Introduction to LDAP

Workshop at

Carnet User Conference,

Zagreb, Croatia,

September 27, 2002

Peter Gietz

peter@daasi.de

DAASI International

Directory Applications for
Advanced Security and
Information Management

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.)

DAASI International

Directory Applications for
Advanced Security and
Information Management

Agenda

- What is a Directory
- What is X.500
  - History
  - Information model
  - Client server model
- What is LDAP
  - History
  - Concepts
  - Information model

DAASI International

Directory Applications for Advanced Security and Information Management

Agenda (contd.)

  - Functional model
  - Extensions
  - Replication
  - Access Control
  - New developments
- What can you do with it?
  - Indexing
  - PKI
- LDAP and Grid computing

DAASI International

Directory Applications for Advanced Security and Information Management

What is a Directory?

A short introduction

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)

DAASI International

Directory Applications for
Advanced Security and
Information Management

So what really is *the* 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
  - static view on the data
  - simple updates without transactions
- It has a network protocol for access
- A Directory Service may include
  - distribution in the net
  - replication of the data

DAASI International

Directory Applications for Advanced Security and Information Management

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

DAASI International

Directory Applications for
Advanced Security and
Information Management

What is X.500?

Some of these slides are for meant reading and are more of historical interest. ○

Some are basics for LDAP

X.500

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

DAASI International

Directory Applications for Advanced Security and Information Management

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

DAASI International

Directory Applications for
Advanced Security and
Information Management

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

DAASI International

Directory Applications for Advanced Security and Information Management

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

History of X.500: Projects

- 1989: NYSERNet White Pages Pilot Project
  - US initiative with participation of 90 organisations in 12 countries
- 1992: North American Directory Forum (NADF)
  - important US project
  - Specifications of directory service
- 1991: Piloting A ResArchers Directory Service in Europe (Paradise)
- 1993: DANTE takes over: NameFLOW-Paradise

What was X.500 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

Qualities of X.500

- Any amount of data can be stored
- On any number of servers
- Clients need to connect to only one server
- Data look the same everywhere
- Open model for any kind of data

DAASI International

Directory Applications for
Advanced Security and
Information Management

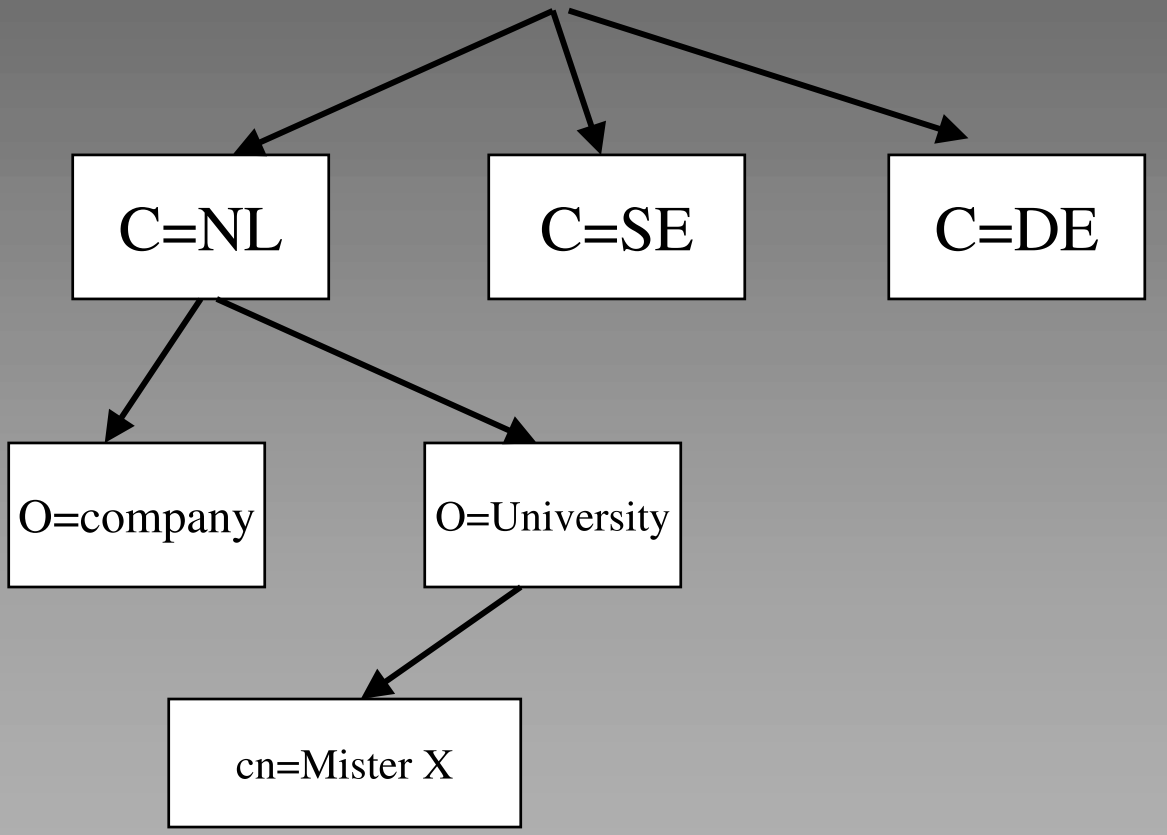
X.500 Information Tree

- 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

DAASI International

Directory Applications for
Advanced Security and
Information Management

Directory Information Tree (DIT)


C=NL

C=SE

C=DE

O=company

O=University

cn=Mister X

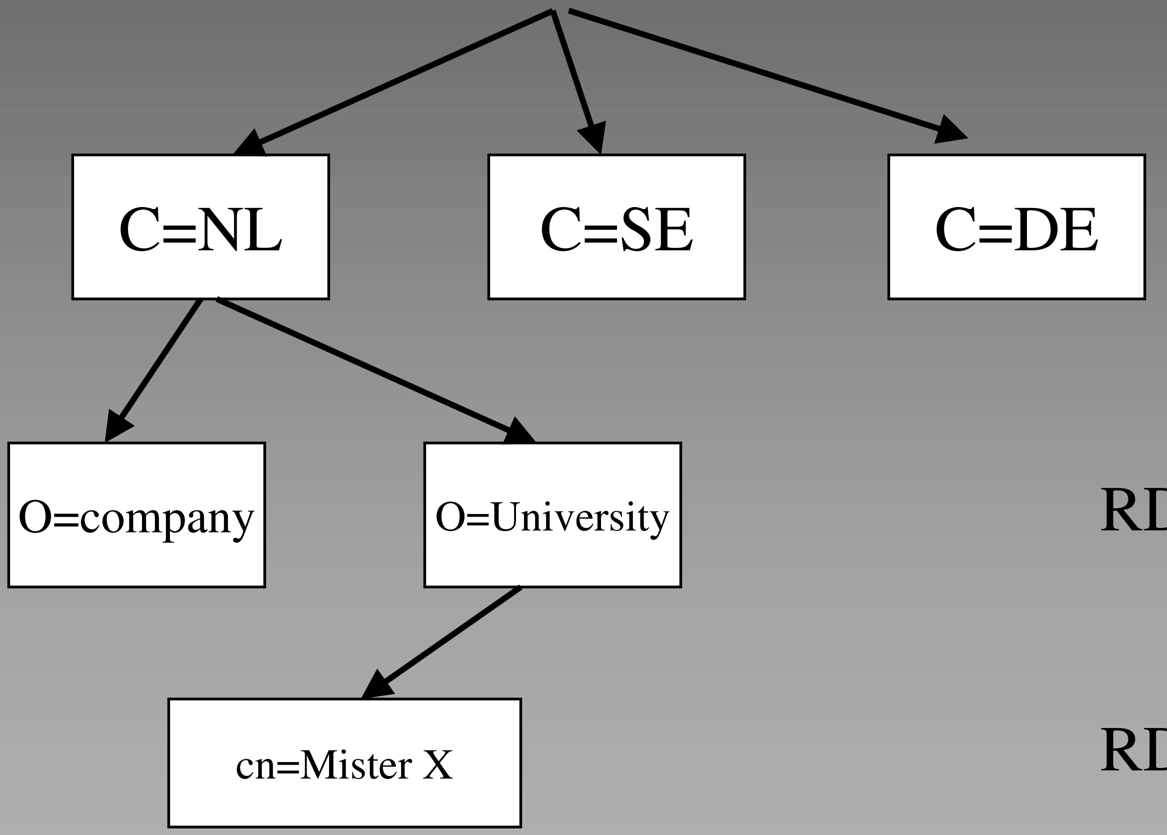
DAASI International

Directory Applications for
Advanced Security and
Information Management

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)



C=NL

C=SE

C=DE

O=company

O=University

cn=Mister X

RDN: C=NL

RDN: o=University

RDN: cn=Mister X

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

DAASI International

Directory Applications for Advanced Security and Information Management

DN Pointer

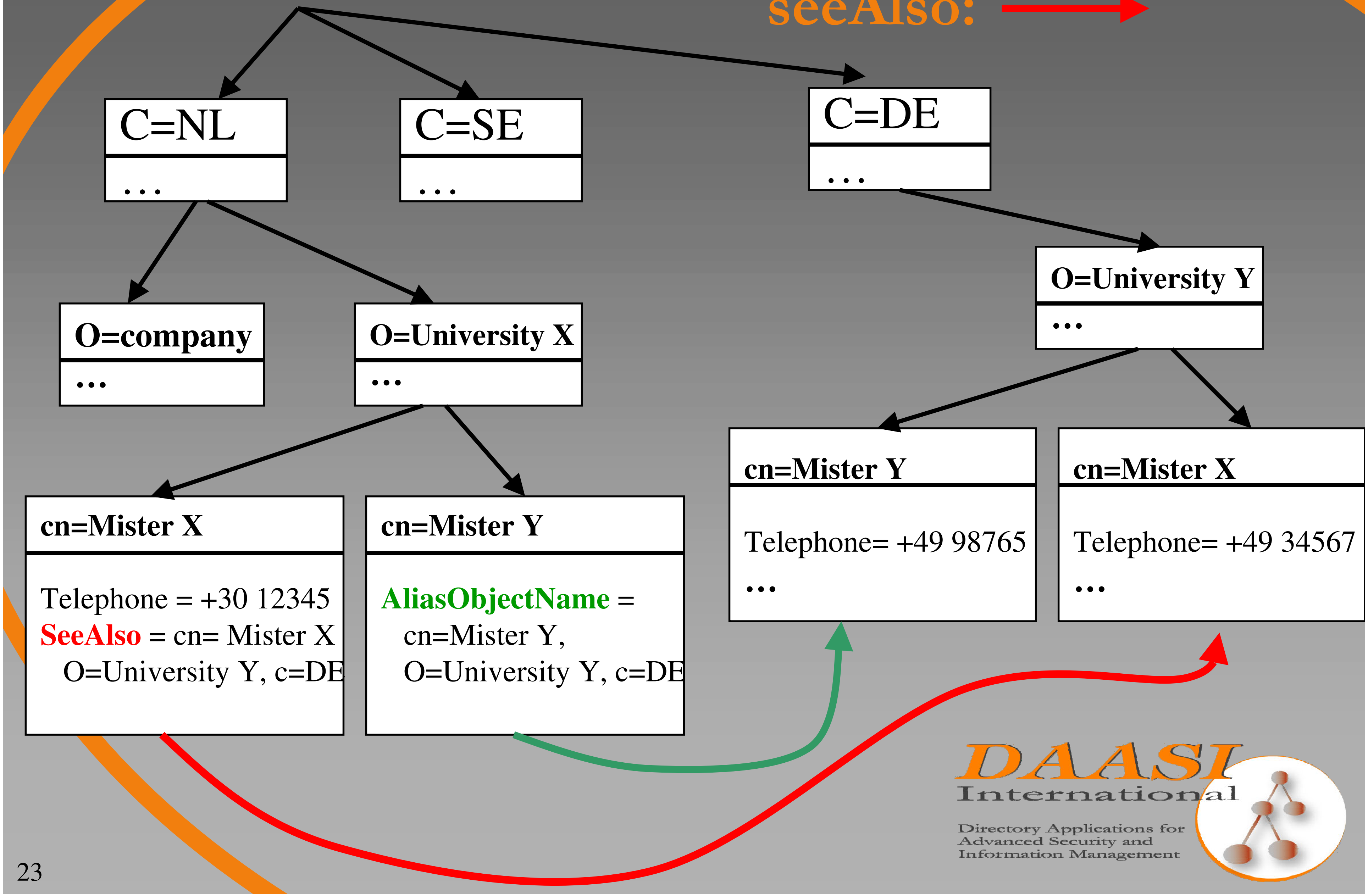
- Alias Entries have a DN and point to another DN via aliasObjectName Attribute
- seeAlso Attribute: Entry contains data and a seeAlso pointer to another DN

DAASI International

Directory Applications for Advanced Security and Information Management

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
…

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

DAASI International

Directory Applications for Advanced Security and Information Management

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 13865 Enterprise-numbers have been assigned

DAASI International

Directory Applications for
Advanced Security and
Information Management

X.500 Information Model

- An Entry contains a number 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
- Objectclasses can inherit Attributes from other Objectclasses
- A set of Objectclasses, Attributes and Syntaxes for a special purpose is called schema

DAASI International

Directory Applications for Advanced Security and Information Management

Objectclass inheritance

- One Objectclass can 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
  - Only used for the Object class at the root of the inheritance called top

Objectclass Types 2

- STRUCTURAL
  - 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 3

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.

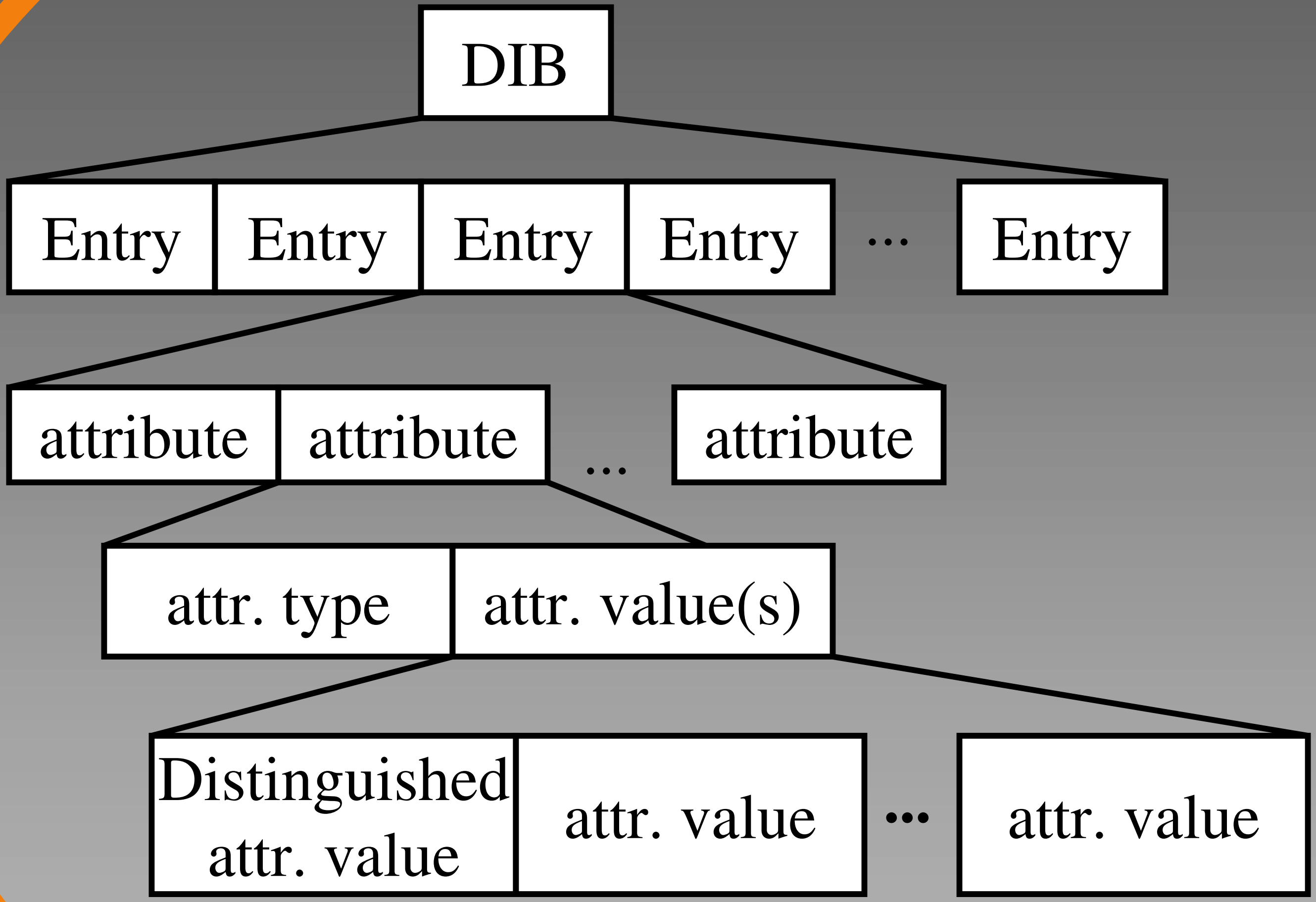
DAASI International

Directory Applications for Advanced Security and Information Management

Attribute inheritance

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

Directory Information Base



DAASI International

Directory Applications for
Advanced Security and
Information Management

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

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, ...

Open structure

- You can define your own:
  - Object Classes
  - Attribute Types
  - 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

DAASI International

Directory Applications for Advanced Security and Information Management

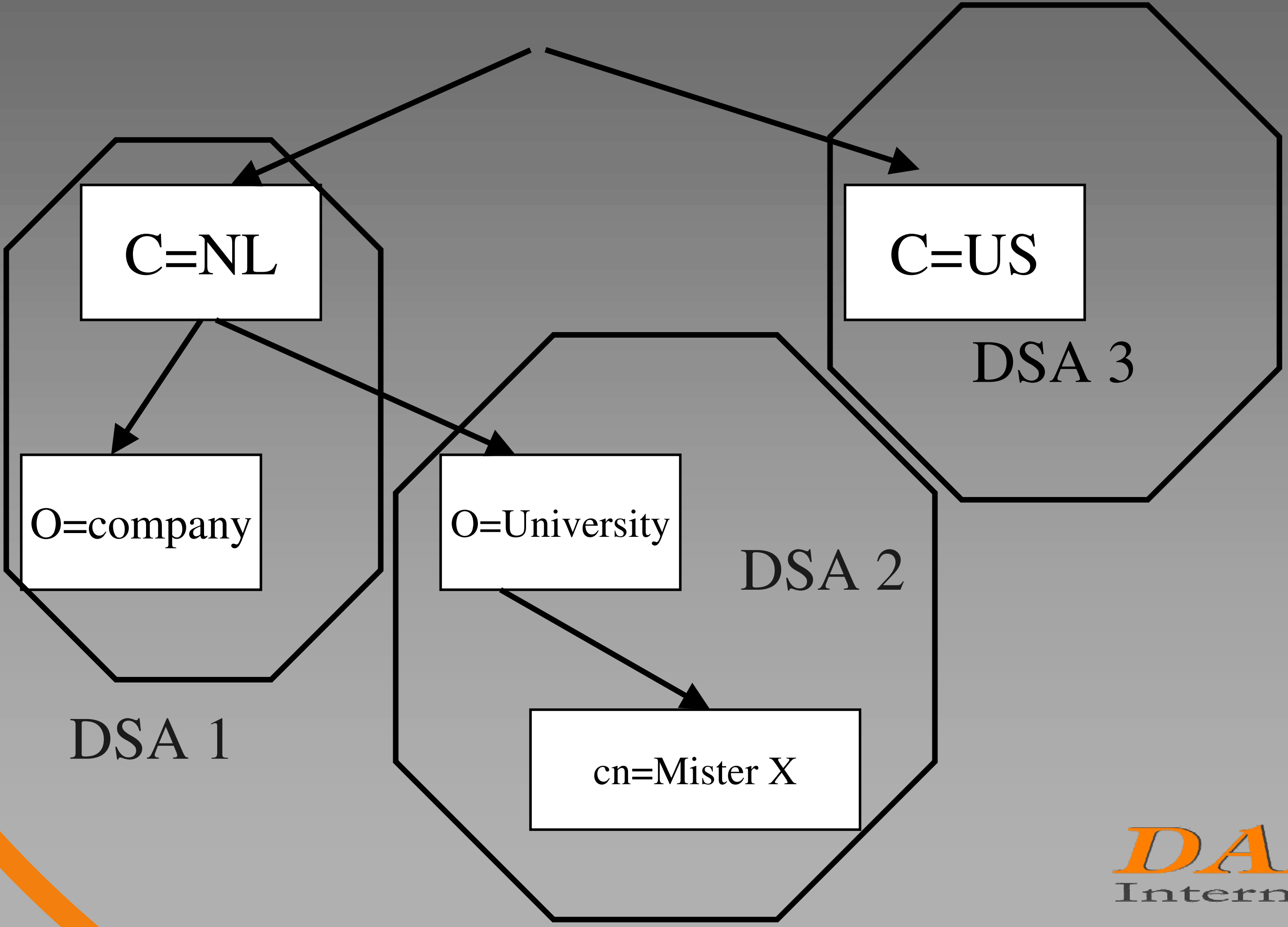
X.500 Client Server model

- 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 access protocol is called Directory Access Protocol DAP
- A lightweight version of DAP is LDAP
  Lightweight **D**irectory **A**ccess **P**rotocol

DAASI International

Directory Applications for Advanced Security and Information Management

Distribution of the data among DSAs

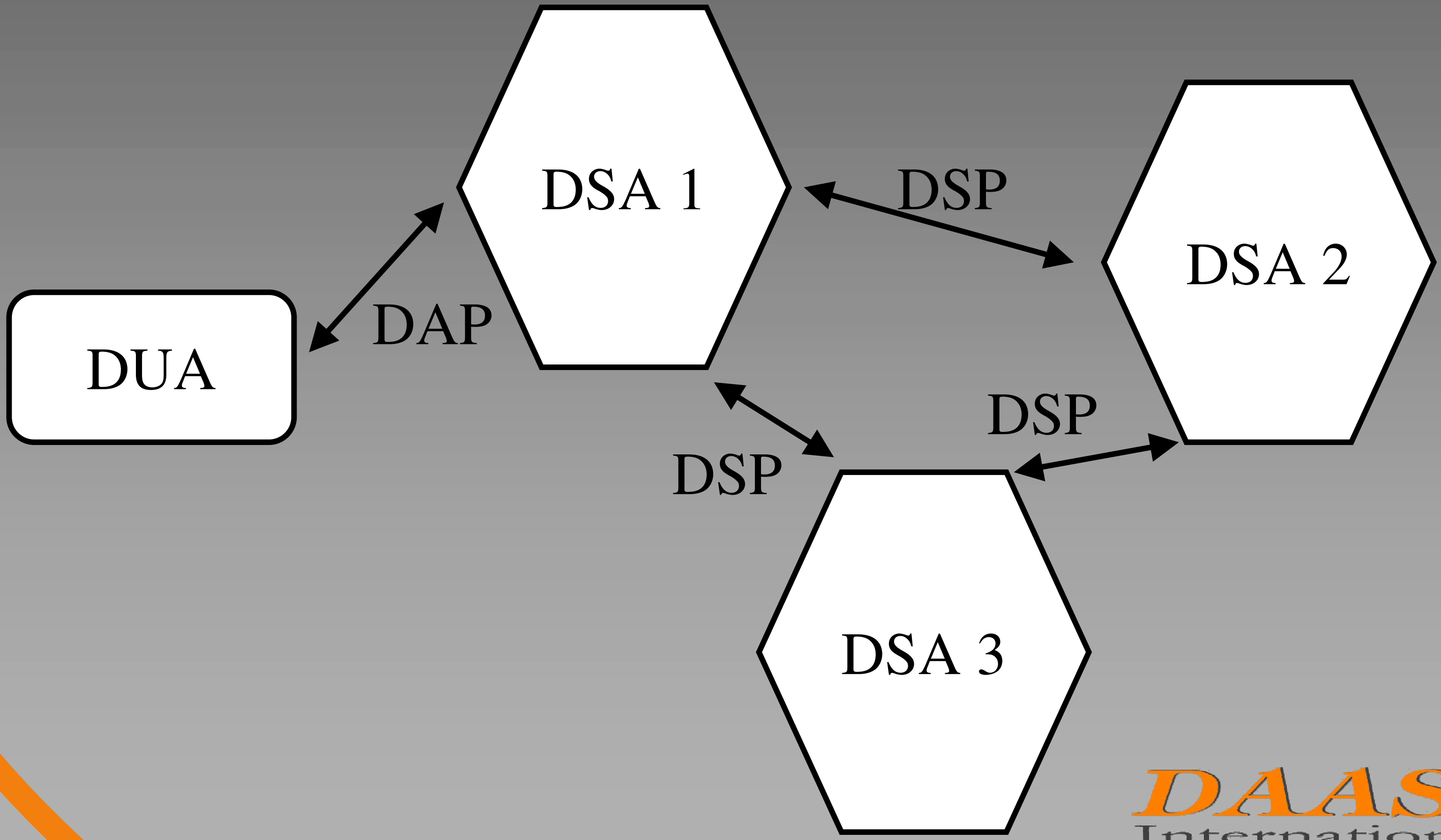


DAASI International
Directory Applications for Advanced Security and Information Management

Directory Server Protocols

- Directory System Protocol (DSP)
  - One DSA retrieves data requested by a client from another DSA
- Directory Operational Binding Management Protocol (DOP)
  - Knowledge references between DSAs
  - Hierarchical Operational Binding (HOB)
  - Shadow Operational Binding
- Directory Information Shadowing Protocol (DISP)
  - One DSA replicates data on another DSA
  - Protocol for replication agreements

Directory Server Protocols



DAASI International

Directory Applications for
Advanced Security and
Information Management

Some more X.500 Features

- All Protocols conform to the OSI Stack
  - 7 protocol layers with interfaces between each other
  - hard to implement
- Attributes can be inherited along the DIT (Collective Attributes)
- Authentication mechanisms
- Access control

DAASI International

Directory Applications for
Advanced Security and
Information Management

What is LDAP?

About LDAP standardization and differences to X.500

DAASI International

Directory Applications for
Advanced Security and
Information Management

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 Proposed Standard

- 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

DAASI International

Directory Applications for Advanced Security and Information Management

1997 LDAPv3 contd.

- RFC 2254:
  - The String Representation of LDAP Search Filters, T. Howes. December 1997
- 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

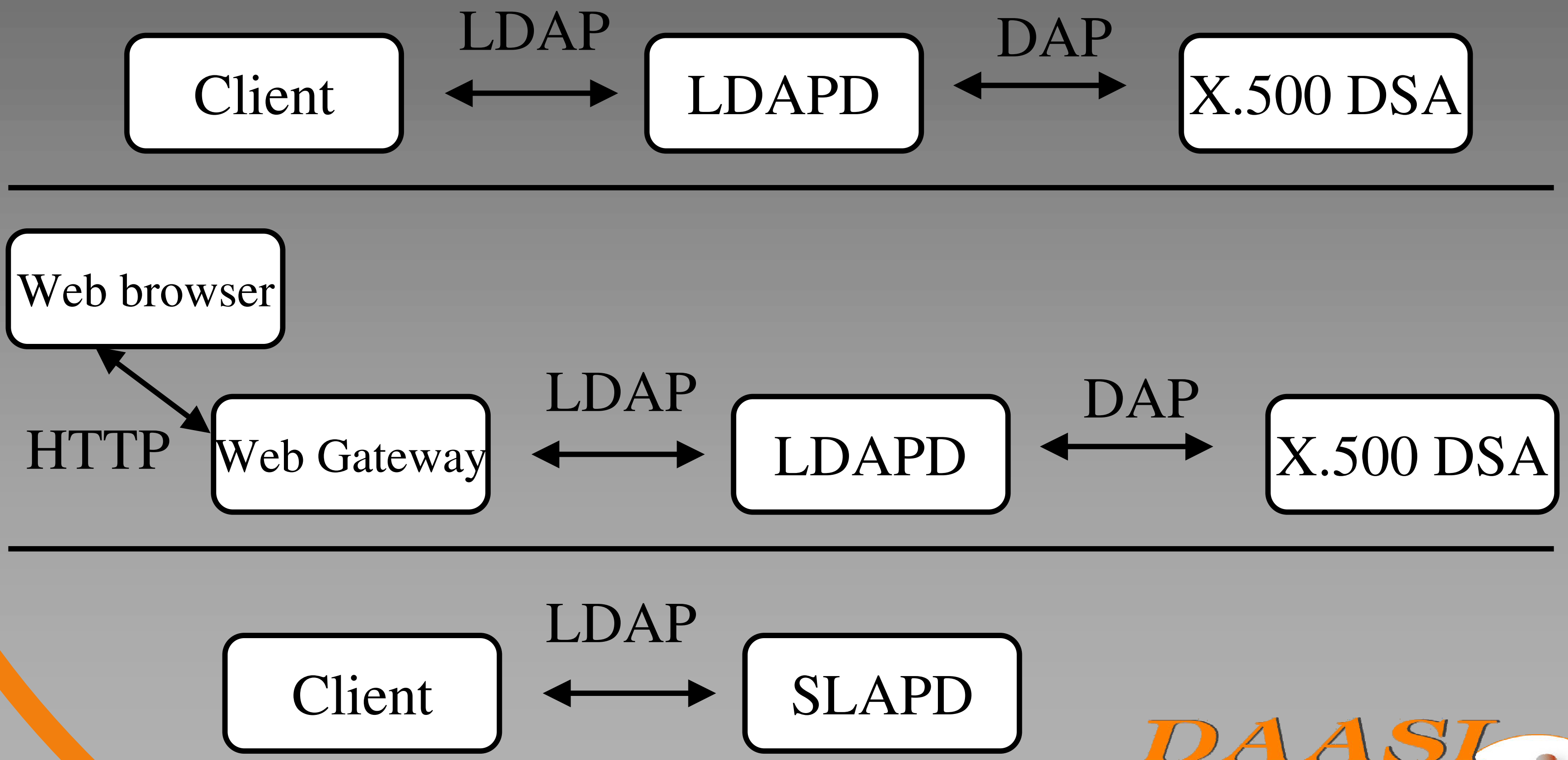
Who talks LDAP?

- 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

DAASI International

Directory Applications for
Advanced Security and
Information Management

LDAP connectivity



DAASI International

Directory Applications for Advanced Security and Information Management

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
  - A C API and a Java API

DAASI International

Directory Applications for Advanced Security and Information Management

Open Source Implementation

- OpenLDAP
  - Current versions 2.x.x are LDAPv3 compliant
  - Lots of important features like TLS, SASL
  - Code well maintained by Kurt Zeilenga and a core developers team
  - Used in large scale production environment
  - Not very slow
  - See www.openldap.org

DAASI International

Directory Applications for
Advanced Security and
Information Management

LDAP Information Model

- Just like X.500
  - Entry
  - Attribute Type
  - Attribute Syntax
  - Attribute Value
  - Matching Rule
  - Object classes
- Different:
  - String representation of the values
  - Attribute Description is AttributeType plus option separated by ´;´ also called tag. E.g. userCertificate;binary

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 ")"
```

DAASI International

Directory Applications for
Advanced Security and
Information Management

Attribute definition contd.

AttributeUsage =
"userApplications" /
"directoryOperation" /
"distributedOperation" / ; DSA-shared
"dSAOperation" ; DSA-specific, value depends on
server

Attribute definition contd.

oid = descr / numericoid

descr = keystring

numericoid = numericstring *(". " numericstring)

woid = whsp oid whsp ; set of oids of either form

oids = woid / ("(" oidlist ")")

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 Y
1.3.6.1.4.1.1466.115.121.1.24]

DAASI
International

Directory Applications for
Advanced Security and
Information Management

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))

DAASI International

Directory Applications for
Advanced Security and
Information Management

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

Schema definition in Open-LDAP

- Schema definition files can be included by a linen 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

DAASI
International

Directory Applications for
Advanced Security and
Information Management

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

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
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

DAASI International

Directory Applications for Advanced Security and Information Management

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

DAASI International

Directory Applications for
Advanced Security and
Information Management

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

DAASI International

Directory Applications for Advanced Security and Information Management

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

DAASI International

Directory Applications for
Advanced Security and
Information Management

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 +

LDAPv3 Extension Standardization

- Extensions have to be standardized:
- IETF WG ldapext
  - "successor" of the original LDAP WG asid
  - Charter: www.ietf.org/html.charters/ldapext-charter.html
  - Big Players like Netscape/Sun (= iPlanet), Microsoft and Novell very active in this WG
  - Still some overdue work to be done
  - Also other works than extension definitions
  - Besides this WG a lot of individual submissions
  - Officially closed WG

DAASI International

Directory Applications for Advanced Security and Information Management

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 2649

- An LDAP Control and Schema for Holding Operation Signatures, B. Greenblatt, P. Richard. August 1999 (EXP)
  - Client send modification of an entry on a secure connection (e.g. TLS) and signs this modification with S/MIME certificate, or lets it be signed by the server
  - a complete journal of modifications is stored

RFC 2649 contd.

- Defines:
  - Control SignedOperation
  - Control Demandsignedresult
  - Control SignedResult
  - Objectclass signedAuditTrail with Attribute Changes
  - Objectclass zombiObject with Attribute Changes and originalObject
  - RootDSE Attribute signedDirectoryOperationSupport

RFC 2696

- LDAP Control Extension for Simple Paged Results Manipulation, C. Weider, A. Herron, A. Anantha, T. Howes. September 1999 (INF)
  - Mechanism by which the server can give back several parts of the result
  - Client defines how many entries at a time
  - RFC Defines:
    - Control pagedResultControl
    - searchControlValue: realSearchControlValue
      - size (number of entries)
      - cookie (to re-identify the search request)

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

RFC 2891

- LDAP Control Extension for Server Side Sorting of Search Results, T. Howes, M. Wahl, A. Anantha, August 2000
  - Client can ask the server to sort the results by specifying an attribute to sort.

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

DAASI International

Directory Applications for Advanced Security and Information Management

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)

DAASI International

Directory Applications for Advanced Security and Information Management

LDAP work on X.509: TLS

- RFC 2830: LDAPv3 Extension for Transport Layer Security, May 2000
  - 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

DAASI International

Directory Applications for
Advanced Security and
Information Management

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

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-07 obsoletes RFC 2251 and portions of RFC 2252
- draft-ietf-ldapbis-models-00 obsoletes portions of RFC 2251, 2252 and 2256
- draft-ietf-ldapbis-syntaxes-02 obsoletes RFC 2252 and portions of 2256
- draft-ietf-ldapbis-dn-07 obsoletes RFC 2253
- draft-ietf-ldapbis-filter-02 obsoletes RFC 2254
- draft-ietf-ldapbis-url-0? obsoletes RFC 2255
- draft-ietf-ldapbis-user-schema-02 obsoletes RFC 2256
- draft-ietf-ldapbis-authmeth-03 obsoletes RFC 2829 and 2830

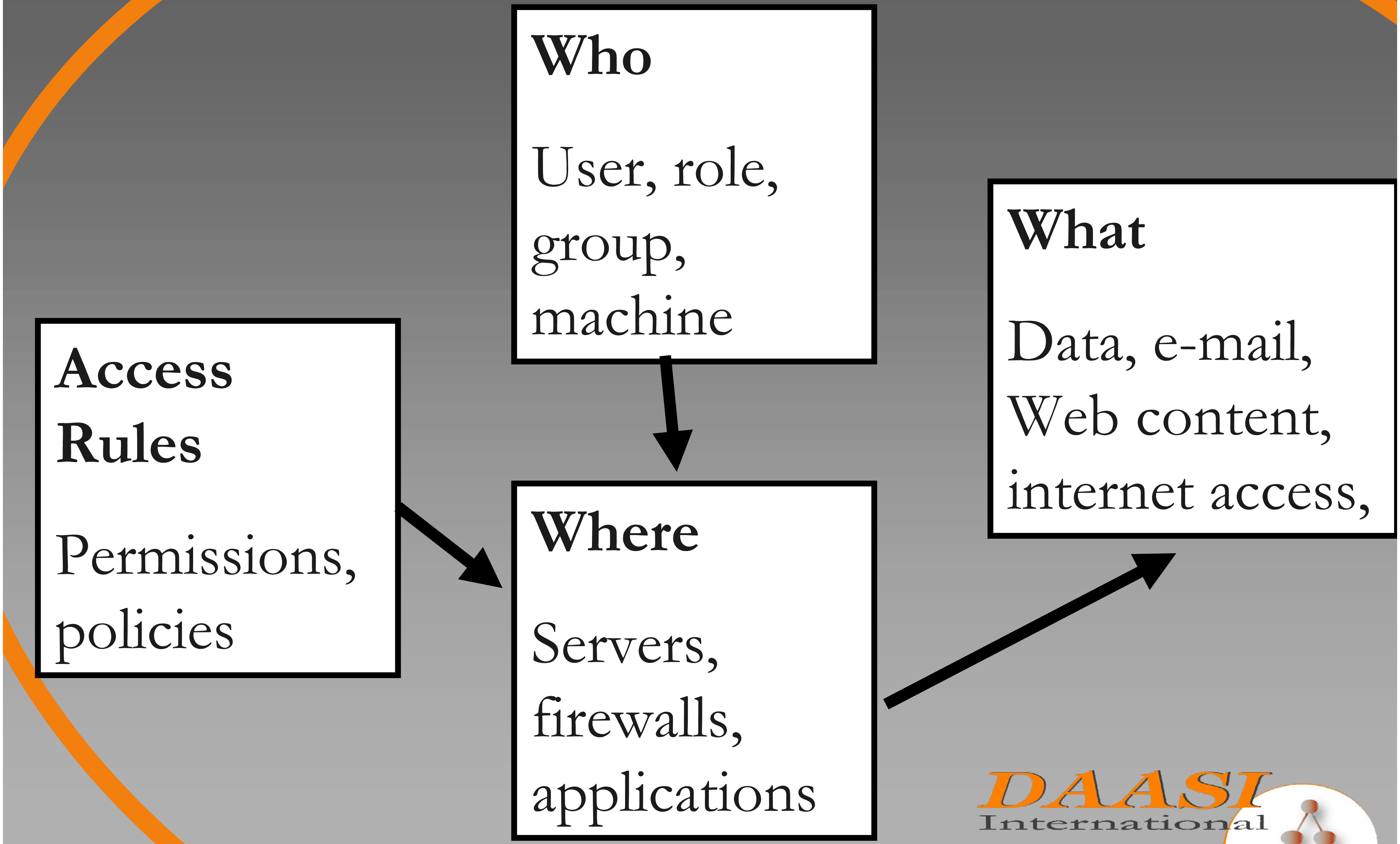
Current LDAPbis Drafts New Documents

1. LDAP: Technical Specification Road Map, Kurt Zeilenga, 21. February 2002
   - draft-ietf-ldapbis-roadmap-00
   - explicitly specify the set of Documents comprising LDAPv3 (RFC 2251-2256 and 2829-2830)
2. IANA Considerations for LDAP, Kurt D. Zeilenga, 12 May 2002
   - draft-ietf-ldapbis-iana-06
   - procedures for registering extensible elements of LDAP

Current LDAPext drafts with unknown status

- Access Control and authentication
  - Access Control model, X.509 Authentication with SASL
- Client Server communication
  - virtual lists, persistent search, referrals, matched values
- APIs
  - C-API and extensions, Java-API and extensions, additional error codes

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

DAASI International

Directory Applications for Advanced Security and Information Management

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
  - ...

DAASI International

Directory Applications for
Advanced Security and
Information Management

Access Control Model

- Access Control Model for LDAP, E. Stokes, D. Byrne, B. Blakey, <draft-ietf-.ldapext-acl-model-08.txt>, 29 June 2001 (expired!)
  - Access control information attributes for entries and subtrees (entryACI and subtreeACI)
  - Access control information subentry class ldapACISubEntry with attribute accessControlSchemes
  - RootDSE Attribute supportedAccessControlSchemes
  - LDAP functional model (add, delete, modify and search) for the manipulation of access control information
  - Additional control: getEffectiveRightsRequest and –Response for these manipulations

Basic ACI Attributes

- entryACI and subtreeACI with common syntax (Beware: this syntax has changed each new Draft version)
- Format:
  - <Rights> “#” <Attributes> “#” <Subject>

DAASI International

Directory Applications for Advanced Security and Information Management

Basic ACI Attributes contd.

- Rights:
  - "grant:" <permissions> and/or "deny:" <permissions>
  - Permissions for entries:
    - add, delete, export, import, renameDN, browseDN, view, returnDN, unveil (disclose on error), getEffectiveRights
  - Permissions for Attributes:
    - read, write, obliterate, search, search presence only, compare, make
  - permissions for attributes and permissions for entries are never found in a single ACI

Basic ACI Attributes contd.

- Attributes:
  - <attributes> or “[all]” or “[entry]”
  - attributes:
    - <attrDescr >[“,” <attrDescr> …]
    - attrDescr:
      - attributeType [“;” <options>]
      - Options: <option> or option “;” options
- Examples:
  - Cn
  - userCertificate;binary

DAASI International

Directory Applications for Advanced Security and Information Management

Basic ACI Attributes contd.

- Subject:
  - "authnLevel:" <authenticationlevel> “:” <identification>
  - Authenticationlevel:
    - "none" or "weak" or "limited" or "strong“
  - Identification:
    - “public:” or
    - “this:” or "authzId-" <authzId> or
      “role:” <DN> or “group: “ <DN> or “subtree:” <DN> or
    - "ipAddress:" <ipAddressRange(s)> or
      "dns:" <partialdomainname(s)>
    - authzId: "dn:" <DN> or "u:" <userid>

DAASI International

Directory Applications for
Advanced Security and
Information Management

ACI Examples:

- Grant read, search and compare of all attributes to all:

```
subtreeACI:grant:rsc#
[all]#
authnLevel:none:public:
```

- But deny for sensitive attributes:

```
subtreeACI:deny:rsc#
userPassword,subtreeACI,entryACI,
     salary#
authnLevel:none:public:
```

DAASI International

Directory Applications for
Advanced Security and
Information Management

ACI Examples contd.

- Let authenticated person modify her entry:

```
entryACI:grant:wo#
[all]#
authnLevel:strong:
authz-ID-dn:cn=ellen,dc=x,dc=com
```

- But let her not change ACIs and salary:

```
entryACI:deny:wo#
entryACI,subtreeACI,salary#
authnLevel:strong:authz-ID-dn:
cn=ellen,dc=x,dc=com
```

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, ...
```

Replication

- IETF WG LDUP
  - LDAP Duplication / Replication / Update Protocols
  - Charter: `www.ietf.org/html.charters/ldup-charter.html`
  - Active since 1998 but no RFC yet
  - Multi-master replication makes it very complicated
  - Atomicity issues
  - No single master replication profile yet

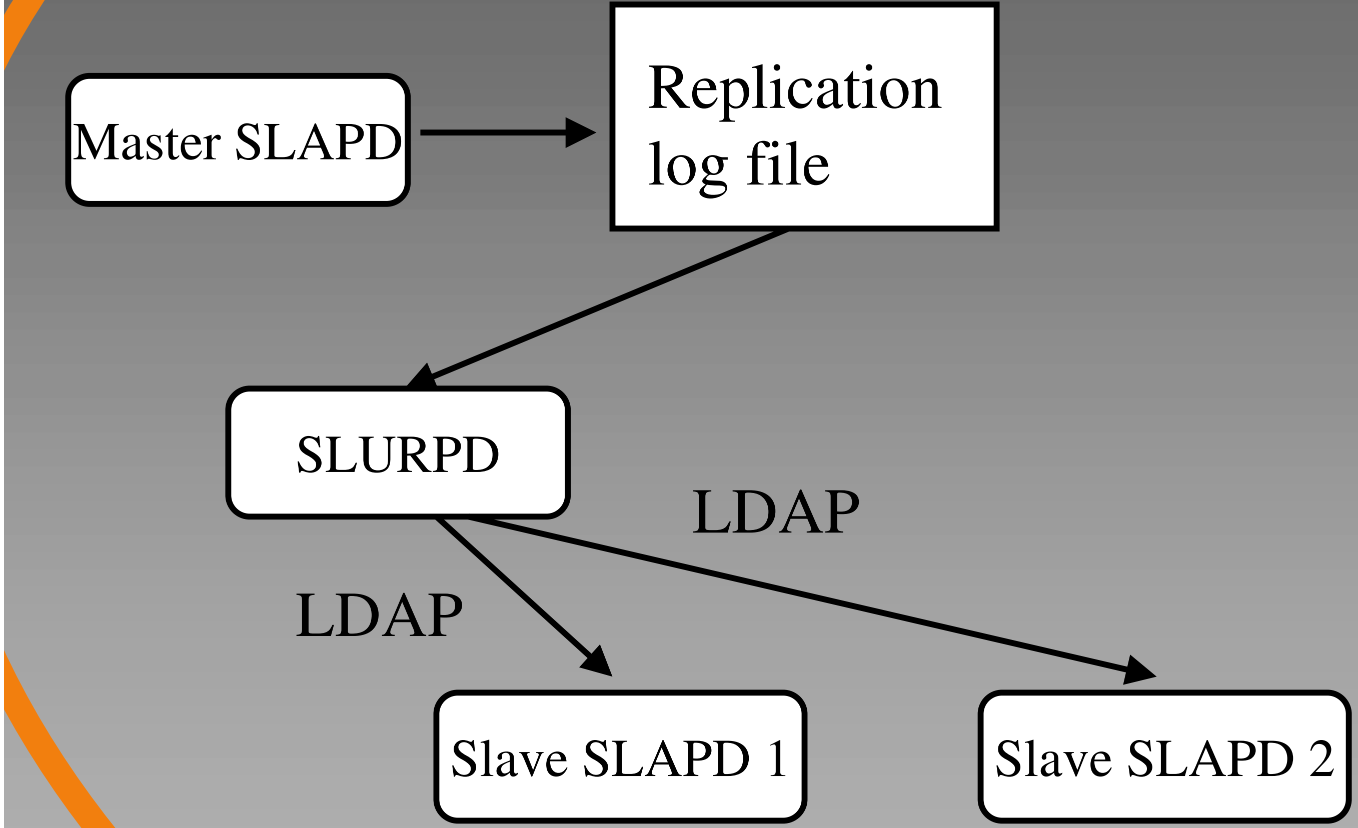
Replication

- Vital missing part in LDAP standardization
- Needed to really replace X.500
- Current LDAP implementations have
  - 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)

DAASI International

Directory Applications for
Advanced Security and
Information Management

Non Standard LDAP Replication



DAASI International

Directory Applications for Advanced Security and Information Management

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
```

DAASI
International

Directory Applications for
Advanced Security and
Information Management

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

  - 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

DAASI International

Directory Applications for
Advanced Security and
Information Management

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: referral: <LDAP-URL(s)>
- Can be stored in a server
- The exact data model is not standardized yet
  - Subordinate reference and superior reference
- A lot of attempts to standardize usage have failed

DAASI International

Directory Applications for
Advanced Security and
Information Management

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 draft-zeilenga-ldap-root-01.txt: „OpenLDAP Root Service - An experimental LDAP referral service“
- DNS SRV *and* referrals:
  - draft-zeilenga-ldapnsref-00.txt
    - Objectclass dNSReferral
    - Ref: ldap:/// + SRV -> complete referral

DNS SRV Records contd.

- **DNS SRV and URIs:**
  - draft-andrews-http-srv-00.txt
    - Can be used for looking up ldap ports
    - Conflict resolution: ports in URI and SRV RR
- **DNS SRV and PKIX:**
  - draft-ietf-pkix-pkixrep-00.txt
  - PKIX Repository Locator Service for:
    - LDAP
    - HTTP
    - OCSP

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

What can we do with Directories

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

DAASI International

Directory Applications for Advanced Security and Information Management

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

DAASI International

Directory Applications for Advanced Security and Information Management

LDAP for unified authentication

- Each user only needs a single username or ID and password for all systems
- Usable for e.g. : IMAP, POP, SMTP auth, FTP, HTTP auth, RSH, SSH, etc. etc.
- Based on PAM (Pluggable Authentication Modules)
  - Authentication management;
  - account management
  - Session management
  - password management

DAASI International

Directory Applications for
Advanced Security and
Information Management

LDAP for unified auth.

- PAM_LDAP
  - Module for account /password/authentication management with LDAP
  - Software available at: http://www.padl.com/pam_ldap.html
- Plaintext SASL mechanisms can make use of PAM_LDAP

DAASI International

Directory Applications for Advanced Security and Information Management

Questions?

- DFN Directory Services
  - peter.gietz@directory.dfn.de
  - www.directory.dfn.de
- DAASI International GmbH
  - Info@daasi.de
  - www.daasi.de

DAASI International

Directory Applications for
Advanced Security and
Information Management