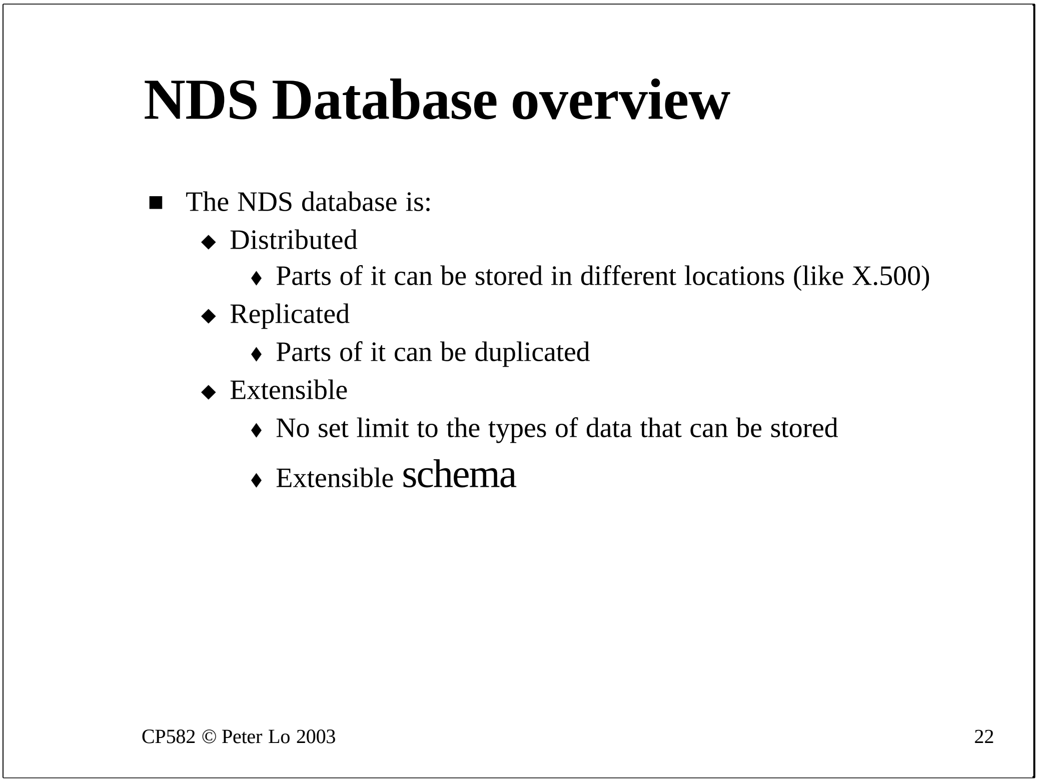
Netware 3

- Netware 3
  - Each separate server has a database of users:
    - The **Bindery**
  - Flat structure
    - No logical separation of resources
  - Must login to multiple servers to access their resources

Earlier versions of Netware (Netware 3 & 2) used a separate database of users on each server.

This database was known as the **Bindery** and was a flat list of all the users on that server - there was no hierarchy to provide a logical separation of resources.

As a consequence it was necessary to login separately to each server on which you needed to access resources.



NDS Database overview

- The NDS database is:
  - Distributed
    - Parts of it can be stored in different locations (like X.500)
  - Replicated
    - Parts of it can be duplicated
  - Extensible
    - No set limit to the types of data that can be stored
    - Extensible schema

The “dictionary” definition of NDS is “a distributed, replicated, extensible database” and it is the combination of these three aspects that account for the effectiveness of NDS.

Distributed

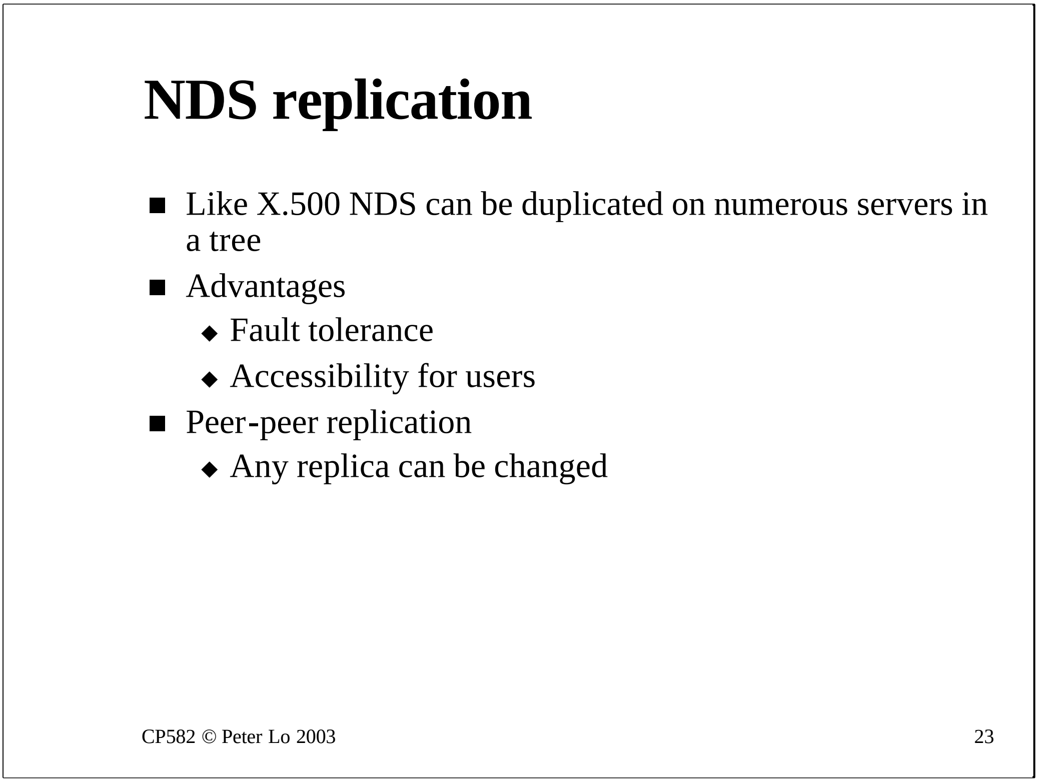
Different parts of the NDS database can be stored in different physical locations. The chief benefit from this is that the bits that you really need access to can be stored near to you. The best example of this is user accounts which need to be more or less local for fast login.

Replicated

Sections of the NDS database can be reproduced on multiple servers. This means that if one server crashes NDS is able to keep working.

Extensible

The definition of what can be stored in NDS is held in the **schema.** The schema itself can be added to and hence what can be stored in NDS can be, and is, constantly adapted. Given a distributed, and highly secure, database there are lots of great ideas around for ways to use it.



NDS replication

- Like X.500 NDS can be duplicated on numerous servers in a tree
- Advantages
  - Fault tolerance
  - Accessibility for users
- Peer-peer replication
  - Any replica can be changed

There is one important difference between X.500 and NDS replication.

X.500 uses a Master/Slave replication model - the Master replica is the only one that can be changed and these changes then flow outwards from the Master to the Slaves.

NDS uses a peer-peer model. Any replica (except one that has been declared read-only) can be changed and the changes flow outwards from the altered replica to its peers.

Synchronisation

- With replication comes the cost of maintaining consistent copies of NDS.
- All changes are synchronised across the all replicas
- Only changes are sent
  - Unlike Microsoft domain databases

The act of replication in NDS is referred to as **synchronisation.** Periodically (or immediately, depending on the importance of the change) the changed data is sent by each server that has been changed to the other servers that are storing replicas of this particular data.

Only the changes are sent - the granularity is at the level of values of a particular property for a particular object. If I change your phone number in the NDS tree that is all that is sent to the replicas.