

CS 5523 Lecture 28: Directory and Discovery Services

- *Questions on Laboratory 4*
- *Directory services*
- *X.500*
- *LDAP development*
- *Discovery services*
- *Jini case study*

Directory services:

- *Stores binding of name to attribute*
- *Allows lookup based on attributes*
- *Sometimes directory services are called “yellow pages services”, while name services are called “white pages services”*
- *Often organized in a hierarchy*
- *Examples of directory services:*
 1. *Microsoft’s Active Directory Services*
 2. *X.500*

Give some examples of applications of directory services

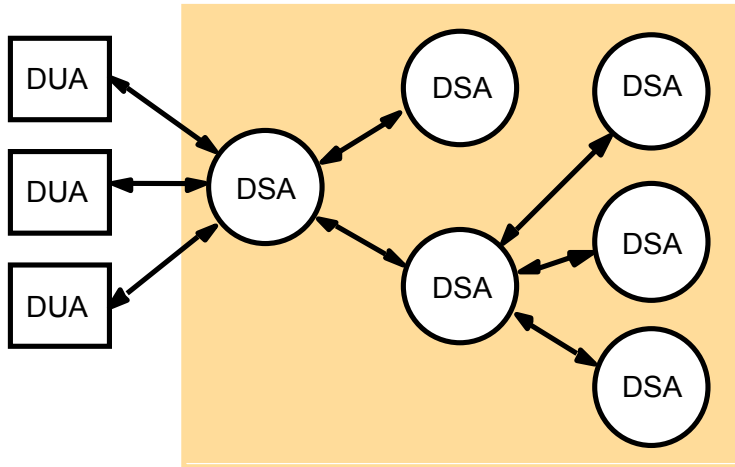
X.500 directory service:

- *Standardized by ITU and ISO*
- *Specified as an application-level in OSI*
- *Data is organized in tree (DIT =directory information tree) with named nodes (DIB=directory information base)*
- *A client (DUA = directory user agent) can query any server (DSA = directory service agent)*
- *Server will respond, query other services or send back server response*
- *A DIB entry has a name and a set of attributes.*
- *The full name is the fully-qualified path*

X.500 directory service (continued):

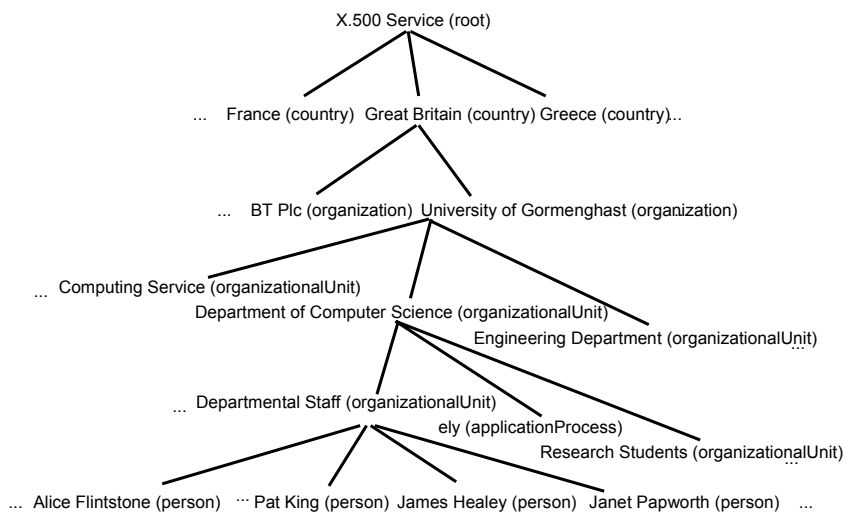
- *Directory Administrative Model – how global DIT is managed and split into domains*
- *Directory Server Protocol (DSP) – used to chain user requests between directory servers*
- *Directory Information Shading Protocol (DISP) – protocol for directory replication*
- *DOP protocol to automate connection agreements between servers between and across management domains*

Figure 9.10
X.500 service architecture



Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 3
© Addison-Wesley Publishers 2000

Figure 9.11
Part of the X.500 Directory Information Tree



Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 3
© Addison-Wesley Publishers 2000

Figure 9.12
An X.500 DIB Entry

info
Alice Flintstone, Departmental Staff, Department of Computer Science,
University of Gormenghast, GB

<i>commonName</i>	<i>uid</i>
Alice.L.Flintstone	alf
Alice.Flintstone	
Alice Flintstone	<i>mail</i>
A. Flintstone	alf@dcs.gormenghast.ac.uk
<i>surname</i>	Alice.Flintstone@dcs.gormenghast.ac.uk
Flintstone	<i>roomNumber</i>
<i>telephoneNumber</i>	Z42
+44 986 33 4604	<i>userClass</i>
	Research Fellow

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 3
© Addison-Wesley Publishers 2000

X.500 data structures:

- *Attributes are typed (e.g. countryName, commonName)*
- *DIB entries are organized like OO classes.*
- *DIB entries have a class-name.*
- *The definition of a class determines which attributes are mandatory and which are optional.*
- *Mandatory and optional attributes are inherited.*

X.500 operations:

- *Read – locates attributes associated with given name*
- *Search – finds records based on attributes and filters*
- *DSA also has operations for adding, deleting and modifying entries*

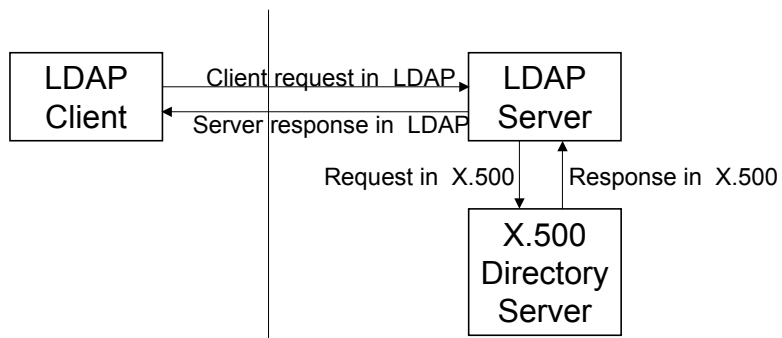
X.500 history and deployment:

- *Specification was first released in 1988 with a significant update in 1993*
- *Heavyweight system:*
 - *Required a full OSI protocol stack that wasn't supported by MACs or PCs in the early 90's when X.500 was being deployed*
 - *The DUAs (directory user agents) also could not be run on PCs or MACs in the early 90's*
- *University of Michigan responded with the development of a DUA that understood a lightweight access protocol called LDAP*

Lightweight Directory Access Protocol (LDAP):

- *LDAP runs over TCP/IP*
- *Simple DUA interface to X.500*
- *Uses textual encoding*
- *Provides secure access through authentication*
- *Other directory services have implemented it*
- *See RFC 2251 [Wahl et al. 1997]*

LDAP strategy:



LDAP evolution:

- *University of Michigan added to LDAP servers the capability of accessing own database.*
- *Use of LDAP databases became widespread*
- *Schemes were developed for registering changes and exchanging deltas between LDAP servers*
- *In 1996 three engineers from U of Michigan joined Netscape. 40 companies (sans Microsoft) announced support of LDAP as the standard for directory services*
- *Core specifications for LDAPv3 was published as IETF RFCs 2251-2256.*

LDAP protocol overview:

- *Strict subset of X.500*
- *One or more servers jointly provide access to a DIT of entries*
- *Entries have:*
 - *A relative distinguished name (RDN) unique among its siblings*
 - *A distinguished name (DN) concatenation of RDNs up to root.*
- *Naming context – largest collection of entries starting at an entry that is mastered by a particular server.*
- *Entries consist of a set of attributes (type + values). Entries also have to have an objectClass attribute*
- *Type consists of a short name and an OID and what kind, e.g. attribute mail must have case insensitive ascii strings as values.*

Discovery services:

- *Directory service that registers services in a spontaneous network*
- *Services notify the discovery service of existence via a registration interface*
- *Discovery has a scope (the context)*
- *Example - Jini (a Java-based system for spontaneous networking)*

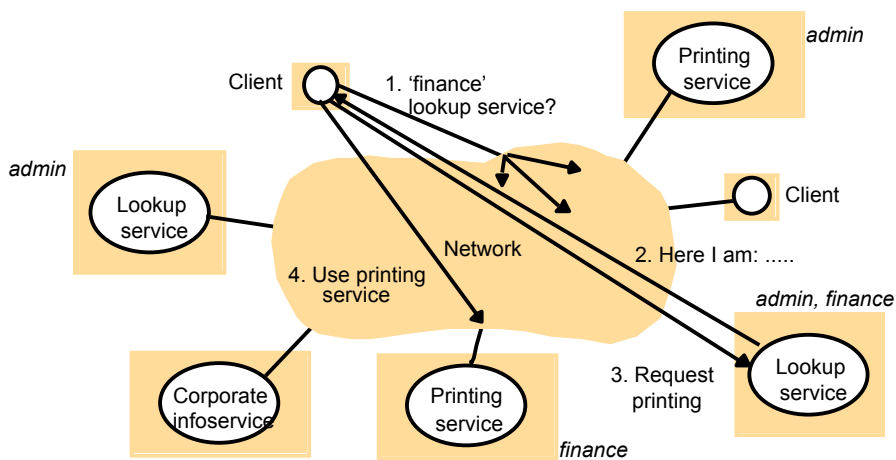
Jini – an example of a discovery service:

- *Java-based – all devices have a Java virtual machine and communicate by RMI to download code*
- *Provides facilities for discovery, transactions and shared data*
- *Targeted for ad-hoc networking at the workgroup level*
- *Based on the notion of services*
- *Examples of services:*
 - *Computation*
 - *Communication channel*
 - *Storage*
 - *Software filter*
 - *hardware*

Jini protocols:

- *Discovery* – service looks for a lookup service to register with
- *Join* – service has located a lookup service and registers with it
- *Lookup* – client wants to locate a service

Figure 9.6
Service discovery in Jini



Jini programming model:

- *Leasing interface – defines method of allocating and freeing resources using a renewable, duration-based model*
- *Event and notification interface – enables event-based communication between services*
- *Transaction interface – enables entities to cooperate so that changes are made atomically to everyone or none at all*

Jini client-service interaction:

- *Service provides a set of interfaces for interaction*
- *Service may provide a “user interface” that allows a human to interact directly*
- *Interfaces may use:*
 - *RMI (remote invocation interfaces)*
 - *Local computation*
 - *Smart proxy (combination of local and remote execution)*

For next time:

- *Be prepared to discuss CDK questions 9.2, 9.3, 9.4, 9.8, 9.9, 9.12 and 9.16*
- *Also questions from Chapter 8*