LDAP - concepts, applications, practical problems

Chaos Communication Camp,

Paulshof, Altlandsberg, 9. August 2003

Peter Gietz

peter@daasi.de

Directory in German Research environment

- Since 1994 DFN research projects at University of Tübingen:
  - AMBIX - an Email directory
  - DFN Directory Services (DDS)
    - Directory competence center
- Since January 2001: DAASI International GmbH
  - Directory Applications for Advanced Security and Information Management
  - Design, implementation and management of directory services
  - Main Customers: Research Institutions in Europe (NRNs, Universities, etc.)

Agenda

- Introduction to LDAP
  - What is a Directory
  - LDAP heritage: X.500 and history of LDAP
  - Information model
  - Operational model
  - LDAP security
  - Open Source implementation OpenLDAP
- Applications of LDAP technology
  - White and Yellow Pages
  - Central authentication service
  - Unified login with OpenLDAP and Samba
  - Unified password with LDAP enabled applications
  - Single Sign On with Kerberos

Agenda (contd.)

- LDAP and X.509 based Public Key Infrastructure
- LDAP and Information management
- Metadata and Ontologies, or LDAP and the Semantic Web
- LDAP and Ressource management, in Grid Computing and elsewhere
- LDAP and Directory Enabled Networking

➢ The future of LDAP - is it XML?

What is a Directory?

What is a Directory?

- Information stored in a hierarchical System
- Examples:
  - File directory of an operating system (MS/DOS, Unix)
  - Domain Name Service (DNS)
  - Network Information System (NIS)
  - X.500 is *the* Directory
  - Lightweight Directory Access Protocol (LDAP)
  - Novell Directory Service (NDS)
  - Microsoft Active Directory (AD)

So what really is *the* Directory

- Concept of on world wide Directory
- It is a sort of a database
  - for storing and retrieving information
- It is a specialized database
  - designed for fast reading, writing is slower
  - simple updates without transactions
- It has a dedicated network protocol for access
- A Directory Service may include
  - distribution in the net
  - replication of the data

What kind of data can you store?

- Text data
  - names, addresses, descriptions, numbers, etc.
- Pointers
  - URLs, pointers to other data, etc.
- Public key certificates
- Graphics
  - photos, diagrams, etc.
- Other binary data
- Anything else you can think of
- Most usefull for:
  - Information about real world objects
  - “Metadata” Data about data

LDAP heritage: X.500 and history of LDAP

X.500 the heritage of LDAP

- Standard of ITU / ISO
- Part of OSI (Open Systems Interconnection)
  - backdraws:
    - theoretical
    - complex
    - little acceptance
  - advantages:
    - conforming to OSI
    - good concept
    - modern design

Standardization boards

- ISO
  - International Standards Organization
  - Name of the Directory standard: ISO 9594
- CCITT
  - Comitée Consultative International Telephonique et Telegraphique
  - The former international board for Telecommunication Organizations
  - Name of the same standard: X.500
- ITU
  - International Telecommunications Union
  - The successor of CCITT

History of the X.500 standard

- 1984 start of efforts for defining a standard for distributed data in the net
- 1988 first version of the standard (X.500v1)
  - X.509 includes authentication based on asymmetric encryption
  - Undefined access control and replication
  - proprietary replication mechanism in first implementation Quipu from the ISODE Consortium
- 1993 second version (X.500v2)
  - includes the missing bits:
    - Replication called shadowing
    - access control

History contd.

- 1997 third version (X.500v3)
  - includes enhanced definitions for certificates in X.509v3: Extensions
- 2001 fourth version (X.500v4)
  - X.509v4 adds Attribute Certificate and Privilege Management Infrastructure

Parts of the X.500 Standard

- X.500 - Overview of concepts, models and services
- X.501 - Models
- X.509 - Authentication framework
- X.511 - Abstract service definition
- X.518 - Procedures for distributed operation
- X.519 - Protocol specifications
- X.520 - Selected attribute types
- X.521 - Selected object classes
- X.525 - Replication
- X.530 - Use of system management for administration of the Directory

What was X.500 originally intended for?

- To give humans information like
  - Data (telephonenumbers etc.) about humans (White Pages)
  - Data (postal address etc.) about organisations (Yellow Pages)
- To give applications data in a known format for
  - Message handling
  - File transfer (File Transfer Access Management, FTAM)
  - Name mapping for OSI
- The Standard defines a set of data fields for these purposes

X.500 Client Server model

- Directory Service Agent (DSA)
  - A Server that holds directory information
- Directory User Agent (DUA)
  - A client that connects to a DSA to access information
- The DUA and DSA communicate via an access protocol
- The X.500 client-server access protocol is called Directory Access Protocol DAP
- A lightweight version of DAP is LDAP **L**ightweight **D**irectory **A**ccess **P**rotocol

LDAP and X.500

- Qualities of X.500
  - Any amount of data can be stored
  - On any number of servers
  - Clients need to connect to only one server (chaining)
  - Data look the same everywhere
  - Open model for any kind of data
- LDAP took over all the good stuff:
  - Information model (ASCII-based)
  - Client-Server model (without the chaining)
  - But it is TCP/IP based and thus “Lightweight”

History of LDAP: LDAP v1

- A group at University of Michigan developed a Lightweight Version of DAP
  - No OSI Stack
  - Directly over TCP
  - Only DUA - DSA communication
  - Most protocol data elements ordinary strings
  - Easier to implement
  - better performance
- First Implementation was called DIXIE
- LDAPv1 was never published as IETF RFC

1993: LDAP v2

- RFC 1487:
  - X.500 Lightweight Directory Access Protocol, W. Yeong, T. Howes, S. Hardcastle-Kille. July 1993
- RFC 1488:
  - The X.500 String Representation of Standard Attribute Syntaxes. T. Howes, S. Kille, W. Yeong, & C. Robbins. July 1993
- RFC 1558:
  - A String Representation of LDAP Search Filters. T. Howes. December 1993

1995: LDAP v2 (Draft Standard)

- RFC 1777:
  - Lightweight Directory Access Protocol, W. Yeong, T. Howes & S. Kille. March 1995
- RFC 1778:
  - The String Representation of Standard Attribute Syntaxes, T. Howes, S. Kille, W. Yeong & C. Robbins. March 1995
- RFC 1798:
  - Connection-less Lightweight Directory Access Protocol, A, Young. July 1995
- RFC 1823:
  - The LDAP Application Program interface, T. Howes & M. Smith. August 1995

1997: LDAP v3 (Proposed Standard)

- RFC 2251:
  - Lightweight Directory Access Protocol (v3), M. Wahl, T. Howes, S. Kille. December 1997
- RFC 2252:
  - Lightweight Directory Access Protocol (v3) - Attribute Syntax Definitions, M. Wahl, A. Coulbeck, T. Howes, S. Kille. December 1997
- RFC 2253:
  - Lightweight Directory Access Protocol (v3) - UTF-8 String Representation of Distinguished Names, M. Wahl, S. Kille, T. Howes. December 1997
- RFC 2254:
  - The String Representation of LDAP Search Filters, T. Howes. December 1997

1997 LDAPv3 contd.

- RFC 2255:
  - The LDAP URL Format, T. Howes, M. Smith. December 1997
- RFC 2256:
  - A Summary of the X.500(96) User Schema for use with LDAPv3, M. Wahl. December 1997
- RFC2829:
  - Authentication Methods for LDAP
- RFC2830:
  - Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
- RFC 3377:
  - Lightweight Directory Access Protocol (v3): Technical Specification

IETF WG LDAPbis

- Revision of all LDAP core RFCs
- With references to mandatory security mechanism of RFC 2829 and 2830 possible to go for Draft Standard
- No changes in the data definitions
- Some clarifications in wording
- Some SHOULDS to MUST etc.

Current LDAPbis Drafts

- draft-ietf-ldapbis-protocol-16 obsoletes RFC 2251 and portions of RFC 2252
- draft-ietf-ldapbis-models-08 obsoletes portions of RFC 2251, 2252 and 2256
- draft-ietf-ldapbis-syntaxes-06 obsoletes RFC 2252 and portions of 2256
- draft-ietf-ldapbis-dn-11 obsoletes RFC 2253
- draft-ietf-ldapbis-filter-04 obsoletes RFC 2254
- draft-ietf-ldapbis-url-03 obsoletes RFC 2255
- draft-ietf-ldapbis-user-schema-06 obsoletes RFC 2256
- draft-ietf-ldapbis-authmeth-06 obsoletes RFC 2829 and 2830
- draft-ietf-ldapbis-roadmap-03 obsoletes RFC 3377
- draft-ietf-ldapbis-strprep-01 Unicode character string matching
- draft-ietf-ldapbis-bcp64-00.txt obsoltes RFC 3383 (IANA consid.)

LDAP Features

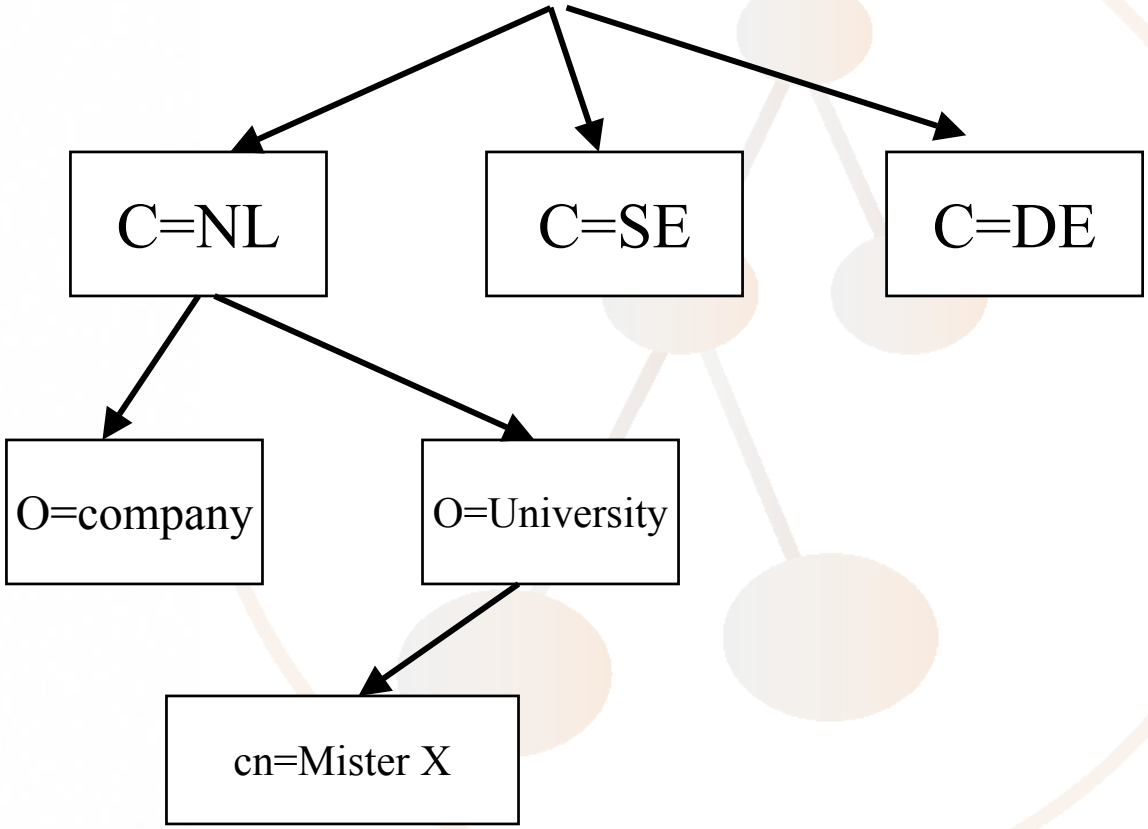
- The LDAP standard defines...
  - a network protocol for accessing information in the directory
  - an information model defining the form and character of the information
  - a namespace defining how information is referenced and organized
  - secure authentication mechanisms
  - an emerging distributed operation model defining how data may be distributed and referenced (v3)
  - Both the protocol itself and the information model are extensible
  - (de facto standard) C API and Java API

LDAP/X.500 information model

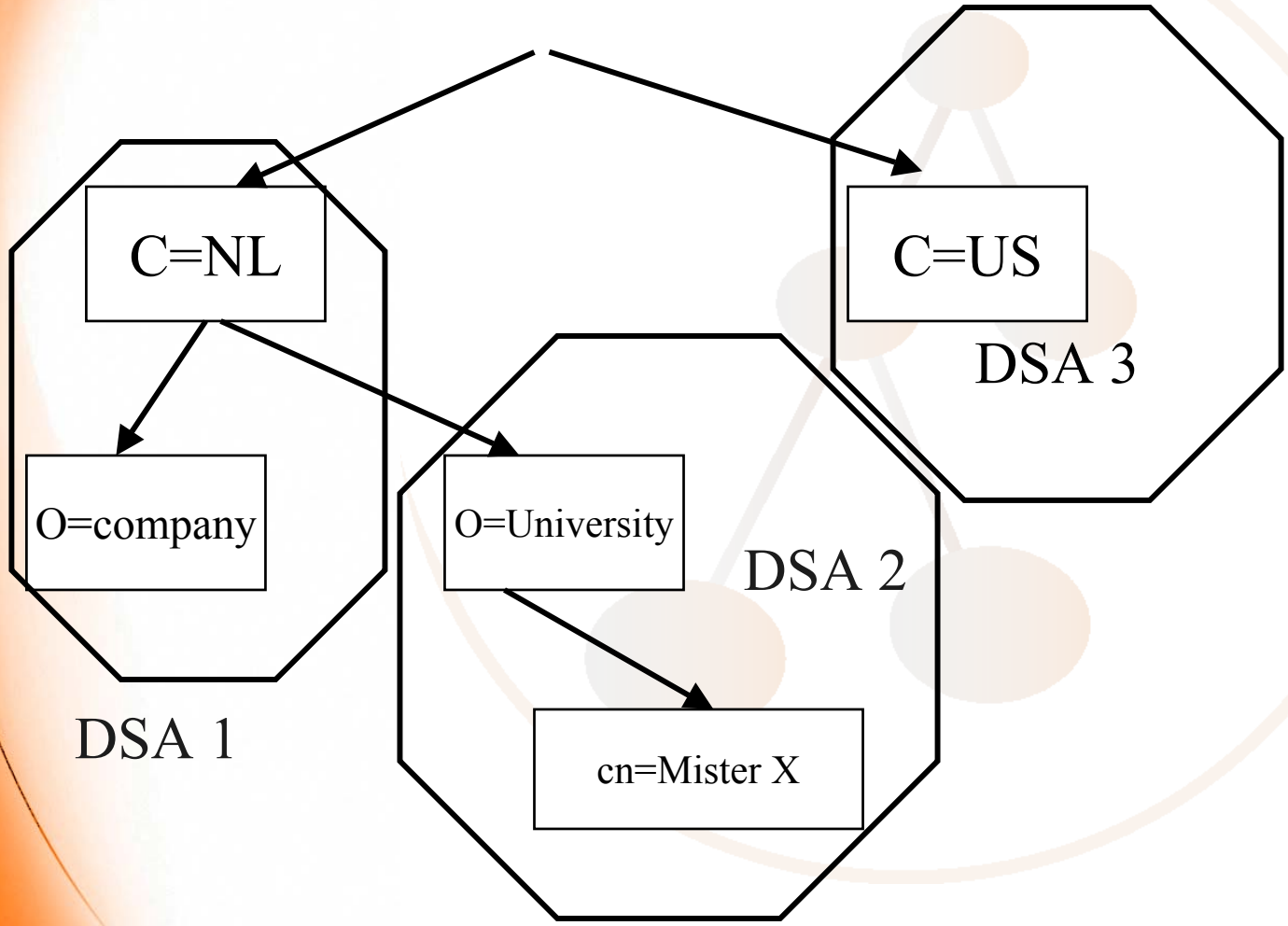
X.500/LDAP Information Tree

- Data are stored in entries
- Entries are ordered as tree nodes
- In the Directory Information Tree (DIT)
  - Every node has 0 to n children nodes
  - Every node except root has 1 parent node

Directory Information Tree (DIT)



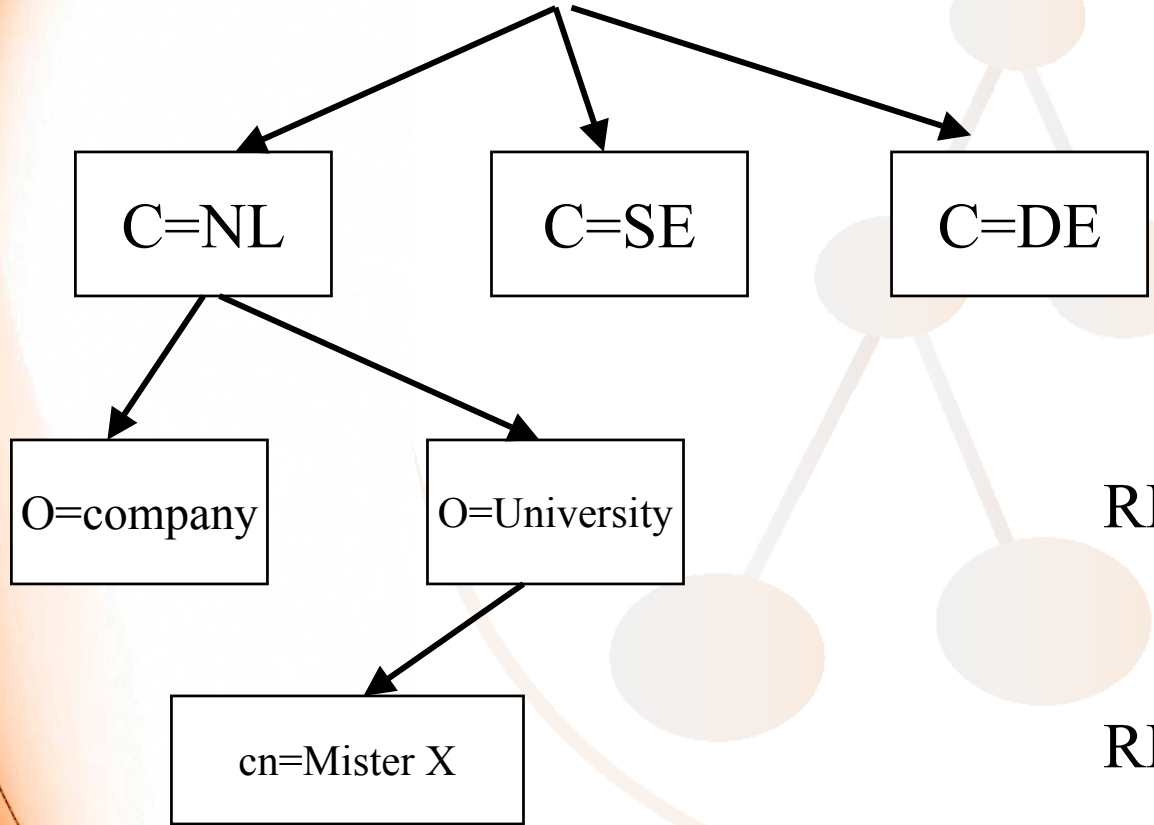
Distribution of the data among Directory Service Agents (DSA)



DN Distinguished Name

- An entry has a distinguished name
  - in its hierarchy level: Relative Distinguished Name (RDN)
  - all RDNs on the path from root form the Distinguished Name (DN)
- No two siblings, i.e. entries with a common parent can have the same RDN
- Thus no two entries in the whole Directory can have the same DN

Relative Distinguished Name (RDN) and Distinguished Name (DN)



RDN: C=NL

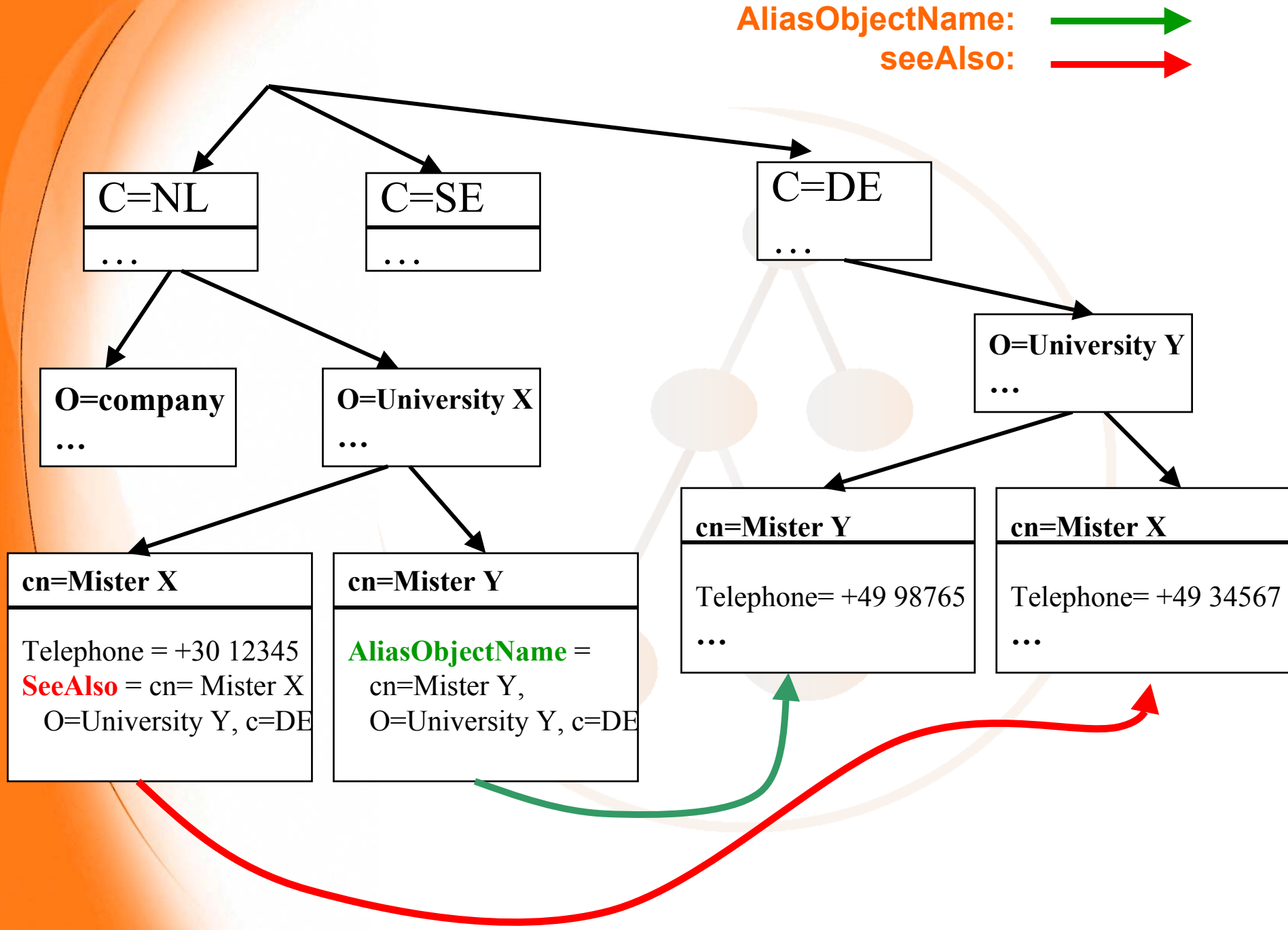
RDN: o=University

RDN: cn=Mister X

DN: c=NL;o=University;cn=Mister X

DN Pointer

- Alias Entries have a DN and point to another DN via aliasObjectName Attribute
- seeAlso Attribute: Entry contains data and a seeAlso pointer to another DN


AliasObjectName:
seeAlso:
C=NL
…
C=SE
…
C=DE
…
O=company
…
O=University X
…
O=University Y
…
cn=Mister X
Telephone = +30 12345
SeeAlso = cn= Mister X
O=University Y, c=DE
cn=Mister Y
AliasObjectName =
cn=Mister Y,
O=University Y, c=DE
cn=Mister Y
Telephone= +49 98765
…
cn=Mister X
Telephone= +49 34567
…

LDAP Naming Model

- Just like X.500:
  - RDN and DN
  - DIT
  - Alias and seeAlso
- Differences:
  - String representation of DNs
  - Alternative to X.520 naming: Domain componant (DC)
    - X.520: cn=Mister X, o=University, c=NL
    - DC: uid=Misterx1, dc=Uni, dc=NL
    - advantage: registration problems are handled by DNS
  - There is no single international DIT

How is the information stored?

- An Entry is an information object
- The mechanisms for representing and describing the data (e.g. value syntax) are objects as well, identified by an OID (Object Identifier)
- OIDs are again represented in an hierarchical tree

OID-Tree

- E.g.: Subtree maintained by DAASI International:
  - Daasi = 1.3.6.1.4.1.10126
  - For more see: http://www.alvestrand.no/objectid/
  - On 1.3.6.1.4.1. See also http://www.iana.org/assignments/enterprise-numbers
  - By now ca. 14.000 Enterprise-numbers have been assigned

X.500/LDAP Information Model

- An Entry is a collection of Attributes
- An Attribute consists of:
  - Attribute Type
  - Attribute Value(s)
- An Attribute Type has an associated Attribute Syntax
- The Attribute Value has to conform to that syntax
- Matching Rules to compare Attribute values for
  - equality
  - substring
  - ordering
  - extensible (selfdefined) matching

Special Attributes

- One or more Attribute type/value pairs form the RDN
  - The Naming Attributes or
  - The Distinguished Attributes
- An Entry must have one or more Objectclass Attributes which:
  - Characterizes the Entry, e.g. Person
  - Defines a set of usable Attributes the entry may contain and must contain
- A set of Objectclasses, Attributes and Syntaxes for a special purpose is called schema

Objectclass inheritance

- Objectclasses can inherit Attributes from other Objectclasses
- One Objectclass can thus be superclass of another
- The subclass inherites all attribute definitions of the superclass. E.g.:
  - Objectclass person includes attribute surname. Etc.
  - organizationalPerson inherits attributes of person and adds new attributes like RoomNumber, etc.

Objectclass Types 1

- ABSTRACT Objectclasses
  - Are only used for base of inheritance
  - No entry can be instanciated with Abstract Object classes
- STRUCTURAL Objectclass
  - These describe a whole thing
  - Represent an entity
  - E.g.: Person, Organisation, etc
  - Every entry may only have one structural objectclass (together with it‘s inheritance descendence, e.g. person and organizationalPerson)

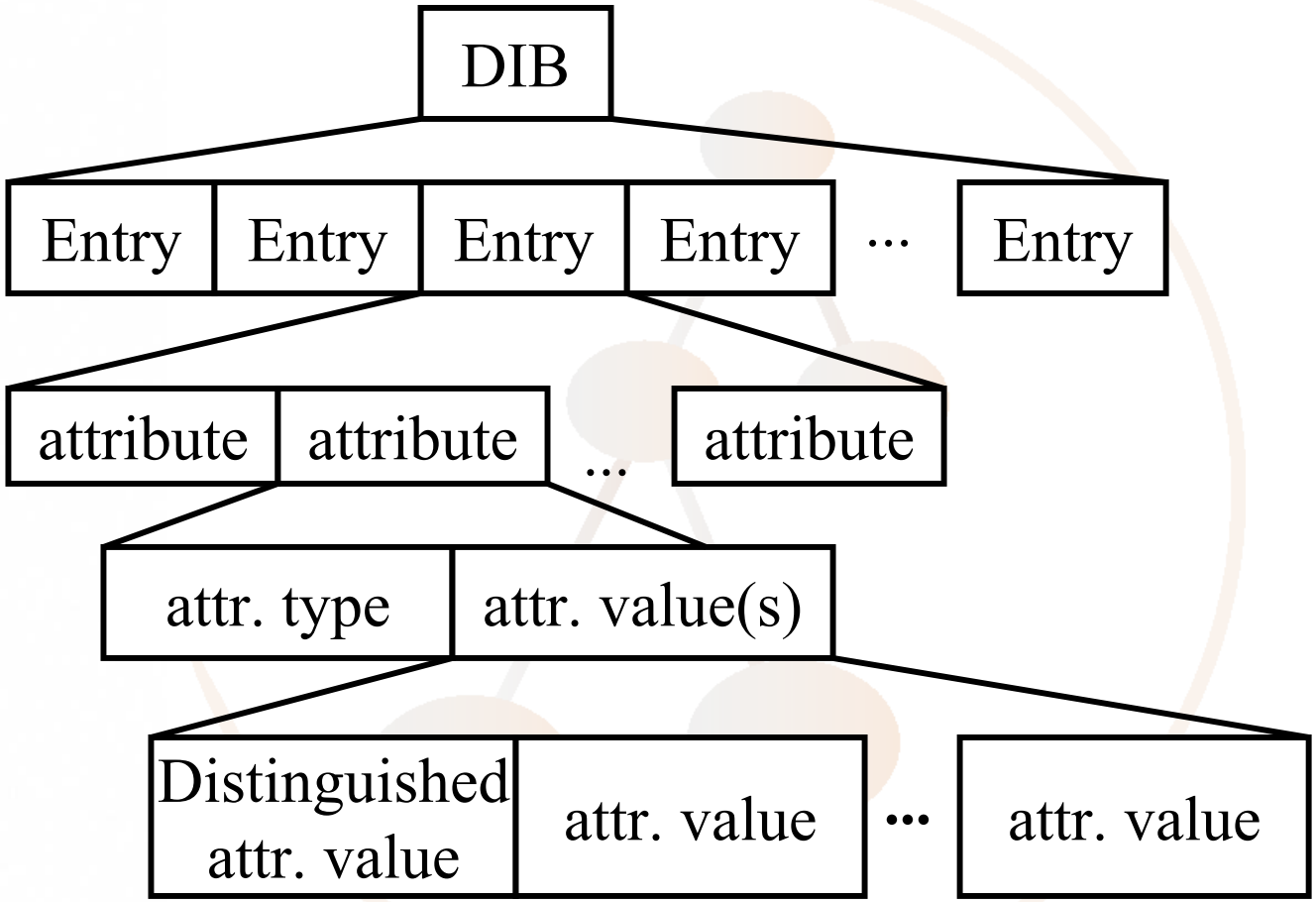
Objectclass Types 2

- AUXILIARY
  - These describe single additional aspects of an entity
  - Different kinds of entities can have common aspects
  - You can add as many AUX classes to an entry as you want
  - E.g.: PKIuser includes the attribute certificate. A person can have a certificate, but a server as well
  - Another example: labeledUriObject, with attribute labeledURI.

Attribute inheritance

- Attributes can also stand in an inheritance hierarchy, E.g.:
  - name -> common name
    -> surname
  - telephone number -> home number
    -> office number
- If you request the more general attribute you will get all more specific attributes

Directory Information Base



Example of an entry:

DN: cn=Mister X, o=University, c=NL

ABSTRACT →	**objectClass=top**
STRUCTURAL (only one inheritance chain) →	**objectClass=person** **objectClass=organizationalPerson** **objectClass=inetOrgPerson**
AUXILIARY →	**objectClass=pKIUser**
cn from person →	**cn=Mister X** **cn=Xavier Xerxes**
Mail from inetOrgperson →	**mail=X@dot.com** **mail=Mister.X@dot.com**
telephoneNumber from organizationalPerson →	**telephoneNumber=1234567**
userCertificate from pKIUser →	**userCertificate=a1b2c3d4e5f6…**

LDAP Data Interchange Format LDIF

- RFC 2849:
  - The LDAP Data Interchange Format (LDIF) - Technical Specification, G. Good, June 2000
- Format for exchanging data
- Example:

```
dn: cn=Mister X, o=University, c=NL
Objectclass: top
Objectclass: person
Objectclass: organizationalPerson
Cn: Mister X
Cn: Xavier Xerxes
Mail: X@dot.com
Mail: Mister.X@dot.com
telephoneNumber: 1234567

dn: cn=next entry, ...
```

Some Objectclasses

ObjectClass	distinguished Attr. and abbreviation	other Attributes
country	countryName or c	description, searchGuide, ...
locality	localityName or l	description, ...
organization	organizationName or o	description, postalAdress, ...
organizational Unit	organizationalUnit -Name or ou	description, postalAdress, ...
person	commonName or cn	surname, title, ...

Collective Attributes

- Sort of attribute value inheritance
- Attributetype-value pair that exists virtually in every entry of a subtree
- Example:
  - All persons in an organizational unit use the same Faxnumber
  - Define a collective attribute at the organizational unit level and it will appear in everey entry down the tree

Open structure of LDAP/X.500 information model

- You can define without modifying the implementation:
  - Object Classes
  - Attribute Types
- You can define (but has also to be implemented in the servers)
  - Attribute Syntaxes
  - Matching Rules
- You can locally use self defined schemas
- If you want them to be used globally you have to
  - standardize them (IETF)
  - or at least register them at Directory Schema Registry soon operational at www.schemareg.org

Attribute definition (RFC2252)

```
AttributeTypeDescription = "(" whsp numericoid whsp
                                  ; AttributeType identifier
   [ "NAME" qdescrs ]   ; name used in AttributeType
   [ "DESC" qdstring ]  ; description
   [ "OBSOLETE" whsp ]
   [ "SUP" woid ]       ; derived from this other AttributeType
   [ "EQUALITY" woid    ; Matching Rule name
   [ "ORDERING" woid    ; Matching Rule name
   [ "SUBSTR" woid ]    ; Matching Rule name
   [ "SYNTAX" whsp noidlen whsp ] ; -> sect. 4.3
   [ "SINGLE-VALUE" whsp ] ; default multi-valued
   [ "COLLECTIVE" whsp ]     ; default not collective
   [ "NO-USER-MODIFICATION" whsp ]; default user modifiable
   [ "USAGE" whsp AttributeUsage ]
                                  ; default userApplications whsp ")"
```

Attribute definition contd.

AttributeUsage =
"userApplications" / ; for normal data
"directoryOperation" / ; for server modifiable data
"distributedOperation" / ; for server-shared data
"dSAOperation" ; server-specific, value depends on
server

Attribute definition contd.

```
oid = descr / numericoid
descr = keystring
numericoid = numericstring *( ". "  numericstring )
oids = woid / ( "(" oidlist ")" )
woid = whsp oid whsp ; set of oids of either form
oidlist = woid *( "$" woid ) ; object descriptors used
            as schema element names
qdescrs = qdescr / ( whsp "(" qdescrlist ")" whsp )
qdescrlist = [ qdescr *( qdescr ) ]
```

Attributdefinition example

(2.5.18.2
NAME 'modifyTimestamp'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE
NO-USER-MODIFICATION
USAGE directoryOperation)

[Generalized Time 1.3.6.1.4.1.1466.115.121.1.24]

Objectclass definition

ObjectClassDescription =
"(" whsp numericoid whsp ; ObjectClass identifier
["NAME" qdescrs]
["DESC" qdstring]
["OBSOLETE" whsp]
["SUP" oids] ; Superior ObjectClasses
[("ABSTRACT" / "STRUCTURAL" / "AUXILIARY")
whsp] ; default structural
["MUST" oids] ; AttributeTypes
["MAY" oids] ; AttributeTypes whsp ")"

OC Definition examples

- (2.5.6.0 NAME 'top' ABSTRACT MUST objectClass)
- (2.5.6.6 NAME 'person' SUP top
  STRUCTURAL
  MUST (sn $ cn)
  MAY (userPassword $ telephoneNumber $ seeAlso $
  description))

OC Definition examples

- (2.5.6.7 NAME 'organizationalPerson'
  SUP person STRUCTURAL
  MAY (title $ x121Address $ registeredAddress $
  destinationIndicator $ preferredDeliveryMethod $
  telexNumber $ teletexTerminalIdentifier $
  telephoneNumber $ internationaliSDNNumber $
  facsimileTelephoneNumber $ street $ postOfficeBox $
  postalCode $ postalAddress $
  physicalDeliveryOfficeName $ ou $ st $ l))

Standardized Schema

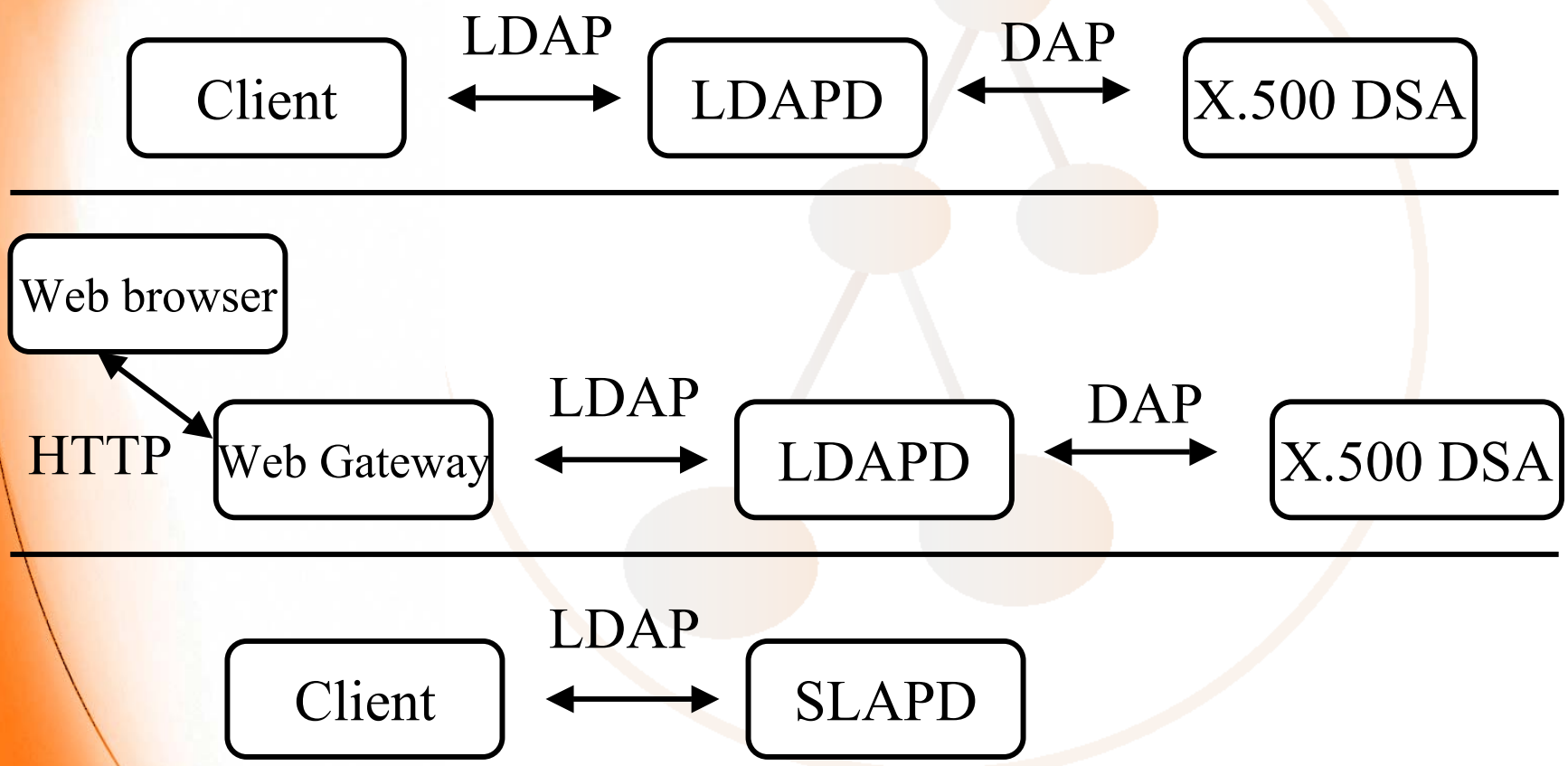
- Schema allready standardized in the core specifications see RFC 2256 (the old X.500 standard schema)

LDAP operational model

Who talks LDAP?

- Originally (v1,v2) just a client access protocol for X.500
- LDAP v3 is a whole client server system
- All directory implementations have an LDAP interface:
  - all X.500(93) implementations
  - Novell Directory Service (NDS)
  - Microsoft Active Directory (AD)
- Many client applications have an LDAP interface:
  - mailagents
  - browsers
  - PGP clients

LDAP connectivity



LDAP Functional Model

- Authentication and control operations:
  - bind
  - unbind
  - abandon
- Interrogation operations:
  - search
  - compare
- Update operations:
  - add
  - delete
  - modify
  - modifyDN

LDAP Search Parameters

1. base object or base DN
   - where in the DIT the search starts
2. scope
   - base (read the entry specified by the base dn)
   - onelevel (search only in the hierarchical level of the basedn)
   - subtree (search in level of base DN and below)
3. derefAliases
   - neverDerefAlias (do not dereference aliases in searching or in locating base object)
   - derefInSearching (dereference only in subordinates of base object)
   - derefFindingBaseObject (dereference only in locating the base object)
   - derefAlways (dereference aliases in searching subordinates and in locationg base object)

LDAP Search Parameters contd.

4. size limit
   - limit the number of entries to get back
5. time limit
   - limit the time the server should spend to fulfil the request
6. attrsOnly
   - Boolean. If set to true only the attributenames will be sent back, not the values
7. Filter
   - expression that describes the entries to be returned

LDAP Search Parameters contd.

8. attributes
   - a list of comma separated attributes Types to be returned
   - e.g.: cn, telephonenumber
   - can be specified by OID as well, e.g. 2.5.4.3, 2.5.4.20
   - * means all user attributes
   - 1.1 (there is no such attribute OID) for no attributes

Search Filter Operators

- Equality
  - Only for attributes with equality matching rule
  - e.g.: (cn=Mister X) only entries with common name equals “Mister X”
- Substring
  - Only for attributes with substring matching rule
  - e.g. (cn=Mister*) all entries with cn beginning with “Mister”
- Approximate
  - Implementation dependent
  - e.g.: (cn~=Mister) all entries with cn sounding similiar to “Mister”
- Negation operator
  - e.g. (!(cn=Mister X)) all entries but the one with cn equals “Mister X”

Search Filter Operators (contd.)

- Greater than or equal to and less than or equal to
  - Only for attributes with ordering matching rule
  - e.g. (sn<=Smith) all entries where sn equals “Smith” or is lexicographically above “Smith” (from sn=Adam to sn=smirnow)
  - (age>21) is not possible, use (!(age<=21)) instead
- Presence
  - e.g. (telephoneNumber=*) all entries that contain a telephone number
  - e.g. (objectclass=*) all entries, since every entry contains at least one objectclass

Search Filter Extensions

- LDAPv3 defines an extensible matching filter
  - syntax: attr [“:dn”] [“:” matchingrule] “:=“ value
    - attr is an attribute name
    - “:dn” says that also the attribute in the dn should be searched as well
    - matching rule given by an OID or associated descriptive name
  - examples:
    - (cn:1.2.3.4.5.6:=Mister X) use matching rule 1.2.3.4.5.6 for comparison
    - (o:dn:=company) search for o=company in attributes and also in DN

Search filter combinations

- Filters can be combined
  - AND operator: &
    - e.g. (& (cn=Mister X) (mail=*dot.com)) only entries that have both cn=Mister X and a mail address ending with dot.com
  - OR operator: |
    - e.g.: (| (cn=Mister X) (sn=Xerxes)) all entries that have cn=Mister X or sn=Xerxes

Search filter special characters

- Five characters have special meaning
  - must be replaced by an hexadecimal escape sequence if you want to search for them:
  - ´*´ (dec. 42, hex 0x2A) must be replaced with : ´\2a´
  - ´(´ (dec. 40, hex 0x28) must be replaced with : ´\28´
  - ´)´ (dec. 41, hex 0x29) must be replaced with : ´\29´
  - ´\´ (dec. 92, hex 0x5C) must be replaced with : ´\5c´
  - NUL (dec. 0, hex 0x00) must be replaced with : ´\00´
- Example
  - value “A*Star” must be written,
    e.g. (cn=A\2AStar)

LDAP URL (RFC 2255)

- Format:
  - ldap://<host>:<portnumber>/<basedn>?<attrlist>?<scope>?<filter>?<extensions>
- Example:
  - ldap://myhost.org:9999/o=University,c=NL?cn,telephonenumber?subtree?(cn=Mister X)

LDAPv3 Extension mechanisms

- LDAP controls
  - RFC 2251, Par. 4.1.12
  - All 9 LDAP operation (bind, search, add, ...) can be extended
  - controls modify behavior of operation
  - consist of controlType, criticality, [controlValue]
  - client and server must support the control

LDAPv3 Extension mechanisms contd.

- LDAP extended operations
  - RFC 2251, Par. 4.12
  - new defined protocol operation in addition to the nine
  - ExtendedRequest: requestName, [requestValue]
  - ExtendedResponse: LDAPresult,[responseName, response]
- SASL mechanisms
  - Framing for support of different authentication mechanisms

Root DSE Entry

- a special entry in the LDAP server
- contains attributes that describe the server:
  - namingContext (which part of the DIT)
  - subschemaSubentry (supported schema)
  - altServer (alternate Server that should contain the same data)
  - supportedLDAPVersion
- has attributes that describe which extensions are supported:
  - supportedExtensions
  - supportedControls
  - supportedSASLMechanisms
- Retrieve the data e.g. by
  - ldapsearch –x –b “” –s base +

RFC 2589

- LDAPv3: Extensions for Dynamic Directory Services, Y. Yaacovi, M. Wahl, T. Genovese. May 1999 (STD)
  - Dynamic entries in the directory
  - periodical refreshing of the information
  - needed, e.g. for person online status information while a video conference
  - Client and server requirements

RFC 2589 contd.

- Defines:
  - ExtendedRequest:
    - RequestName (OID), entryName (DN), requestTtl (Time to live in seconds)
  - ExtendedResponse:
    - LDAPresult enhanced by responseName and responseTtl (Time to live in seconds, may be larger than requested)
  - Objectclass dynamicObject with Attr. EntryTtl
  - RootDSE Attribute:
    - dynamicSubentries

RFC 2596

- Use of Language Codes in LDAP, M. Wahl, T. Howes. May 1999 (STD)
  - uses Attribute tag mechanism:AttributeDescription
  - language codes as in RFC 1766
  - Format: <Attr.>;lang-<language code>
  - Example: givenName; lang-en-US
  - is not allowed in DN
  - allowed in:
    - search filter, e.g. (cn;lang-en=X*)
    - compare request
    - requested attribute, e.g. ldap://hist:999/c=NL/cn;lang-en?(objectclass=*)
    - add operation
    - modify operation

LDAP Security

LDAP Security Model

- Client authentication at start of the LDAP connection
  - simple bind
    - send a DN and a password that is stored in the userPassword attribute of that entry
    - password gets sent in the clear
  - Simple bind with SSL (Secure Socket Layer): LDAPS
    - whole session is encrypted
  - Simple bind with TLS (Transport Layer Security)
    - StartTLS operation
    - whole session is encrypted

LDAP Security Model

- Alternatively bind with SASL mechanisms
  - Simple Authentication and Security Layer
  - E.g.:
    - Digest MD5 (challenge response)
    - GSSAPI (Kerberos 5)
    - External: using authentication information established on lower levels (SSL, IPSec)

Secure sockets in LDAP

- RFC 2830
  - TLS as defined in RFC 2246
  - Client sends Start TLS extended request
  - Server sends Start TLS extended response
  - TLS version negotiation (handshake)
  - Client may bind with SASL mechanism EXTERNAL
  - Client MUST check server identity
  - Client MUST refresh cached server capability information (eg. RootDSE)

LDAP Authentication

- RFC 2829: Authentication Methods for LDAP, May 2000

1. Read only, public directory
   - Anonymous authentication
   - No bind or empty Bind DN
2. Password based authentication directory
   - MUST support DIGEST-MD5 SASL mechanism (RFC 2831)
   - Client binds sasl mechanism DIGEST-MD5
   - Server sends back digest-challenge
   - Client binds again sending digest-response

LDAP Authentication contd.

3. Directories needing session protection
   - SHOULD use certificate-based authentication with TLS (RFC2830) together with simple bind or SASL EXTERNAL
   - Client uses Start TLS operation
   - Client and server negotiate ciphersuite with encryption algorithm
   - Server requests client certificate
   - Client sends certificate and performs a private key based encryption to prove its posession
   - Server checks validity of certificate and its CA
   - Client binds simple or with SASL "EXTERNAL" mechanism

Security threats in LDAP

- There had been a few buffer overflow holes in, e.g. OpenLDAP that could be used for DoS attacks
  - Only few and fixed quickly
- There are some password crack tools
  - Kold „Knocking on LDAPs Door“ online brute force dictionary attack in C (www.phenoelit.de)
  - The same in Perl: LDAP_Brute.pl (angreypacket.com)
  - Lumbejack offline brute force dictionary attack on LDIF files (www.phenoelit.de)
- Sniffers could read clear passwords
- None of these can harm, if you have your access control right and use encryption for LDAP connections or LDIF file transfer

Access control missing in LDAP standardization

- The IETF WG ldapext was working on that but failed to find consensus on Access Contol and is now closed down
- Only result: LDAP Access control requirements RFC 2820
- Work has been taken up by the replication group, which has consensus problems as well (see next slide)
- But of course every LDAP implementation has access control mechanisms
- For OpenLDAP Access Control see below

Access Control Requirements

- RFC 2820: Access Control Requirements for LDAP, E. Stokes, D. Byrne, B. Blakey, P. Behera. May 2000
  - Requirements for access control lists
  - easy, efficient, extensible
  - specific policies rule over non specific
  - default policy for new entries
  - sorting of the ACLs irrelevant
  - all ACLs must be explicit
  - ...

Replication missing in LDAP standardization

- The IETF WG ldup (LDAP Duplication/Replication/Update Protocols) works on replication standardization since 1998!
- The only RFC yet is again a requirements document RFC 3384
- Problem: group started with multi master replication problem and got stuck
- Almost finalized are two lightweight replication proposals:
  - LCUP (LDAP Client Update Protocol) for client synchronization: draft-ietf-ldup-lcup-05.txt
  - LDAP Content Synchronization Operation: draft-zeilenga-ldup-sync-03.txt

LDAP Replication cont.

- All current LDAP implementations have replication mechanisms
  - Either proprietary replication mechanisms
  - Or stick to the pseudo standard of University of Michigan implementation (SlurpD)
  - Or just use plain LDIF
  - New possibility: XML (DSML)
- Of course OpenLDAP has two replication mechanisms (see below)

How to find LDAP Servers

- R. Moats, R. Hedberg: A Taxonomy of Methods for LDAP Clients Finding Servers, <draft-ietf-ldapext-ldap-taxanomy-05>, July 2001 (expired but will be republished soon)
  - Client configuration
  - Well known DNS aliases
  - Referrals
  - SRV records
  - Service Location Protocol

Client configuration

- Simple
- Manual maintanance
- Not scalable

Well known DNS aliases

- RFC 2219: Use of DNS Aliases for Network Services, M. Hamilton, R. Wright, October 1997 (BCP)
  - Either: `ldap.university.nl IN A 194.167.157.2`
  - Or: `ldap.university.nl IN CNAME wp.university.nl`
  - Easy to implement
  - Not widely-used
  - Additional info (baseDN) needed to contact LDAP-server

Referrals

- Defined in LDAPv3
  - Referral part of LDAPresult to indicate that the server does not have the requested data but the servers referred to might have
  - Format: ref: <LDAP-URL(s)>
- Can be stored in a server
- Subordinate reference is specified in new RFC 3296
- A lot of other (more complete) attempts to standardize referral usage have failed

DNS SRV Records

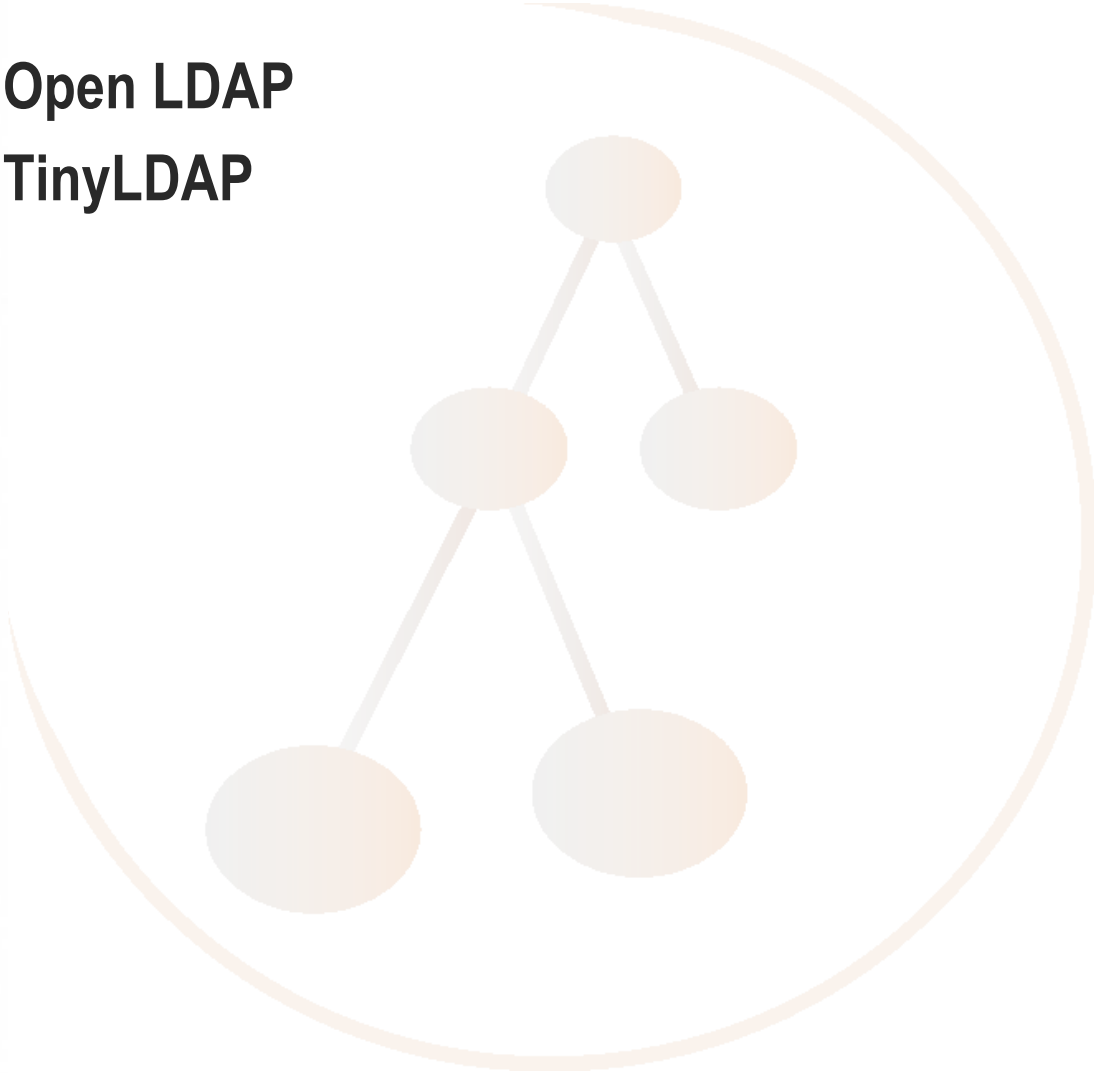
- RFC 2052, RFC 2782 and draft-ietf-dnsext-rfc2782bis-00.txt
  - _Service._Proto.Domain IN SRV Priority Weight Port Target
  - Used in RFC 3088: „OpenLDAP Root Service - An experimental LDAP referral service“
  - The automated system generates referrals based upon service location information published in DNS SRV RRs

Service Location Protocol

- V2: RFC 2608
  - Service template for LDAP
  - Highly sophisticated protocol
    - Uses multicast
    - User Agent – Service Agent
    - User Agent – Directory Agent – Service Agent
  - Rather designed for intranets

Open Source Implementations of LDAP

- Open LDAP
- TinyLDAP



Open Source Implementation OpenLDAP

- Current versions 2.x.x are LDAPv3 compliant
  - Current stable 2.1.22
  - Current Head: 2.2.0alpha
- Lots of important features like TLS, SASL
- A number of database backends to choose from (e.g., ldbm, bdb, hdb, sql)
- Good granular access control
- Stable replication mechanism (push and pull)
- Code well maintained by Kurt Zeilenga and a core developers team
- Used in large scale production environment
- Not very slow
- See www.openldap.org, especially Administration guide at www.openldap.org/doc/admin21/

Schema definition in Open-LDAP

- Schema definition files can be included by a line in slapd.conf, e.g.:
  - Include /etc/openldap/schema/core.schema
- Schema definition files contain RFC 2252 like attribute and objectclass definitions described above
  - One difference:
    add „attributetype “ or „objectclass “ before the round bracket

Access Control

Who

User, role, group, machine

Access Rules

Permissions, policies

Where

Servers, firewalls, applications

What

Data, e-mail, Web content, internet access,

From RL Morgan, PKI Standards Overview

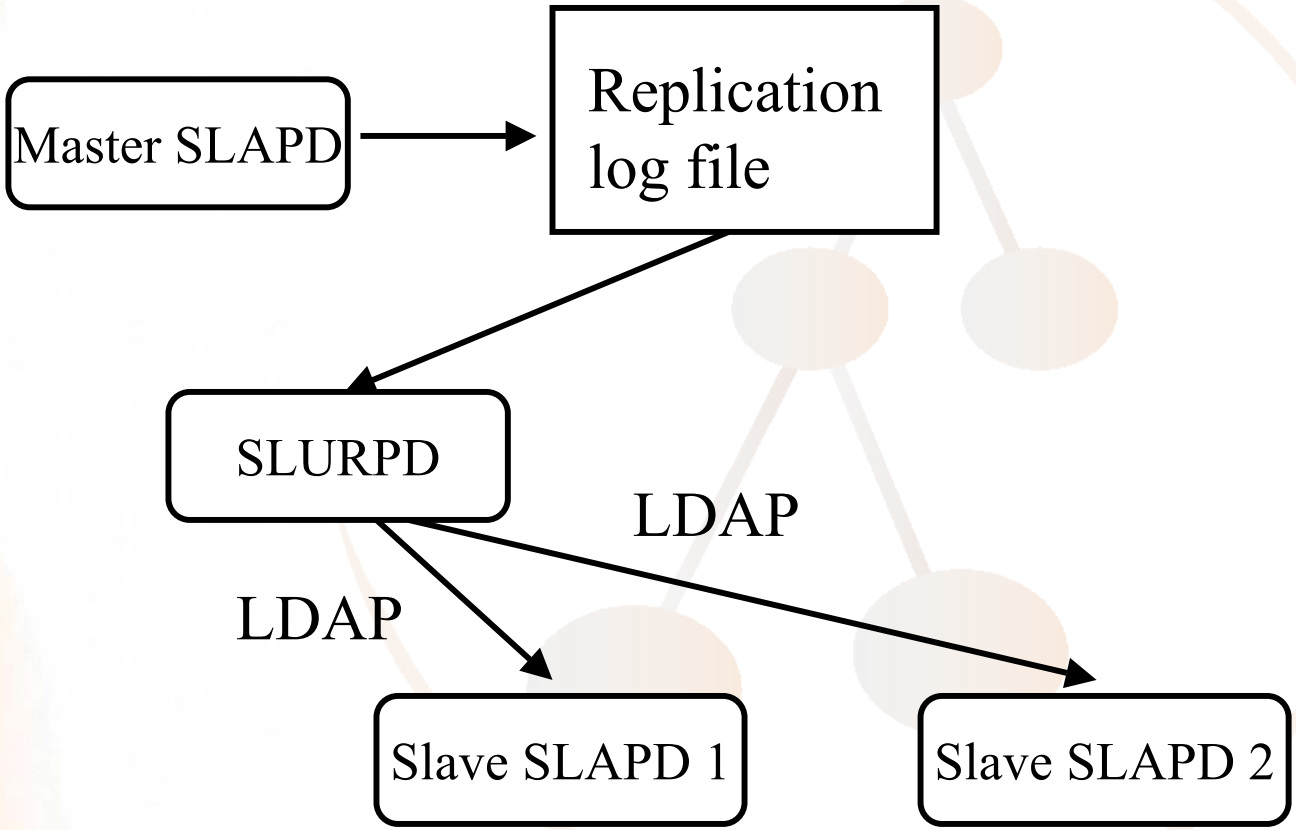
Access control in OpenLDAP

- Where access is controled is the LDAP server
- The access rules are stored in the configuration file (slapd.conf)
  - They are evaluated in order the rules appear in the config file
  - First rule that matches is used
  - Following permissions are specifiable: none | auth | compare | search | read | write
- What to control access to can be specified by:
  - Distinguished Name
  - Filter that matches some attributes
  - Attributes
- Who has access can be specified by:
  - Anonymous users
  - Authenticated users
  - Distinguished name
  - IP address or DNS name
  - „Self“, the user who owns the entry

OpenLDAP replication mechanisms

- Mechanism allready used in the U-Mich LDAP implementation using a replication daemon (slurpd)
  - Push model: master writes replication log slurpd pshes data to slaves
- The new draft LDAP Content Synchronization Operation: draft-zeilenga-ldup-sync-03.txt is being implemented in OpenLDAP head
  - Pull model: slaves ask for new data

Slurpd replication



Replication log file format

```
replica: host1.hu:9999
replica: host2.hu:8888
time: 960373276
dn: cn=Mister X, o=University, c=HU
changetype: delete

replica: host1.hu:9999
replica: host2.hu:8888
time: 960373277
dn: cn=Mister X, o=University, c=HU
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Xavier Xerxes
mail=X@dot.com
mail=Mister.X@dot.com
telephoneNumber=1234567
```

Performance Tests of Chadwick et.al.

- Thornton, Mundy, Chadwick: „A Comparative Performance Analysis of 7 Lightweight Directory Access Protocol Directories“ http://www.terena.nl/conferences/tnc2003/programme/papers/p1d1.pdf

Performance Tests of Chadwick et.al.

- **Tested LDAP implementations:**

Directory/Vendor	Operating System	Notes
Critical Path InJoin Directory Server 4.0	Windows 2000 Server	Loaned for evaluation from Critical Path @ http://www.cp.net
IBM SecureWay Directory 3.2.2	Windows 2000 Server	Free full product download available at http://www-3.ibm.com/software/network/directory/
iPlanet/SunONE Directory Server 5.1 (evaluation)*	Windows 2000 Server	Free trial download available at http://wwws.sun.com/software/products/directory_srvr/home_directory.html
Microsoft Active Directory	Windows 2000 Server	Integrated into Windows 2000 operating system.
Novell e-Directory 8.6	Windows 2000 Server	Free full product download available at http://www.novell.com
OpenLDAP 2.0.23	RedHat Linux 7.2	Free to full product download and source code available at http://www.openldap.org/
Syntegra Aphelion 2002	Windows 2000 Server	Loaned for evaluation from Syntegra @ http://www.syntegra.com

Table 1 – Directories Tested

Performance tests of Chadwick et.al.

- Plattform Intel Pentium 3 - 1GHz, 512MB RAM
  Microsoft Windows 2000 Server/Red Hat Linux 7.1
  Dual Partitioned Operating System
- Testsuite DirectoryMark 1.2.1
- 4 Testdatabases with number of entries:
  - 10,000
  - 100,000
  - 1,000,000
  - 10,000,000

	10K	100K	1 million	10 million
Critical Path IDS 4.0	00:01:32	00:22:31	11:00:34	-
IBM SecureWay Directory 3.2.2	00:01:58	00:14:04	02:21:58	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	00:00:10	00:01:22	00:12:13	29:54:13
Microsoft Active Directory	00:05:03	00:61:54	22:36:06	-
Novell eDirectory 8.6	00:14:12	-	-	-
OpenLDAP 2.0.23	00:00:37	00:08:36	13:12:36	-
Syntegra Aphelion 2002	00:00:07	00:00:35	00:04:29	01:54:05

Table 3 – Indexed Directory Load Times (HH:MM:SS)

	10K	100K	1 million	10 million
Critical Path IDS 4.0	00:01:12	00:09:11	01:35:49	-
IBM SecureWay Directory 3.2.2	00:01:49	00:12:57	02:08:12	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	00:00:08	00:01:02	00:09:10	02:08:12
Microsoft Active Directory	00:04:55	00:54:44	21:01:33	-
OpenLDAP 2.0.23	00:00:14	00:01:15	02:01:11	-

Table 4 – Un-Indexed Directory Load Times (HH:MM:SS)

	10K	100K	1 Million	10 Million
Critical Path InJoin Directory Server 4.0	1562.5	1562.5	1562.5	-
IBM SecureWay Directory 3.2.2	1666.7	1562.5	1666.7	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	2173.9	2272.7	2381.0	2272.7
Microsoft Active Directory	2000.0	2000.0	2000.0	-
Novell e-Directory	342.5	-	-	-
OpenLDAP 2.0.23	2272.7	1923.1	2173.9	-
Syntegra Aphelion 2002	2173.9	2000.0	2083.3	2272.7

Table 5 – Simulated Read (Base entry search on distinguished name) (operations/second)

	10K	100K	1 Million	10Million
Critical Path InJoin Directory Server 4.0	1515.2	1515.2	1515.2	-
IBM SecureWay Directory 3.2.2	1724.1	1612.9	1724.1	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	2272.7	2173.9	2272.7	2272.7
Microsoft Active Directory	2272.7	2272.7	1562.5	-
OpenLDAP 2.0.23	2381.0	1923.1	2381.0	-
Syntegra Aphelion 2002	2381.0	2173.9	2272.7	2381.0

Table 6 – Full subtree exact match search on common name (operations/second)

	10K	100K	1 Million	10Million
Critical Path InJoin Directory Server 4.0	1470.6	1470.6	1470.6	-
IBM SecureWay Directory 3.2.2	595.2	581.4	588.2	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	2381.0	2272.7	2381.0	2500.0
Microsoft Active Directory	2272.7	2272.7	1666.7	-
OpenLDAP 2.0.23	2500.0	1923.1	2500.0	-
Syntegra Aphelion 2002	2381.0	2173.9	2272.7	2381.0

Table 7 – Full subtree substring search on common name (operations/second)

	10K	100K	1 Million	10 Million
Critical Path InJoin Directory Server 4.0	83.3	6.8	3.8	-
IBM SecureWay Directory 3.2.2	20.0	16.7	11.5	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	28.6	16.1	15.9	11.6
Microsoft Active Directory	31.3	32.3	10.4	-
OpenLDAP 2.0.23	6.7	5.3	2.1	-
Syntegra Aphelion 2002	8.4	8.5	7.0	2.8

Table 9 – Add organizationalPerson Entry to Indexed Directory (operations/second)

	10K	100K	1 Million	10 Million
Critical Path InJoin Directory Server 4.0	200.0	200.0	31.3	-
IBM SecureWay Directory 3.2.2	21.3	19.6	15.6	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	40.0	43.5	30.3	18.5
Microsoft Active Directory	34.5	17.5	10.9	-
OpenLDAP 2.0.23	12.2	13.7	13.7	-

Table 10 – Add organizationalPerson Entry to Un-Indexed Directory (operations/second)

	10K	100K	1 Million	10 Million
Critical Path InJoin Directory Server 4.0	188.7	333.3	59.9	-
IBM SecureWay Directory 3.2.2	40.3	34.0	23.0	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	40.0	37.3	30.2	13.0
Microsoft Active Directory	96.2	98	32.8	-
OpenLDAP 2.0.23	3.3	2.4	1.3	-
Syntegra Aphelion 2002	12.5	12.2	9.4	2.7

Table 13 – Modify indexed attribute cn (operations/second)

	10K	100K	1 Million	10 Million
Critical Path InJoin Directory Server 4.0	312.5	277.8	45.2	-
IBM SecureWay Directory 3.2.2	78.1	70.4	48.1	-
iPlanet/SunONE Directory Server 5.1 (evaluation)	50.8	51.8	36.5	21.0
Microsoft Active Directory	99	95.2	48.1	-
OpenLDAP 2.0.23	5.8	4.2	1.6	-
Syntegra Aphelion 2002	26.5	27.5	23.1	37.5

Table 14 – Modify un-indexed attribute telephoneNumber (operations/second)

tinyLDAP

- Open Source project for a lightweight LDAP server implementation
- Database backend optimised for read performance
- Aim: very small and very fast
- Yet in an initial state:
  - Only search and bind operation implemented
  - No configuration file
  - No access control
  - No TLS or SASL support
- Project page: http://www.fefe.de/tinyldap/

Applications of LDAP

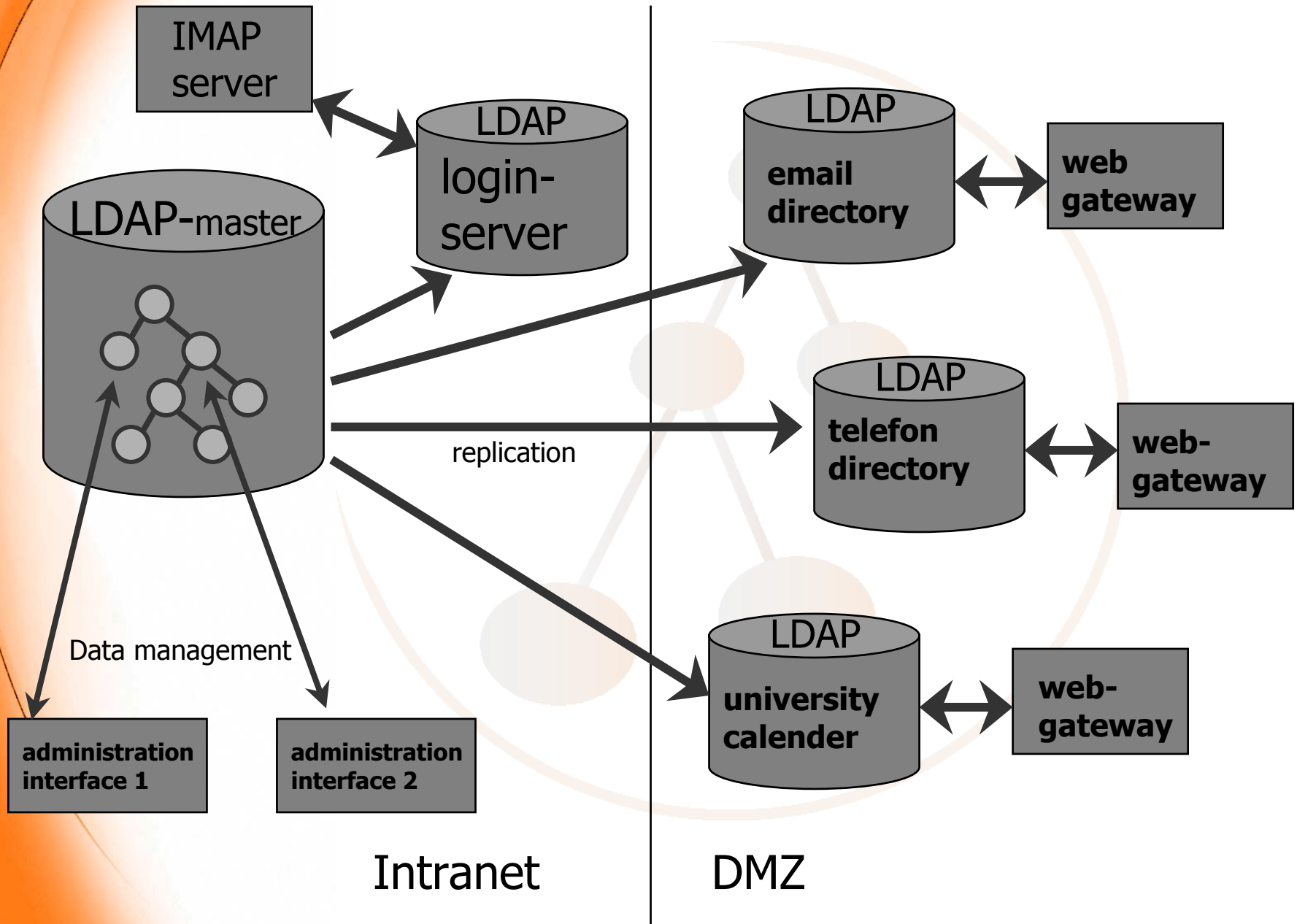
Classical Services

- Contact information of people
  - Name, address, telephone number, email address, ...
  - White Pages Directory Service
- Contact information of Organisations
  - Organisational structure, addresses, telephone numbers, email address, ...
  - Yellow Pages Directory Service

BTW: Good News!

- You can build up different Services with the same data
  - E.g. combine White Pages, Yellow Pages and User management in one Directory Information Tree on one or several Servers
  - Just add appropriate Objectclasses and data to your entries and set up a new user interface to the new data
  - This sincerly reduces management costs!

Example integration into other services



Lots of applications use LDAP now

- Some examples:
  - Apache user auth (http://www.muquit.com/muquit/software/mod_auth_ldap/mod_auth_ldap.html)
  - Squid ACLs (http://www.topf-sicret.de/projects/squid_auth.html)
  - Netscape Address book
  - Sendmail routing (http://www.sendmail.org/m4/ldap_routing.html)
  - Samba (http://www.unav.es/cti/ldap-smb-howto.html)
  - Netscape Roaming access (http://www.lut.fi/~hevi/ldap/netscape-roaming.html)

LDAP and Authentication services

LDAP for NIS

- RFC 2307: An Approach for Using LDAP as a Network Information Service, L. Howard, March 1998
  - Defines mechanisms for mapping entities related to TCP/IP and the UNIX system to LDAP
  - Deployment of LDAP as an organizational nameservice
  - Software available at: http://www.padl.com/nss_ldap.html

LDAP for NIS

- Defines objectclasses for:
  - UNIX user (/etc/passwd and shadow file)
  - Groups (/etc/groups)
  - IP services (/etc/services)
  - IP protocols (/etc/protocols)
  - RPCs (/etc/rpc)
  - IP hosts and networks
  - NIS network groups and maps
  - MAC addresses
  - Boot information

Useful Technologies 1

- Kerberos
  - Network authentication protocol with strong authentication for client/server environments
  - Each participant shares a secret key with a central Key Distribution Center (KDC)
  - KDC consists of Authenticate Service and Ticket Granting Service
  - Heimdal Kerberos can store data for principles etc. in LDAP
- GSSAPI (Generic Security Service Application Program Interface)
  - Security framework that abstracts from underlying protocols
  - Includes a Kerberos mechanism

Useful Technologies 2

- X.509
  - Certificate based strong authentication via assymetric encryption
  - Certificate issued by a third trusted party (CA)
- Security Layers
  - Integrity and privacy protection via encryption
  - Secure Socket Layer (SSL) / Transport Layer Security (TLS)
    - X.509 Certificate based
  - Kerberos and SASL also can establish Security Layers
  - IPSec: X.509 certificate based security at the network layer

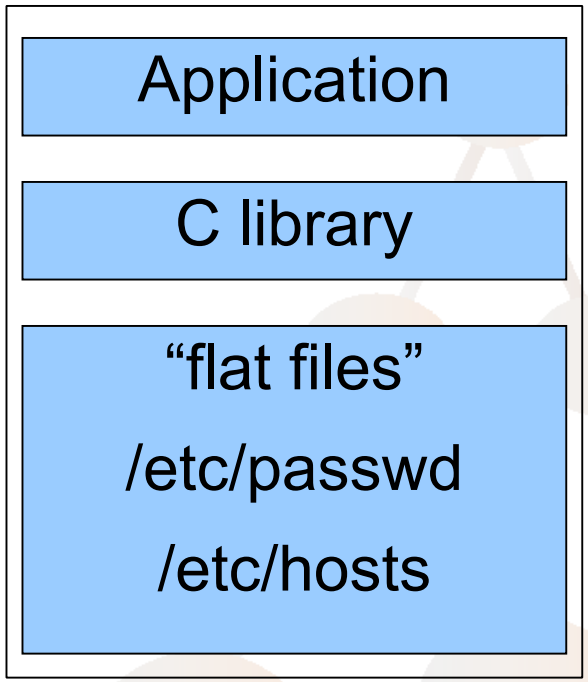
Useful Technologies 3

- SASL (Simple Authentication and Security Layer)
  - Method for adding authentication support to connection-based protocols
  - Supported by LDAP Servers
  - Specified mechanisms:
    - PLAIN (plain text password, we don‘t want that!)
    - DIGEST-MD5 (challenge Response no clear text PW)
    - GSSAPI (and thus Kerberos)
    - EXTERNAL (e.g. X.509 certificate used in the underlying SSL / TLS)

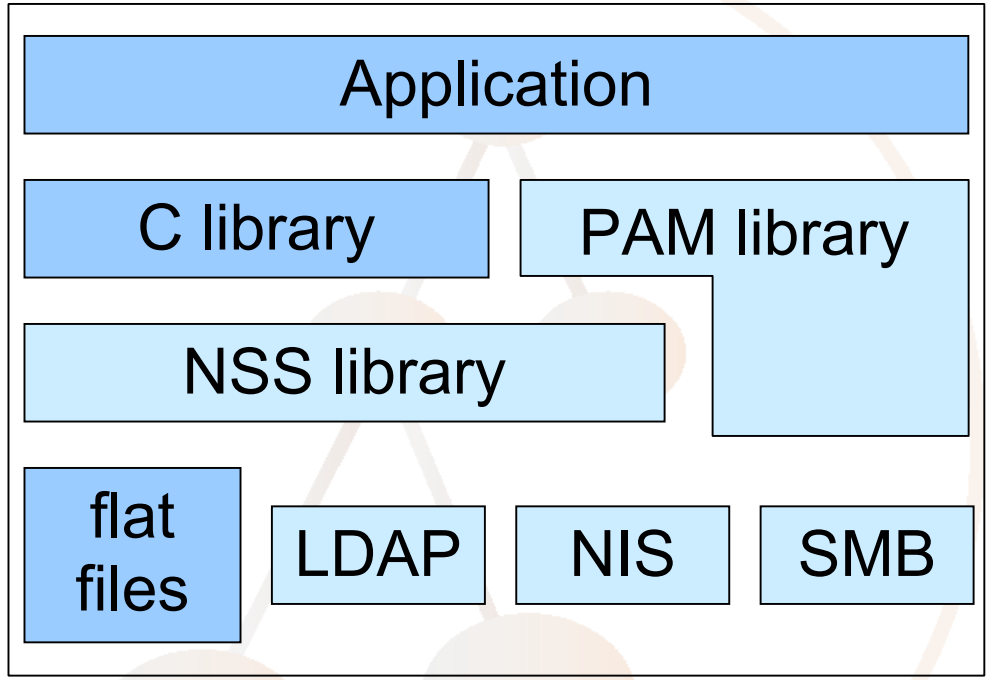
Useful Technologies 4

- Name Service Switch (NSS)
  - Layer in Unix C libraries that provides different means for listing or searching users, groups, IP services, networks, etc.:
    - Flat files (etc/passwd, etc.) = hard to administrate
    - NIS (Network Information Service) = security holes
    - LDAP = ☺
- Pluggable Authentication Modules (PAM)
  - Framework for login services
  - Manages authentication, accounts, sessions and passwords
  - Modules exist for LDAP, Kerberos, etc.

Unix authentication old



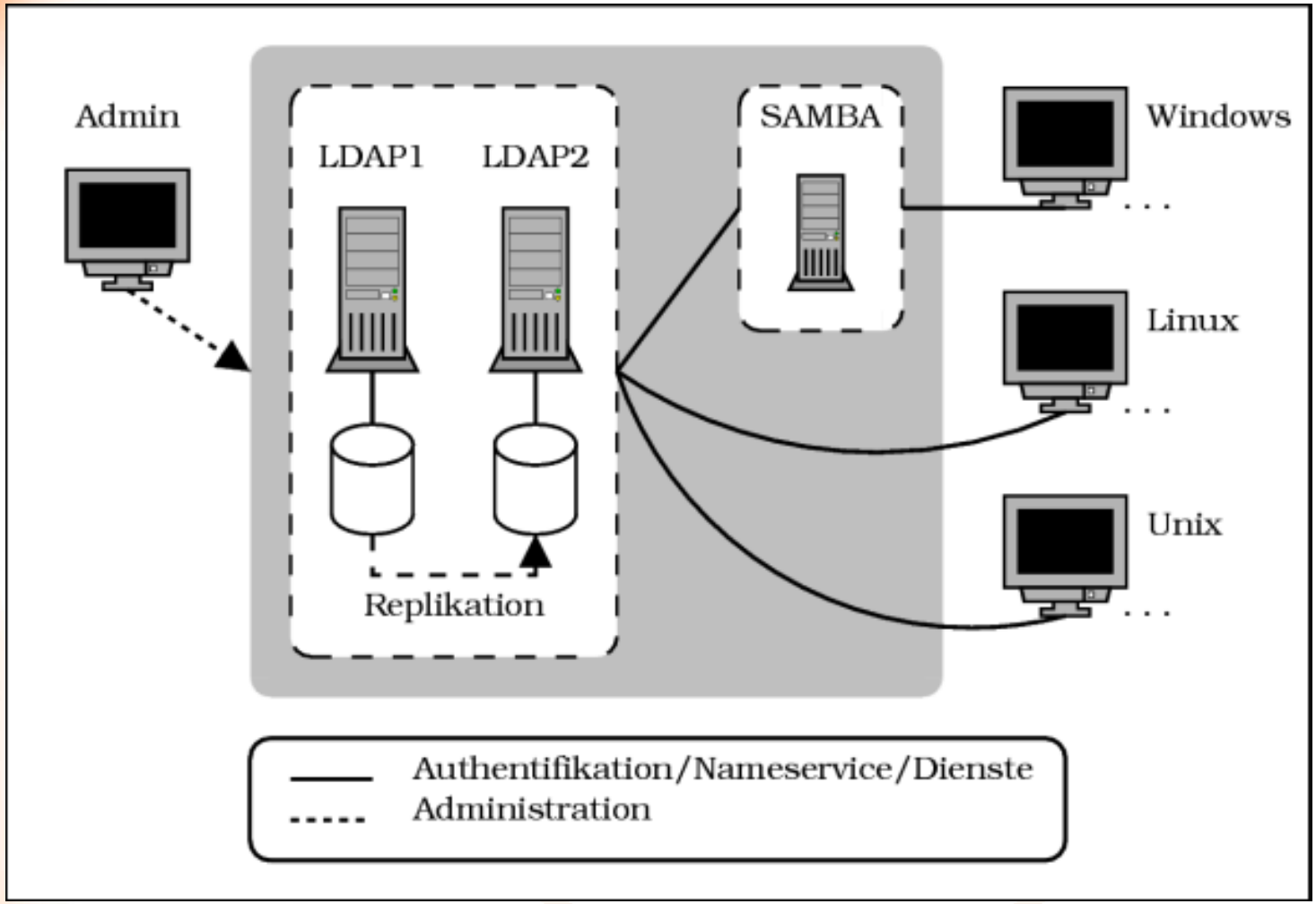
Unix authentication new



OpenLDAP/Samba recipe

- Take a linux box with minimal linux installation
- Add the following (newer versions will also do):
  - binutils-2.11.90.0.29-15.i386.rpm
  - gcc-2.95.3 136.i386.rpm
  - glibc-devel-2.2.4-40.i386.rpm
  - make-3.79.1-180.i386.rpm
  - nss_ldap-167-54.i386.rpm
  - openldap2-2.0.12-33.i386.rpm
  - openldap2-client-2.0.12-28.i386.rpm
  - openldap2-devel-2.0.12-28.i386.rpm
  - openssl-devel-0.9.6b-62.i386.rpm
  - pam-devel-0.75-78.i386.rpm pam_
  - ldap-122-77.i386.rpm
- And don‘t forget Samba, we took 2.2.8a
- Useful are the IDEALX smbldap-tools-0.7.tgz

Overview picture



Client platforms that work

- Unix:
  - Linux
  - FreeBSD
  - OpenBSD
  - NetBSD
  - Solaris
  - HP-UX
  - AIX
- Windows:
  - 2000
  - XP

Production service

- We currently use central authentication for:
  - Linux client login
  - BSD client login
  - Win2k client login
  - Cyrus-imapd
  - Sendmail smtp auth
  - sshd
  - cyrus-sasl
  - tutos (open source project planner / CRM)
- We do cashing via Name Service Cashing Daemon (nscd) which increases performance

Problems

- Memory allocation reentrance bug in SASL made the following authentication chain crash: cyrus-imapd -> cyrus-sasl -> pam -> pam_ldap
- Either redesign the SASL library (☹) or use the work around patch of Rein Tollevik

Zope based user/admin interface

- Easy to use interface for users and admins
- Using Zope
  - Very portable
  - Nice CMS functions
  - Has an LDAP API („LDAPUserFolder“)
- Interface uses SSL/TLS
- Manages any kind of data
- You can also use any other of the lot of LDAP administration tools

ULS Administrator Bereich - Konqueror

Location Edit View Go Bookmarks Tools Settings Window Help

Location: http://athena.directory.dfn.de:8080/authadmin/ULSadmin

ve direc... | Index of /samba/ftp | devel.samba.org | SAMBA - opening windows to ... | ULS Administrator Bereich

DAASI International

Directory Applications for Advanced Security and Information Management

Unified Login Server

ULS Administration | ULS Benutzereinstellungen | DAASI Homepage | Uni Tübingen | Verwende Rechte von Benutzer: **Anonymous User**

Zope @ DAASI

Configure | LDAP Schema | Caches | Users | Groups | Log | Undo | Ownership | Security

LDAPUserFolder at /authadmin/acl_users Help!

Change the basic properties of your LDAPUserFolder on this form.

Title	Zentrale Authentifikation		
Login Name Attribute	uid (uid)		
RDN Attribute	uid (uid)		
Users Base DN	ou=Users,o=smb,dc=daasi,dc=de	Scope	SUBTREE
Group storage	Groups stored on LDAP server		
Groups Base DN	ou=Groups,o=smb,dc=daasi,dc=de	Scope	SUBTREE
Manager DN	cn=root,o=smb,dc=daasi,dc=de	*Password*	*******
Manager DN Usage	For login data lookup only		
User object classes	top,inetOrgPerson,posixAccount,sambaAccou		
User password encryption	SSHA		
Default User Roles	Anonymous		

Page loaded.

Migration from AD to OpenLDAP

- IDEALX tools help to migrate passwords
- We wrote a script that migrates all infos stored in AD to the OpenLDAP server
- You can in theory also migrate the profiles since samba supports the roaming profile feature (we are still working on that)

Pros and cons

- Advantages:
  - User remebers only one password
  - Admin‘s and helpdesk‘s life is far easier
  - Unification of authentication processes
  - Central point for password evaluation
  - Before implementation you need a concept
- Caveats:
  - single point of failure (if without replication)
  - You need to enforce password policy (not yet implemented in OpenLDAP)
  - Admin access to clients should use local passwords

Our view on Samba 3.0

- The "ldap passwd sync" feature main reason to switch to Samba 3.0.
  - Users can change their password using the standard windows password change dialog.
  - Samba cares for the necessary steps to update both, the passwords used by windows (LDAP attributes: ntPassword and lmPassword) an the userPassword attribute that is used by Unix clients.
  - Samba can delete a complete dn if the user is to be deleted from the Samba account database (= ldapsam) or only remove the attributes concerning windows.

Samba 3.0 (contd.)

- The "ldap trust ids" feature
  - assumes that user ids returned from the LDAP database are always correct
  - So no need to lookup the corresponding Unix user.
  - This is very useful for our setup since we use nss_ldap and thus have valid UIDs in our database anyway.
- The upgrade process was clean and easy.
  - Having the account data in an LDAP directory does really help this process.
- Now the Code must prove its stability in our production environment.
- Not yet experimented with:
  - PDC replication stuff to set up a multimaster environment with Samba.
  - Samba Active Directory emulation.
  - group mapping of Samba 3.0 (still incomplete ?)

Where to go from here ?

- Use Samba 3.0 in production service
- We are about to include SSO functionality via Kerberos
- Password policy in OpenLDAP!
- What about a complete domain controller simulation via Samba?
  - AD replication!

LDAP and Kerberos

- Kerberos provides Single Sign On
- Two Open Source implementations: MIT and Heimdal
- Heimdal can use LDAP as data store for principals, etc.
- With PAM_kbr clients are easy to implement

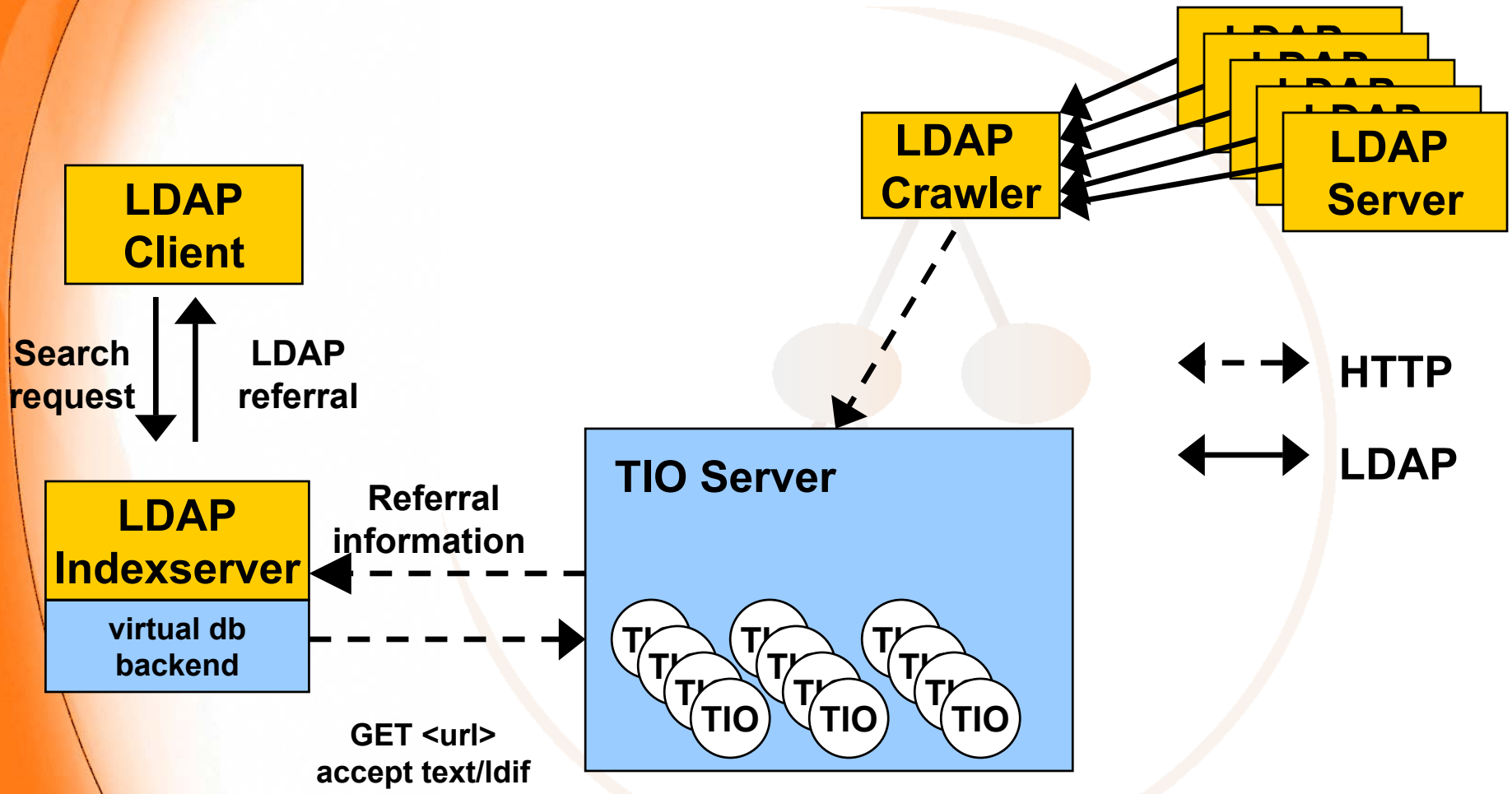
LDAP based Directory Services 2

Indexing for providing central services on distributed data

Common Indexing Protocol CIP (RFC 2651 – 2655)

- Index definitions for any directory technology
- Index meshs
- MIME wrapper
- Several Transport protocols (email, FTP, HTTP)
- Several Index Object Formats
  - E.g.: Tagged Index Object (TIO)

The LDAP Indexing System



What can the index system be used for?

- White Pages Service
- Metadata Service
- Certificate Service
- Every wide scale service on distributed data

LDAP based Directory Services 3

Public Key Infrastructure

PKI and Directory

The Burton Group: Network Strategy Report, PKI Architecture, July 1997: (Quoted after: S. Zeber, X.500 Directory Services and PKI issues, http://nra.nacosa.nato.int/pki/hdocs/pkiahwg30/index.htm)

" ... Customers should always consider PKI a directory-enabled set of services and infrastructure. Without directory services, PKI will be exponentially harder to implement and manage. Consequently, customers should't deploy PKI widely without an accompanying directory plan"

Directory as Key Server Requirements

- Publishing medium for public keys and certificates
- Gets public keys from user
- Gets certificates from CA
- Documents revocation of keys/certificates (CRL)
- Documents status of a certificate at a specific time

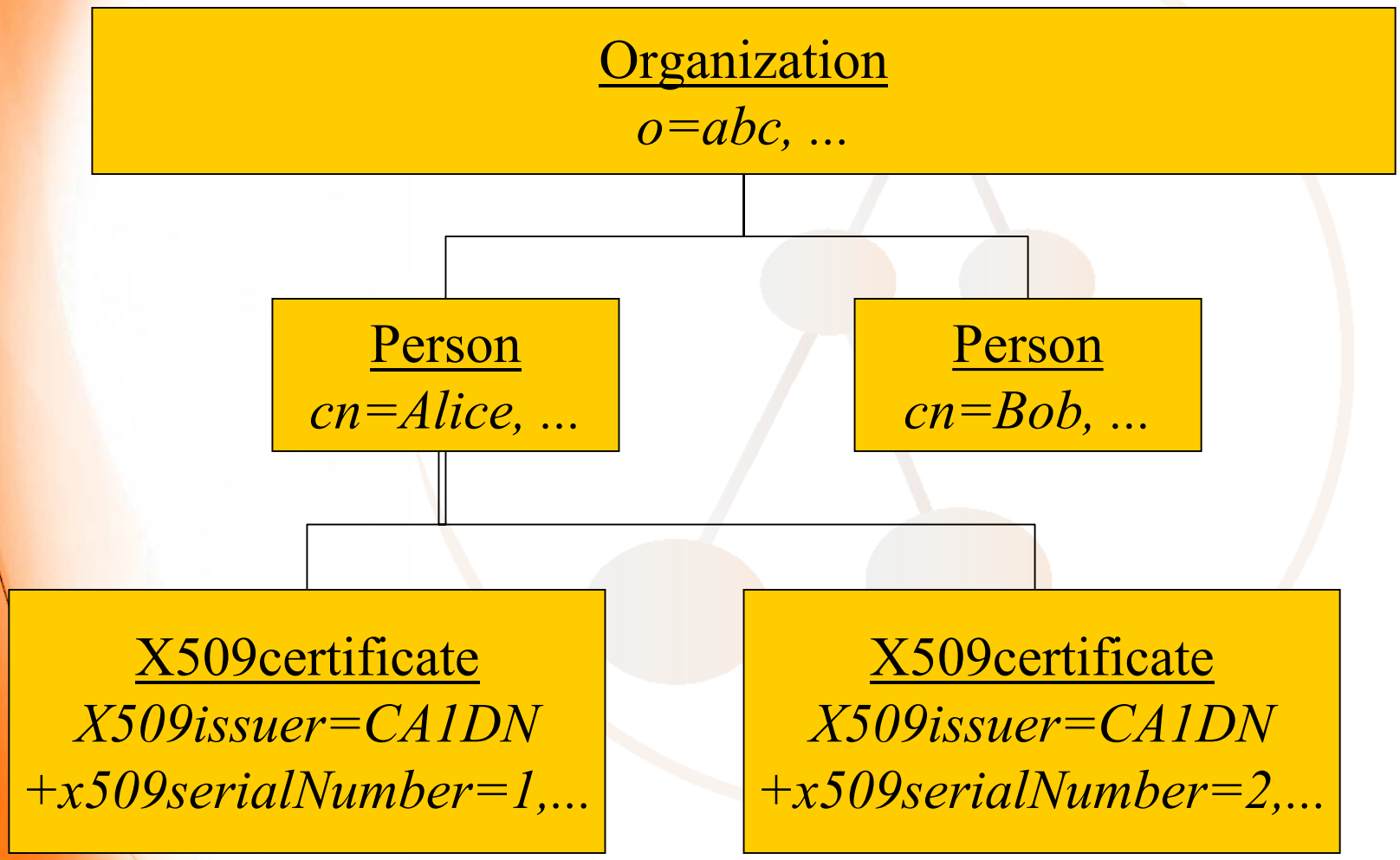
Motivation

- Address problem of multiple certificates for one entity
  - How can the client find the right certificate?
- Find a simple and easy to implement solution
- Solution should be usable in the frame of a large scale distributed LDAP / Common Indexing Protocol (CIP) based certificate repository

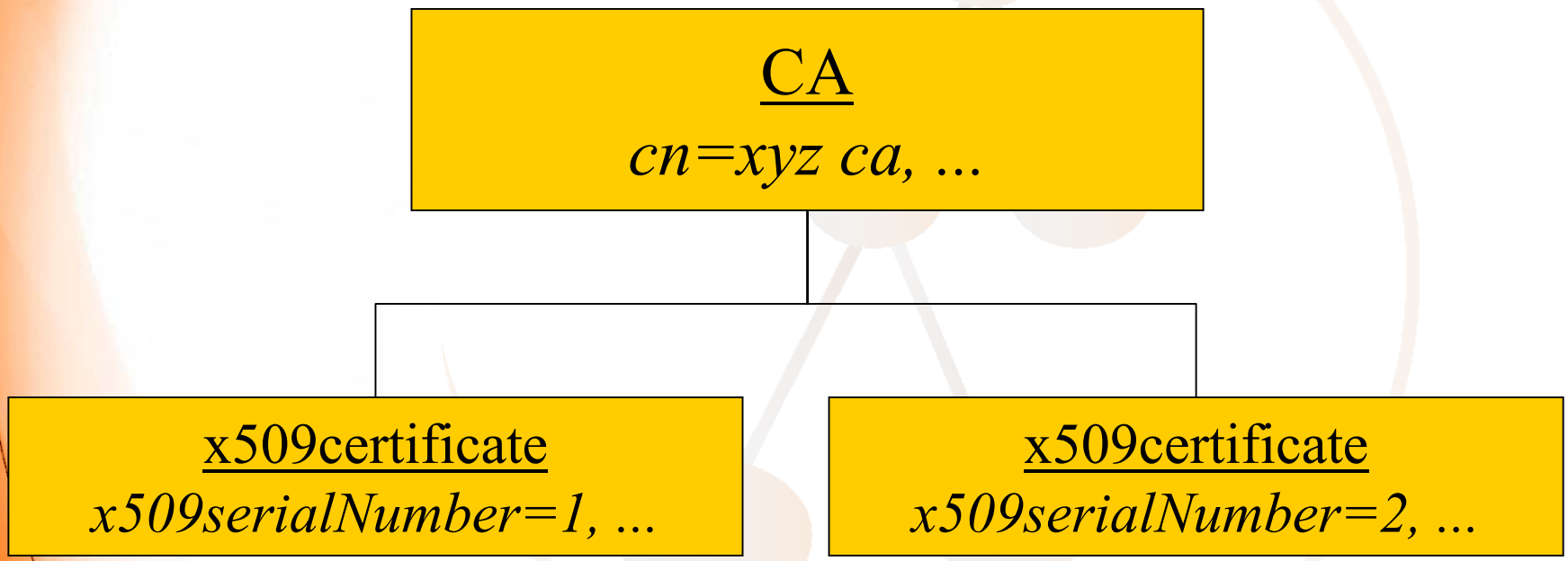
Schema as a simple solution

- Find a set of certificate fields and extensions that one might want to search upon
  - Meta-data approach
- Parse the certificate and store this set as LDAP attributes
- Advantages:
  - no new server features needed
  - easy to implement in clients
  - usable in a CIP environment

DIT Structure in white-pages services



DIT Structure in certificate repositories



LDAP based Directory Services 4

Metadata Service and the Semantic Web

Metadata

- Easiest definition: Data about data, e.g.:
  - Data: Texts, i.e. anything that tells us some kind of story (books, articles, webpages, films, etc.)
  - Metadata: Information about the texts (author, title, date of creation, etc.)
- There is one kind of Metadata that is really complicated: Keywords
  - How can we be sure that we use the same keywords for describing the same topics?
  - Controlled vocabularies!

Controlled Vocabulary

- Classification System
  - E.g. Dewey Decimal Classification
  - Classes, subclasses, subsubclasses, ...
  - One kind of relation between concepts
- Thesaurus
  - Assembly of homonyms
  - Could include antinyms and some more relations
  - A limited set of relations between concepts

Ontologies

- Again: Concepts and relations between them
- No limitation as to the number of different relations
  - Including sub/superclass
  - Including relationships of thesauri
  - ...
- Thus a perfect knowledge store

Current WWW

- Mere publishing medium
- Huge amount of information
- Designed for human access only
- Lack of structure and organization
- Insufficiant access methods
- Ambiguous:
  - bank (finance institute) the same as
  - Bank (river bank)

Visions for the future

- „Semantic Web“ (Tim Berners-Lee)
- Web Services
- Accessed by humans and programs
- Quality content better structured
- Knowlegde enhanced through Ontologies
- Disambigued:
  - Bank (finance institute) is not the same as
  - Bank (river bank)

How can Ontologies help?

- Remember: Concepts and relations between them
- Computer knows more than inputed

  Input: Parents have children

  Input: Mother = female parent

  Output: Mothers have children

Ontologie Storage Proposal

- Combined repository for metadata and ontologies
  - based on LDAP technology
  - thus accessible with the same protocol
- Large scalability
  - by setting up an Indexing system
  - based on Common Indexing Protocol (CIP)

Ontologie Storage Proposal

- Ontologie data model based on Common Information Model (CIM)
  - provides a model for associations that can be used for mapping the relations between objects
  - CIM is commonly used in Resource management and for Policy data
  - Technology independant modelling language (sort of UML)
  - Mappings to e.g. LDAP and XML

Common Information Model

- Object oriented meta model for structuring information technology independantly
- Capable of describing the whole computer world
- Basically an Ontology
- Three layers
  - Core: the basic lego bricks
  - Common: standardized descriptions
  - Extesion: vendor‘s extras

CIM, LDAP and Ontologies

- Any kind of relations can be defined with CIM and mapped to LDAP
- LDAP provides:
  - Object Class inheritance
  - Attribute inheritance
- Associations and aggregations can be mapped by object classes

Apropos Web Services

- SOAP
  - Simple Object Access Protocol
  - XML based Remote Procedure Calls
- WSDL
  - Web Services Description Language
  - XML based Interface description
- UDDI
  - Universal Description, Discovery and Integration
  - Repository for WSDL descriptions
  - Can be well replaced by LDAP

What can you do with it?

- Semantic Web, with search engines that *know* concepts (not strings)
- Intelligent agents that get data via automated Web Services from Portals
- Digital Libraries
- Content Management Systems
- E-Learning

LDAP based Directory Services 5

Policy repository

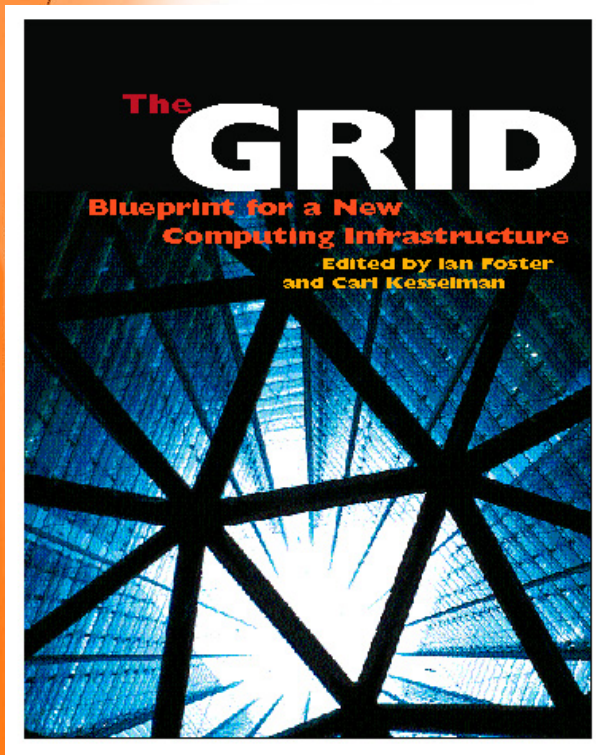
Policy repository

- Policy for Routers defining which packets to priortise, if and how to check authenticity, etc.
- Based Common Information Model (CIM)
- Directory Enabled Networks (DEN)
  - Quality of Service (QoS)
- IPSec policy
  - IETF WG IPSecpol
- Any other policies
- First attempt sort of failed. The idea still lives

LDAP based Directory Services 6

- Information for Grid Computing

The book



- Ian Foster, Carl Kesselmann (Ed) The Grid: Blueprint for a new Computing Infrastructure Morgan Kaufman Publishers, 1998
  - a summary of the state of the art of super computing,
  - now seen as the beginning of a new vision

The metapher

- Power Grid is a complex infrastructure that has a very simple user interface: the power outlet. Everything else is hidden from the user
- Grid Computing wants to provide an equally simple interface to computing power (CPUs, data storage, etc.) from the network.

Definitions

„The Grid is a consistent and standardized environment for collaborative, distributed problem solving that requires high performance computing on massive amounts of data that are stored, and/or generated at high data rates using widely distributed, heterogeneous resources „

„The Grid is an inherently layered architecture that provides for common services and a diversity of middleware that supports building distributed, large-scale, and high performance applications and problem solving systems. „

(W.E. Johnston as quoted by Ian Foster)

The tasks

- Distribution of data and computing ressources in broadbandwith networks to be able to provide petabyte storage and petaflops computing power
- Promotion of international collaboration
- Optimal utilization of resources (storage, CPUs, measuring devices, experimental devices

What is new?

- Metacomputing is in existance for quite a while
- New is the concept of standardized interface to meta computing, the so called Middleware
- The Global Grid Forum (GGF) took up the task to create such standards in an IETFish way
- Complicated requirements: “Run program X at site Y subject to community policy P, providing access to data at Z according to policy Q”

Requirements

- High bandwidth between powerfull systems
  - To specify simulations, initiate and steer computation
- Security
  - Use Encryption, Certificates, Single sign on
  - To Authenticate, negotiate and delegate authorization
- Data management
  - Use Distribution, Replication, Metadata
  - To locate and acquire resources, access remote datasets, collaborate on results

Grid Resource Information Service

- (Dynamic) Information about specific resources:
  - Load, process information, storage information, etc.
- Supports multiple information providers
- Answers questions like:
  - How much memory does machine have?
  - Which queues on machine allows large jobs?
- LDAP is an ideal technology and has been used for this
- New version of Open Source Grid computing implementation uses XML/Web Services now (Open Grid Services Architecture, OGSA)

Replica management

- Maintain a mapping between logical names for files and collections and one or more physical locations
- replica cataloging and reliable replication as two fundamental services
  - LDAP is used as catalog format and protocol, for consistency
  - LDAP object classes for representing logical-to-physical mappings in an LDAP catalog

New Trends in Grid Computing

- Web Services (see above)
  - Open Grid Services Architecture (OGSA)
  - Using SOAP and WSDL
  - A whole set of new GGF working groups
- CIM (see above)
  - Used for modeling grid related data
  - New working group on modelling Job Submission Information
  - CIM will be integrated in OGSA

Visions for the future

Well ....

- I didn‘t mention the term middleware
  - Lets have a short definition here:
  - A software layer between the network and network applications that provides standardized interfaces to commonly needed services

The Vision

- Globally used LDAP based Middleware that provides:
  - The same Authentication services to different applications
  - Ontology information to intelligent services
  - Information about automated services to agents
  - Policy information to network devices for intelligent routing

LDAP and XML

- Both are means to represent data
- XML databases are yet without advanced features of LDAP like authentication, Access Control, replication etc.
- XML is therefor often stored in relationaldatabases
- It could well be stored in LDAP, which can far better map the hierarchical structure of XML

Non LDAP References

- RFC 1510, „The Kerberos Network Authentication Service (V5)“
- RFC 1964, „The Kerberos Version 5 GSS-API Mechanism“
- RFC 2222, „Simple Authentication and Security Layer (SASL)“
- RFC 2246, „The TLS Protocol Version 1.0“
- RFC 2307, „An Approach for Using LDAP as a Network Information Service“
- RFC 2743, „Generic Security Service Application Program Interface Version 2, Update 1“
- RFC 2831, „Using Digest Authentication as a SASL Mechanism“

More references

- Samba: www.samba.org
  - IDEALX tools: www.idealx.org/prj/samba/index.en.html
- LDAP:
  - New drafts: www.ietf.org/html.charters/ldapbis-charter.html
  - OpenLDAP: www.openldap.org
  - NSS_LDAP: www.padl.com/OSS/nss_ldap.html
  - PAM_LDAP: www.padl.com/OSS/pam_ldap.html
  - Reentry patch from Rein Tollevik: www.openldap.org/lists/openldap-software/200108/msg00594.html
- X.509:
  - www.ietf.org/html.charters/pkix-charter.html
- Cyrus project (SASL, IMAP): asg.web.cmu.edu/cyrus/
- Zope: www.zope.org

Some more references

- Open Source LDAP Implementation: www.openldap.org
- Indexing system see TF-LSD: www.terena.nl/task-forces/tf-lsd
- PKI see IETF PKIX WG: www.ietf.org/html.charters/pkix-charter.html
- Semantic Web: www.w3c.org/2001/sw/
- Policy and CIM see DMTF: www.dmtf.org
- Grid Computing: www.gridforum.org

Thanks for your attention

- Questions?

- Peter@daasi.de