D D S

An Introduction to LDAP and its applications

Sixth CEENet Workshop on Network Technology

24.8.2000, Budapest

Peter Gietz

Peter.gietz@directory.dfn.de

D D S

Agenda

- What is a Directory
- What is X.500
- What is LDAP

D D S

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
  - Novell Directory Service (NDS)
  - Microsoft Active Directory (AD)
  - Lightweight Directory Access Protocol (LDAP)

D D S

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 on several servers

D D S

What kind of data can you store?

- Text data
  - names, addresses, descriptions, numbers, etc.
- Graphics
  - photos, diagrams, etc.
- Pointers
  - URLs, pointers to other data, etc.
- Public key certificates
- Other binary data
- Anything else you can think of

D D S

What is 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

D D S

Responsible International boards

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

D D S

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 X.500 standard
  - 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
  - includes the missing bits
- 1997 third version
  - includes enhanced definitions for authentication (X.509v3)
- 2000 fourth version

D D S

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 (since 1993 version)

D D S

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 and names the project NameFLOW-Paradise

D D S

What was X.500 originally intended for?

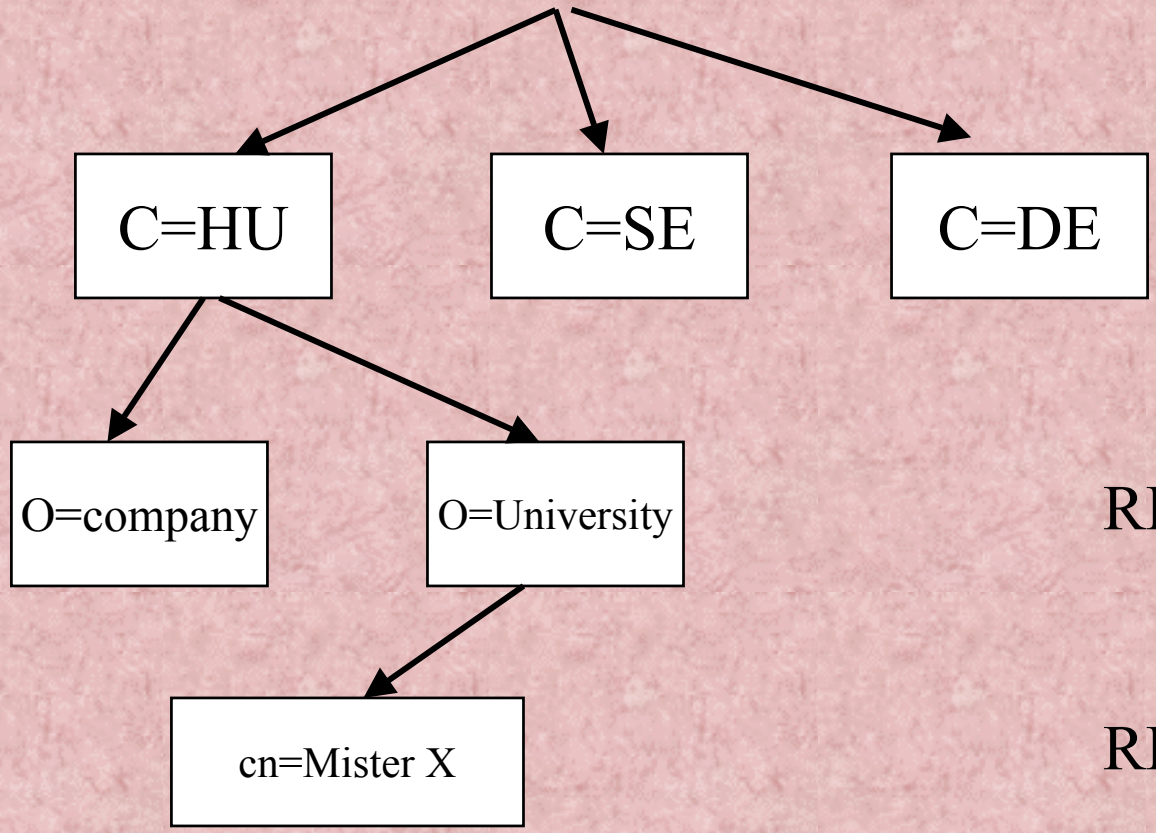
- To give humans information like
  - Data (Telephonenumbers etc.) about humans (White Pages)
  - Data (postal address etc.) about organisations (Yellow Pages)
- To give applications data in a known format for
  - Message handling
  - File transfer (File Transfer Access Management, FTAM)
  - Name mapping for X.400 addresses
- The Standard defines a set of Data fields for these purposes

D D S

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
- Data are stored in entries
- Entries are ordered as tree nodes
- In the Directory Information Tree (DIT)

Directory Information Tree (DIT)



RDN: C=HU

RDN: o=University

RDN: cn=Mister X

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

D D S

DN Distinguished Name

- A entry has a distinguished name
  - in its hierarchy level: Relative Distinguished Name (RDN)
  - all RDNs from root onwords build the Distinguished Name (DN)
- No two entries in one hierarchy level can have the same RDN
- No two entries in the whole Directory can have the same DN
- Alias Entry having a DN and pointing to another DN via aliasObjectName Attribute
- seeAlso Attribute: Entry has data and a seeAlso pointer

D D S

How is the information stored?

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

D D S

X.500 Information Model

- An Entry contains a number of Attributes
- An Attribute consists of:
  - Attribute Type
  - Attribute Value
- An Attribute Type has an associated Attribute Syntax
- The Attribute Value has to conform to that syntax
- To compare Attribute there are Matching Rules

D D S

Special Attributes

- One or more Attribute Types form the RDN
  - The Naming Attributes or
  - The Distinguished Attributes
- An Entry must have one or more Objectclass Attributes
  - It characterizes the Entry, e.g. Person
  - It defines a set of usable Attributes
    - may contain
    - must contain
- Objectclasses can inherit Attributes from other Objectclasses
- A set of Objectclasses, Attributes and Syntaxes for a special purpose are called schema

D D S

Example: **DN: cn=Mister X, o=University, c=HU**

Objectclass=top
Objectclass=person
Objectclass=organizationalPerson
cn=Mister X
cn=Xavier Xerxes
mail=X@dot.com
mail=Mister.X@dot.com
telephoneNumber=1234567

DDS

Some Objectclasses

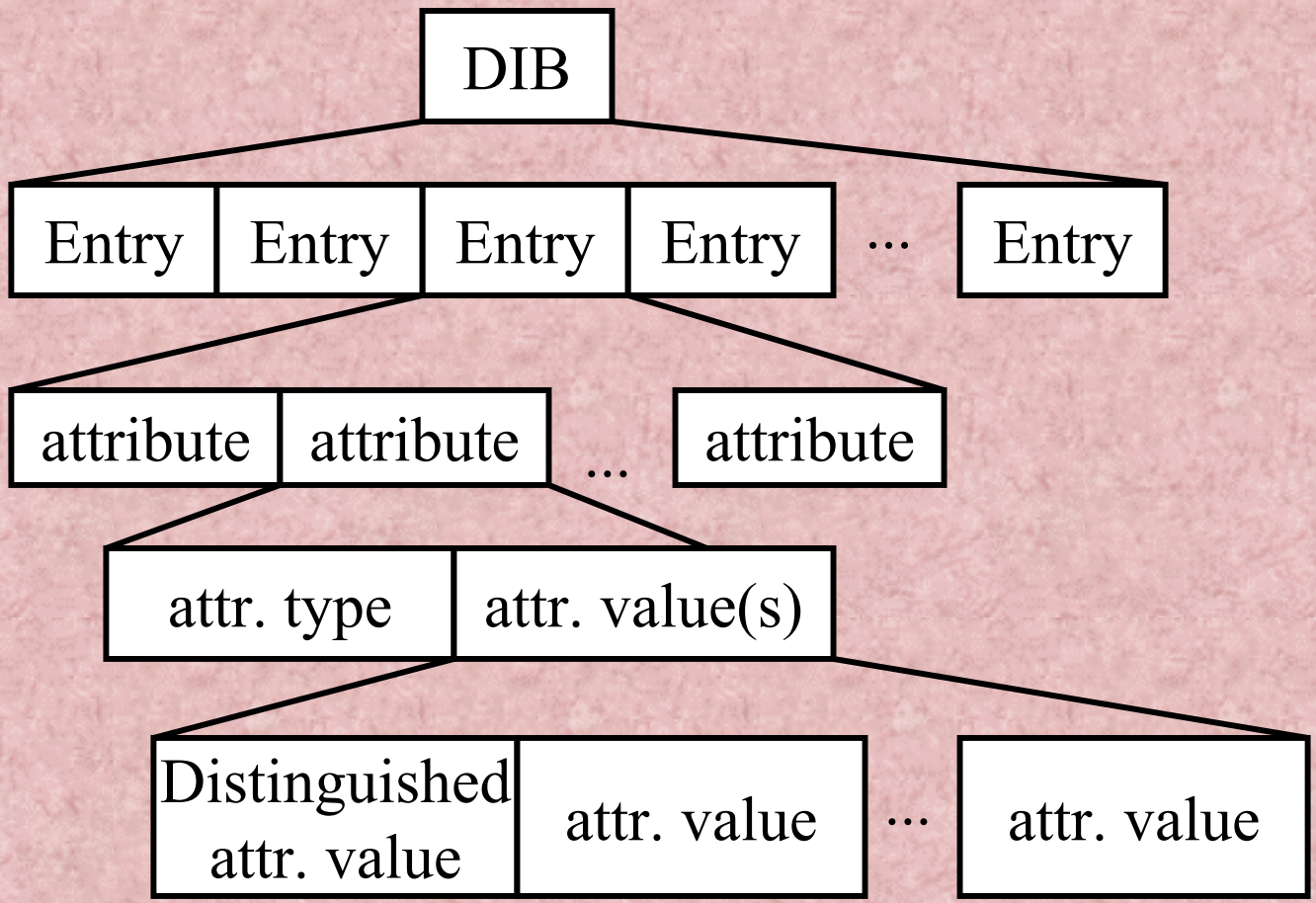
ObjectClass	distinguished Attr. and abbreviation	other Attributes
country	countryName or c	description, searchGuide, ...
locality	localityName or l	description, ...
organization	organizationName or o	description, postalAdress, ...
organizationalUnit	organizationalUnit-Name or ou	description, postalAdress,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

D D S

Open structure

- You can define your own:
  - Attribute Types
  - Attribute Syntaxes
  - Object Classes
- You can only locally use self defined schemas
- If you want them to be used globally you have to standardize them

Directory Information Base (DIB)

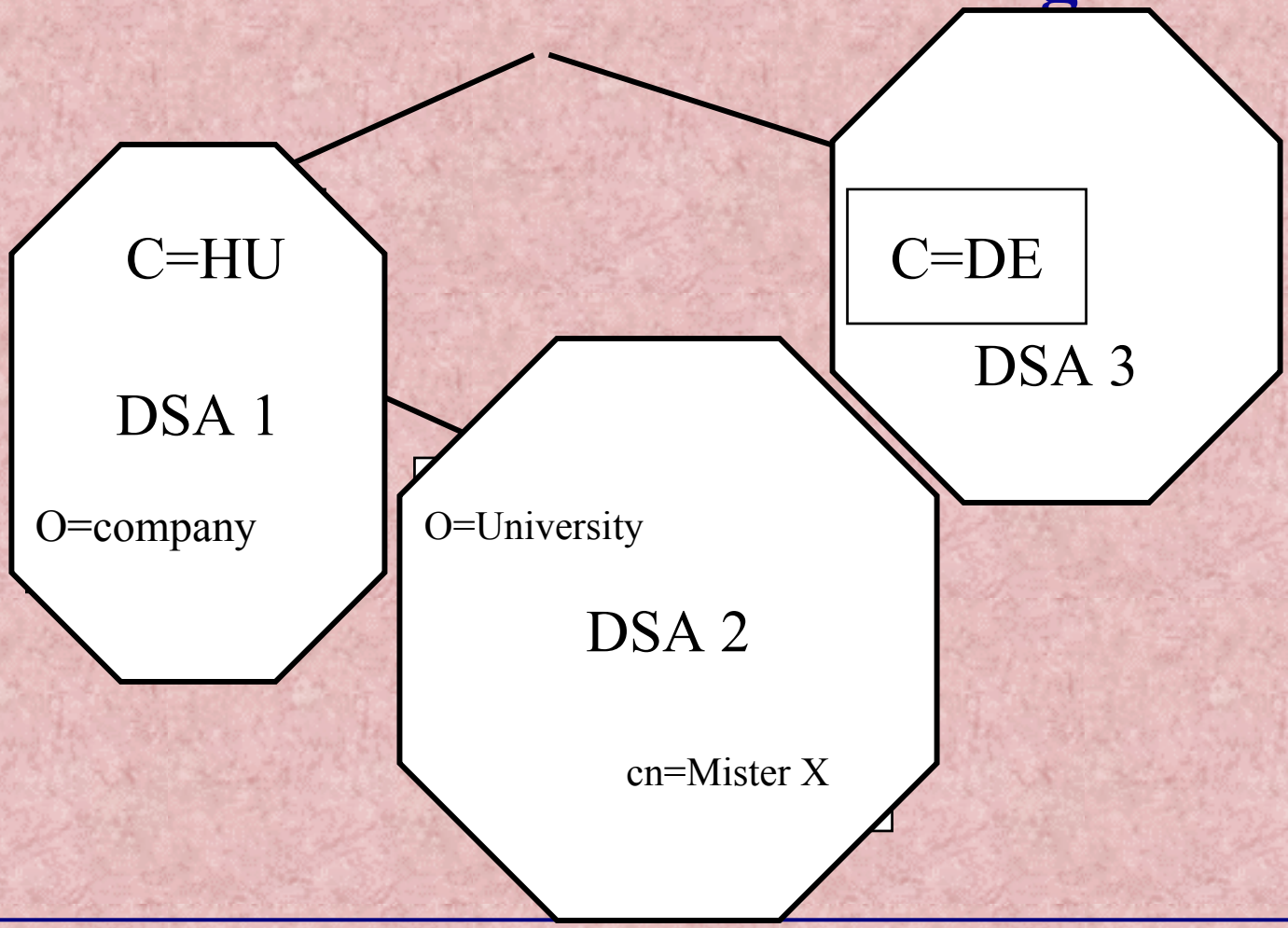


D D S

X.500 Client Server model

- Directory Service Agent DSA
  - A Server that holds directory information
- Directory User Agent
  - A client that connects to a DSA to access information
- The DUA and DAS communicate via an access protocol
- The X.500 access protocol is called Directory Access Protocol DAP
- A lightweight version of DAP is LDAP Lightweight Directory Access Protocol

Distribution of the data among DSAs

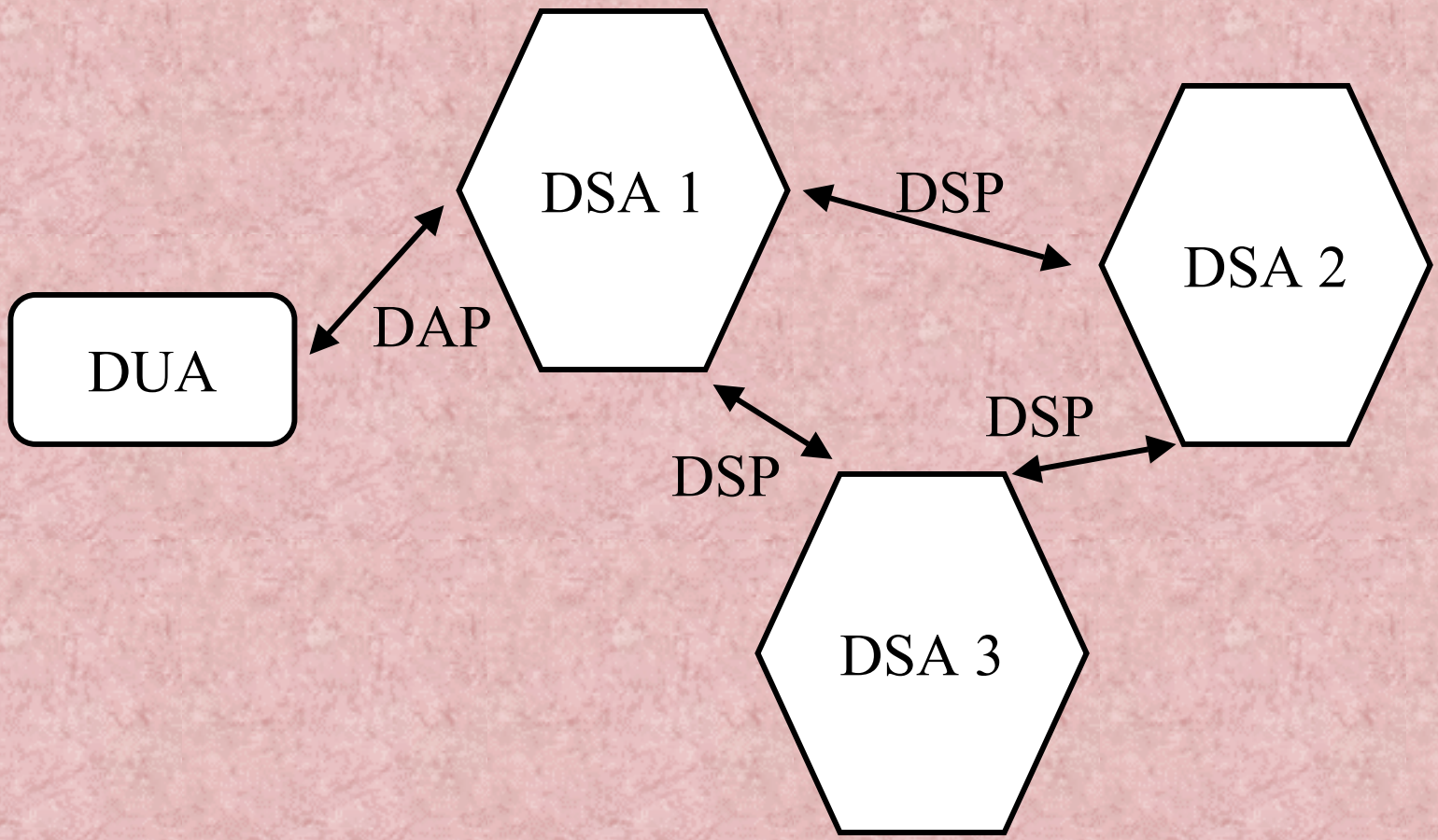


D D S

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



D D S

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
- Authentication mechanisms
- Access control

D D S

History of LDAP: LDAP v1

- A group at University of Michigan developed a Lightweight Version of DAP
  - No OSI Stack
  - Directly via TCP/IP
  - Only DUA - DAS communication
- 1993 Version 1
  - 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

D D S

LDAP v2

- 1995 Version 2
  - 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

LDAP v3 core protocol

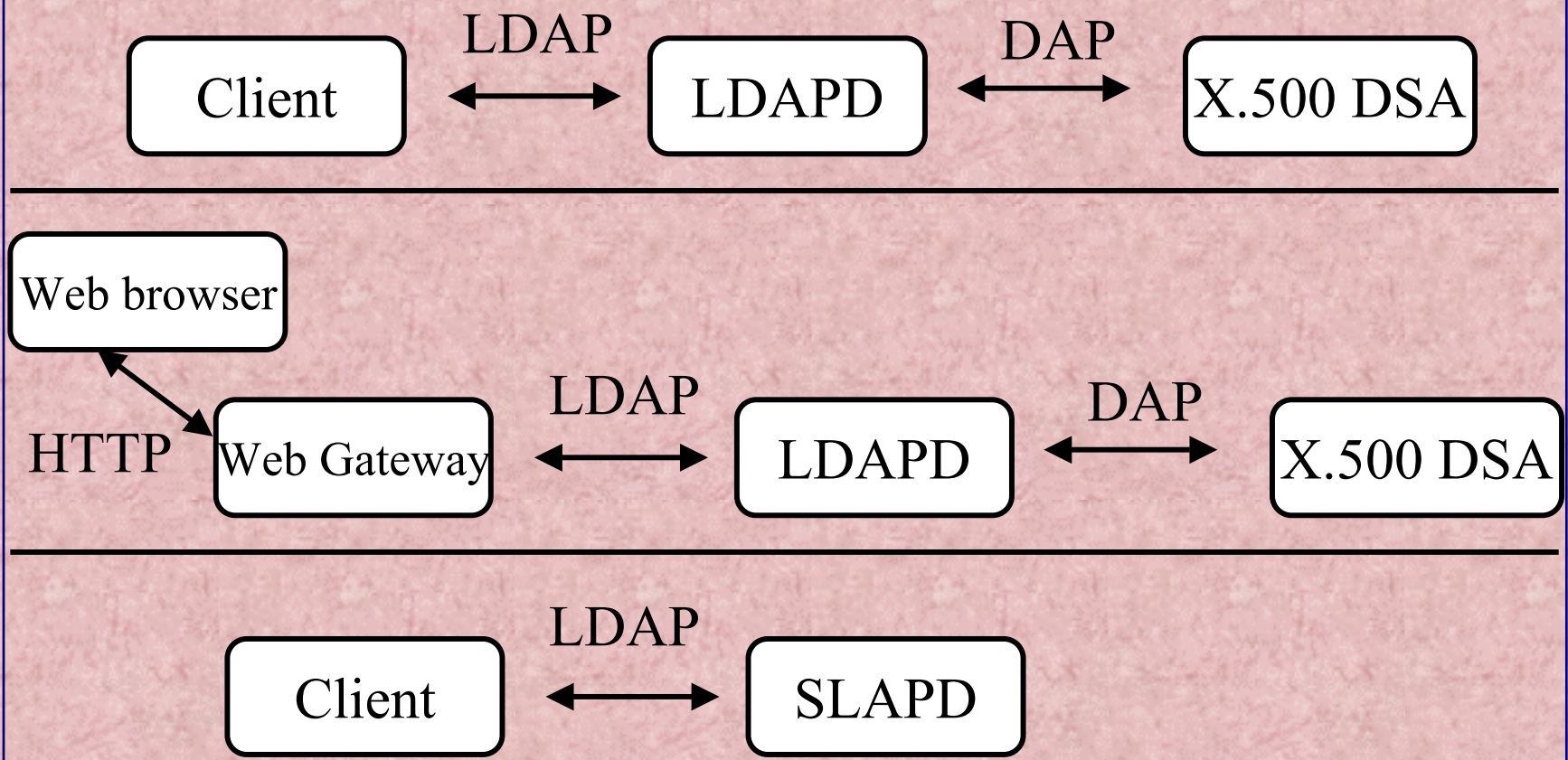
- 1997 LDAP v3
  - RFC 2251: Lightweight Directory Access Protocol (v3), M. Wahl, T. Howes, S. Kille. December 1997
  - RFC 2252: Lightweight Directory Access Protocol (v3) - Attribute Syntax Definitions, M. Wahl, A. Coulbeck, T. Howes, S. Kille. December 1997
  - RFC 2253: Lightweight Directory Access 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
  - 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

D D S

What is LDAP?

- Originally (v1,v2) 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 Clientapplications have an LDAP interface:
  - mailagents
  - browser
  - PGP clients

LDAP connectivity



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
  - 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
- LDAP v3 is more than just an access protocol:
  - like X.500 a whole client server system

D D S

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

D D S

LDAP Data Interchange Format LDIF

- Format for exchanging data
- Example:

```
dn: cn=Mister X, o=University, c=HU
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, ...
```

D D S

LDAP Naming Model

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

LDAP Functional Model

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

D D S

8 Parameters of LDAP search (1-3)

- 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)
- 4.) size limit
  - limit the number of entries to get back

8 Parameters of LDAP search (5-)

- 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

LDAP search filter

- Equality
  - e.g.: (cn=Mister X) only entries with common name equals “Mister X”
- Negation operator
  - e.g. (!(cn=Mister X)) all entries but the one with cn equals “Mister X”
- Substring
  - e.g. (cn=Mister*) all entries with cn beginning with “Mister”
- Approximate
  - e.g.: (cn~=Mister) all entries with cn sounding similiar to “Mister”
- Greater than or equal to and less than or equal to
  - 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

LDAP search filter (II)

- 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 comparision
    - (o:dn:=company) search for o=company in attributes and also in DN
- Filters can be combined
  - AND operator: & or OR operator: |
    - e.g.: (| (cn=Mister X) (sn=Xerxes)) all entries that have cn=Mister X or sn=Xerxes
    - e.g. (& (cn=Mister X) (mail=*dot.com)) only entries that have both cn=Mister X and a mail address ending with dot.com

LDAP search filter (III)

- 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)

D D S

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
- 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
  - Frame for support of different authentication mechanisms

D D S

LDAPv3 Extension mechanisms (II)

- **Extensions have to be standardized: IETF WG ldapext**
- **Big Players like Netscape (Iplanet), Microsoft and Novell very active in this WG**

LDAPv3 Extension mechanisms (III)

- **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**

D D S

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
  - SSL (Secure Socket Layer): LDAPS
    - whole session is encrypted
    - strong authentication with X.509 Certificates
  - SASL (Simple Authentication and Security Layer) mechanisms
    - TLS (Transport Layer Security) = new version of SSL
    - StartTLS operation

LDAP URL (RFC 2255)

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

D D S

Some new standardized Extensions

- RFC 2589
- RFC 2596
- RFC 2649
- RFC 2696

RFC 2589

D D S

- 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
  - Defines:
    - Client and server requirements
    - 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

D D S

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=HU/cn;lang-en?(objectclass=*)
    - add operation
    - modify operation

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

D D S

RFC 2696

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

D D S

Some current LDAPext drafts

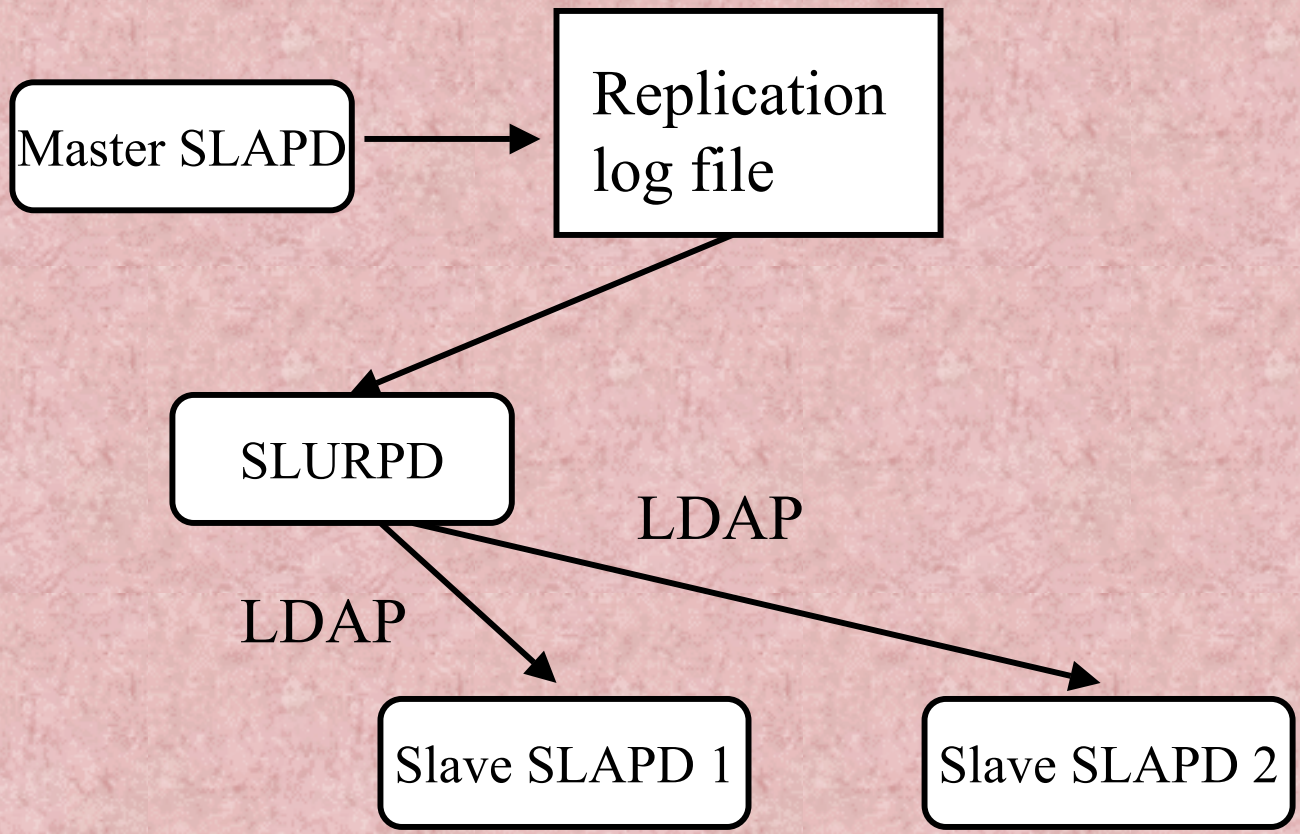
- Access Control and authentication
  - TLS extensions, X.509 Authentication with SASL
- Client Server communication
  - serverside sorting, virtual lists, persistant serach, referrals, matched values
- APIs
  - C-API and extensions, Java-API and extensions, additional error codes
- lots of individual submissions
  - LDIF, client update, MS AD, Novell NDS

D D S

Some LDAP drafts of other IETF WGs

- LDUP
  - LDAP Duplication / Replication / Update Protocols
- Policy
  - Policy Framework
  - Directory Enabled Networks (DEN)
  - Quality of Aservice (QoS)
- PKIX
  - public Key Infrastructure (X.509)
- Calsch
  - calendaring and scheduling

Non Standard LDAP Replication



Replication log file format

D D S

```
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
```

D D S

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**
  - **one policy for several distributed entries**
  - **...**

D D S

Access Control Model (I)

- Access Control Model for LDAP, E. Stokes, D. Byrne, B. Blakey. <draft-ietf-.ldapext-acl-model-o6.txt> July 2000
  - LDAP functional model (add, delete, modify and search) for the manipulation of access control information
  - additional control:
    - getEffectiveRightsRequest and -Response
  - RootDSE Attribute supportedACIMechanism with Attribute aCIMechanism
  - privileges for Attributes: read, write, search, compare
  - privileges for entries: add, delete, editDN, browseDN
  - policyOwner Attribute names who is allowed to set ACIs

Access Control Model (II)

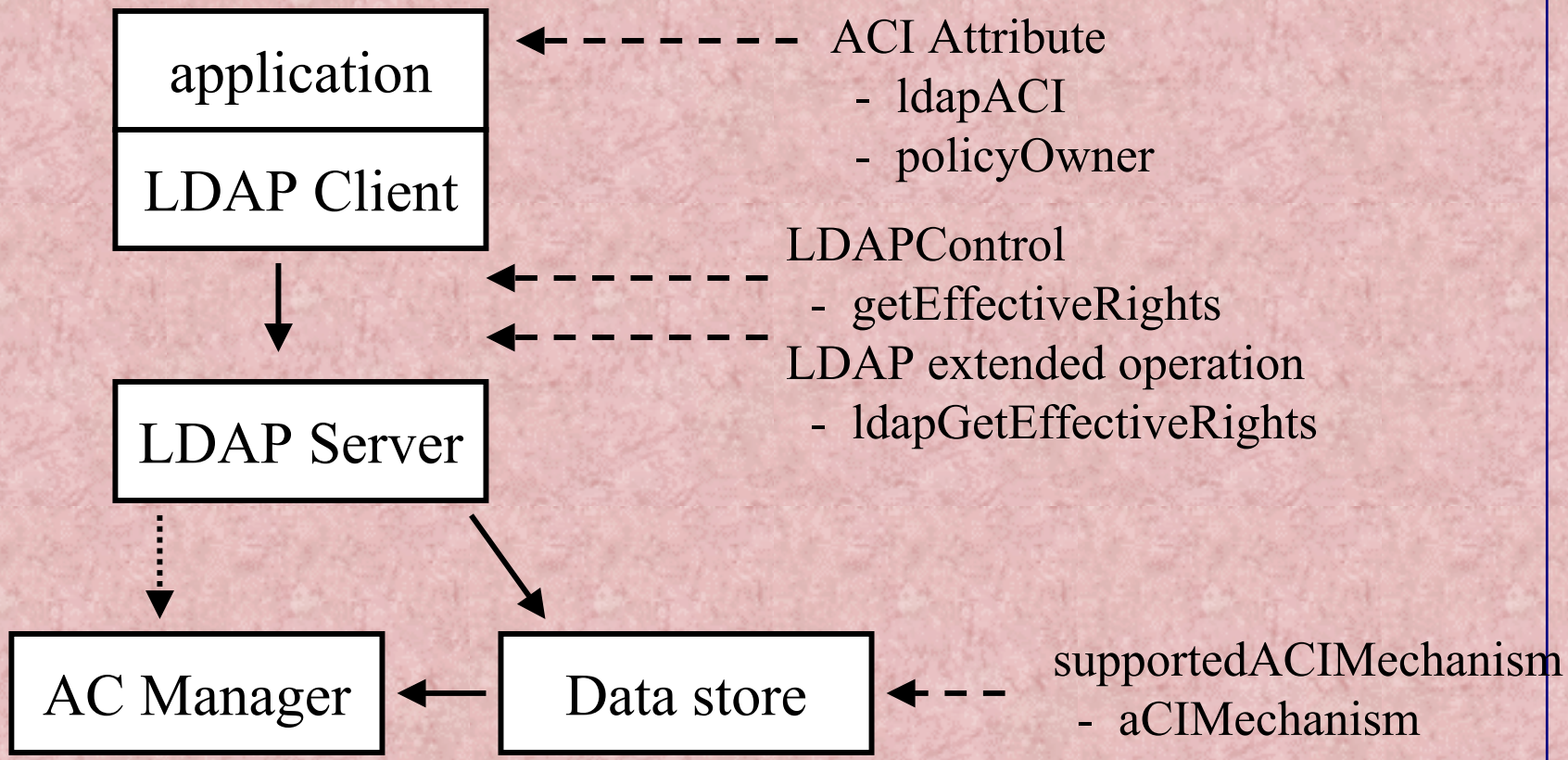
- **Basic ACI Attribute ldapACI**
  - **stores the AC information:**
    - **OID**
    - **scope (entry/ subtree)**
    - **rights (grant / deny)**
      - **grant; <permissions>; <Attribute>**
      - **permissions: a, d, r, s, w, c, e, b**
      - **Attribute: “collection”, [“all”], [“entry”]**
    - **dnType (“accessid” / “group” / “role”, ...)**
    - **subjectDN (DN / “public” / “this”)**

D D S

Access Control Model (III)

- **Examples:**
  - **a user is defined as policyOwner:**
    policyOwner: 1.2.3#subtree#access-id#cn=Mister X
  - **a group may read, search and compare an Attribute in a subbtree:**
    ldapACI: 1.2.3#subtree#grant;r,s,c;attribute attr1#group#cn=o=University,c=HU
  - **a Roleoccupant may add entries in subtree and mya read, search and compare attributes 2 and 3:**
    ldapACI: 1.2.3#subtree#grant;a;collection:[entry]# role#cn=SysAdmins,o=Company
    ldapACI: 1.2.3#subtree#grant;r,s,c;attribute:attr2# role#cn=SysAdmins,o=Company
    ldapACI: 1.2.3#subtree#grant;r,s,c;attribute:attr3# role#cn=SysAdmins,o=Company

Access Control Model (IV)



D D S

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"

D D S

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

D D S

What is Directory?

- X.500 Database standard
- ISO/ITU v1: 1988, v2: 1993, v3: 1997, v4: 2000
- Worldwide distributed data
- All data accessible worldwide
- Hierarchical organized data tree
- Objectoriented design (inheritage of objectclasses)
- extensible data model - anything goes

D D S

LDAP

- Lightweight Directory Access Protocol
- Current version: 3
- IETF standard (RFC 2251-2256)
- Not anymore only access protocol, but a full blown client server system
- All Directory implementations have LDAP interface (X.500 products, Novell NDS, M$ Active Directory)
- Lots of client applications have LDAP interface (mail user agents, browser, PGP software)

D D S

PGP key server

- First only replication of pubring via email
- Marc Horrowitz Keyserver (PKSD)
  - Started 1995
  - Own database backend
  - Email and HTTP interface
  - Operational model (add, revoke, etc.)
  - Net of server
  - Every server has all keys
  - Server synchronisation via email

D D S

PKSD Problems

- No distributed system: all keys on all server
- Permanent server synchronisation causes high bandwith usage
- Chaos when one server is down (bouncing emails)
- No guarantee that a key is replicated on all server
- Not scalable

D D S

Problems of the Web of trust

- Most keys only selfsigned (=islands of trust)
- The web of trust is only existing for people belonging to certain inner circles
- Many users don’t know what they are signing
- Even at IETF Key signing parties there is no proof of identity

D D S

New concepts for PGP key server

- PKSD with enhanced backend (Open Keyserver from Highware)
- Keyserver based on DNSSec (www.ietf.org/html-charter/dnssec-charter.html)
- Synchronisation via multicast (G. Baumer, Distributed Server for PGP Keys synchronised by multicast, www.vis.ethz.ch/~baumi/sa/thesis/thesis.html)
- Keyserver based on LDAP (PGP Certificate Server from NAI)

D D S

LDAP PGP-Keyserver requirements

- Standardizes solution
  - data model
  - operational model
- Keys searchable by different criteria
- Certification path followable
- Key status retrievable

D D S

Process of standardization

- 1994 Draft from Roland Hedberg
- 1994 proprietary solution in Tübingen
- Both models fail to include more than one certificate in a person's entry
- 1998 new initiative by DANTE
- DDS and University of Stuttgart take part in the discussion and announce an Internet Draft
- Roadmap: Draft in Summer 2000

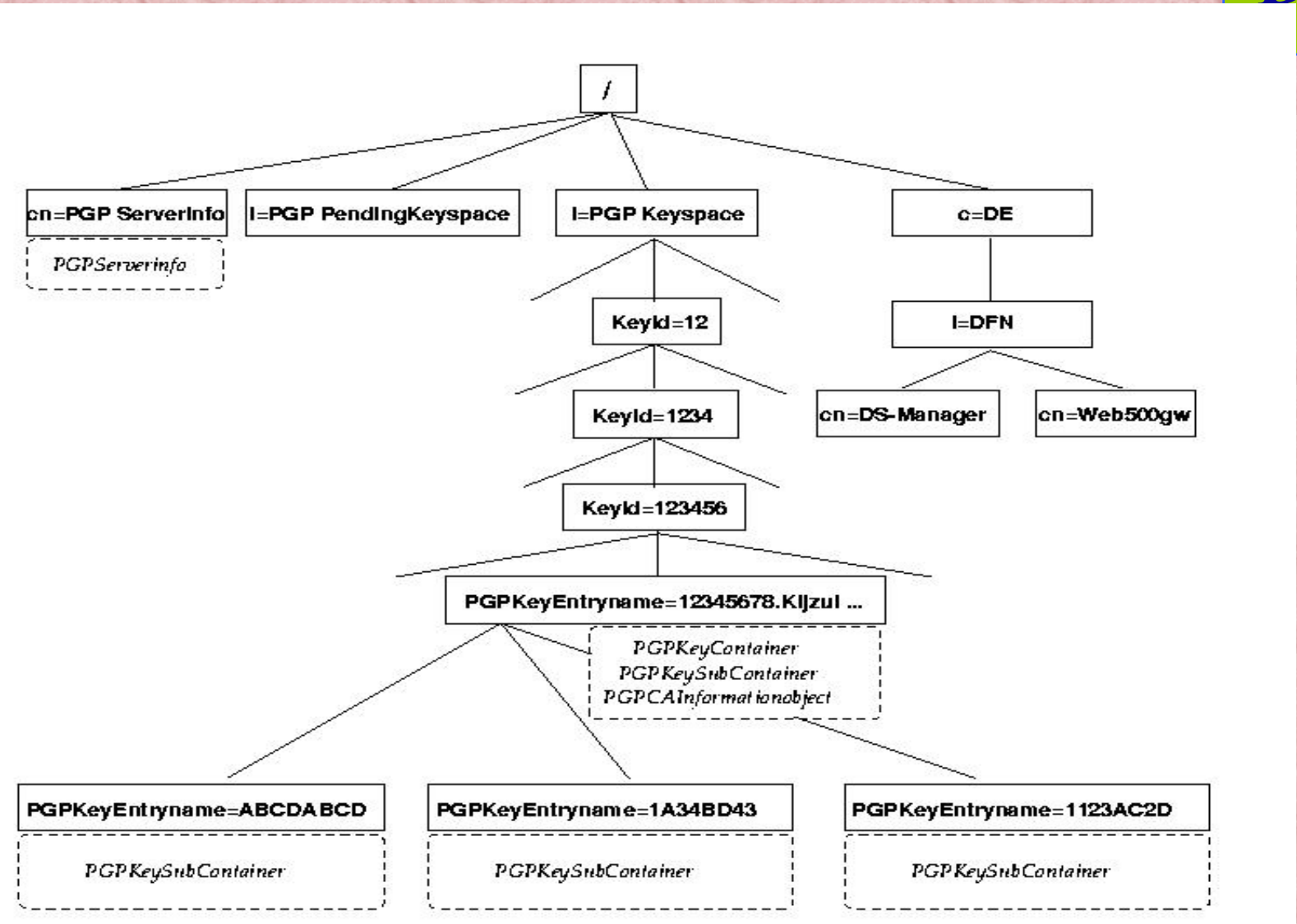
D D S

Status of LDAP PGP key server

- PGP test server based on LDAP:
  - ldap://as.directory.dfn.de:11010/l=PGP Keyspace??sub?(cn=*)
  - http://as.directory.dfn.de:11011
- Policy for a service
- Definition of a data model for PGP
- Definition of a format for CAs to send certificates
- Software for storing and retrieving certificates
- A user can store his key into the server via WWW formular

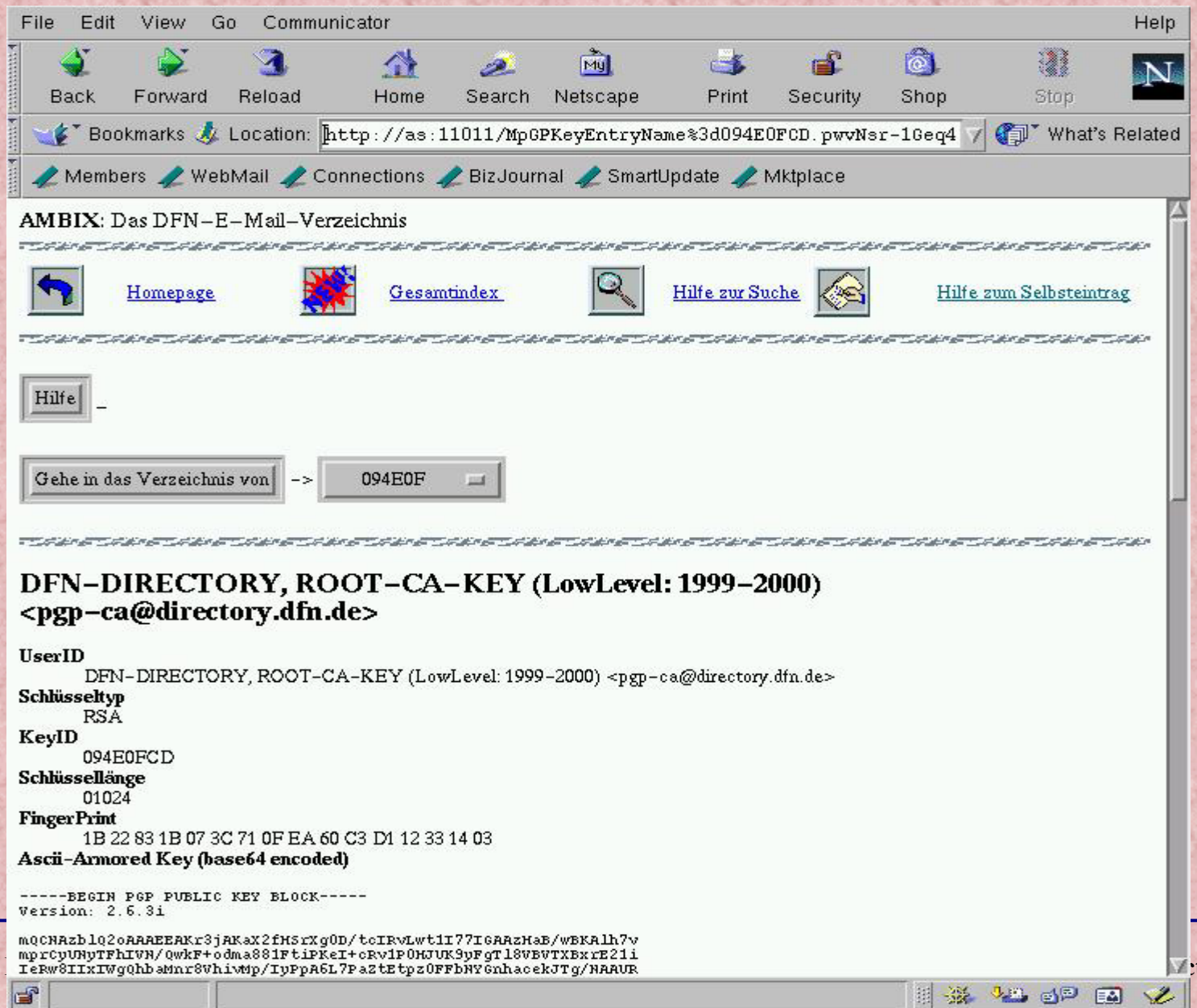
The Directory Information Tree for PGP

D D S



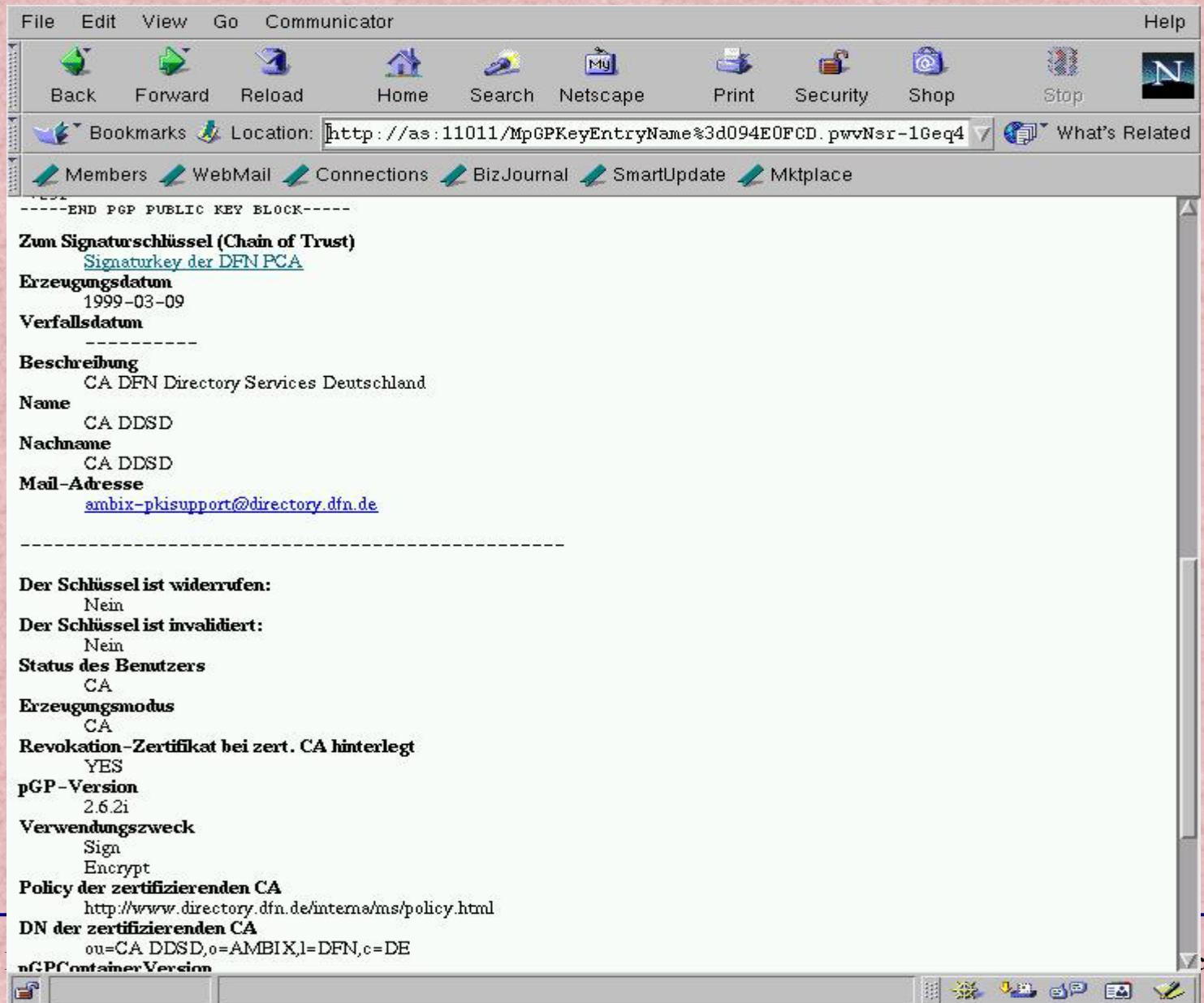
A PGP key displayed (1)

D D S



A PGP key displayed (2)

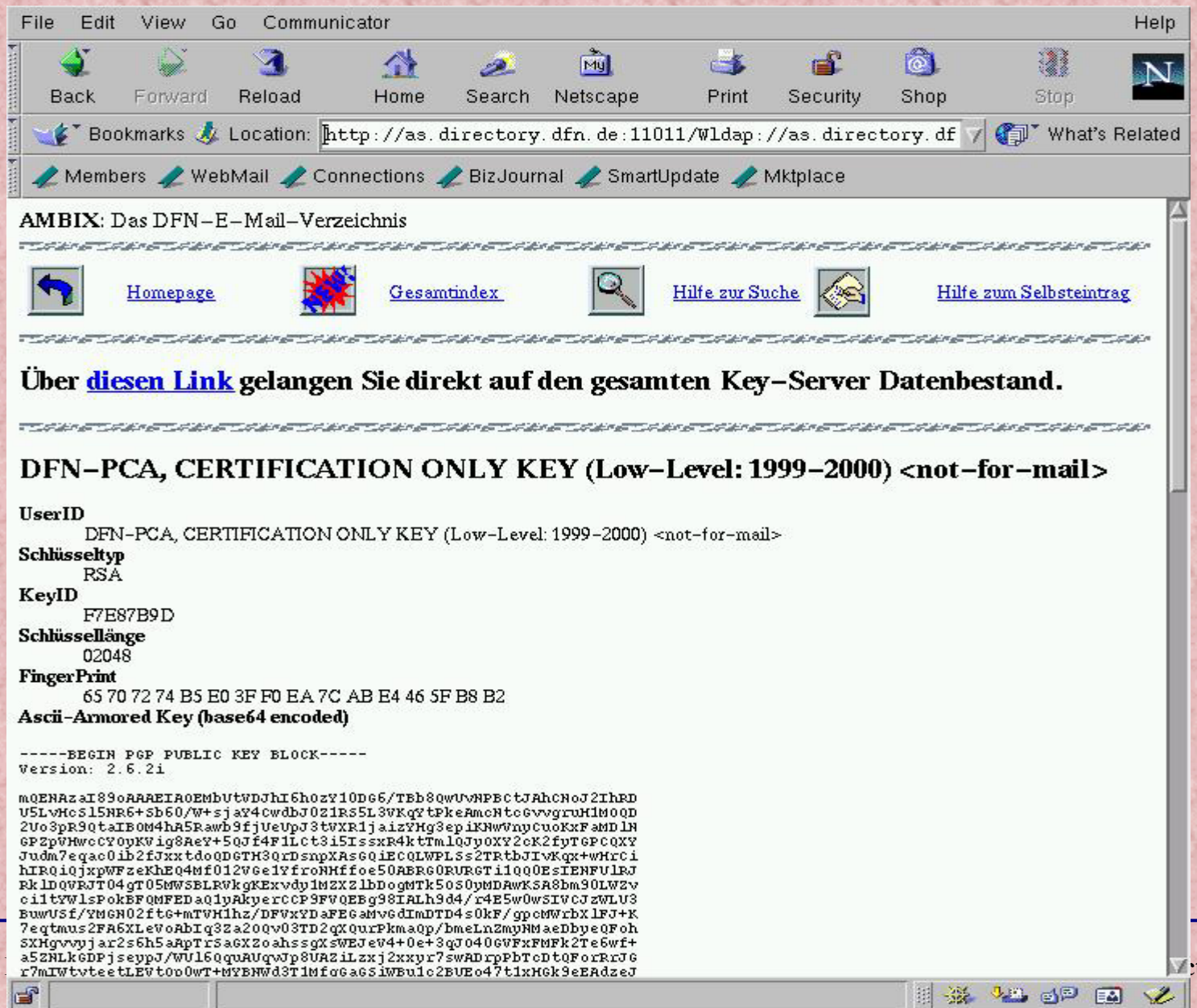
D D S



tory services

The DFN-PCA key

D D S



D D S

Objectclasses for PGP 1: pGPKeyContainer

- must contain:
  - pGPKeyEntryName (name of the entry);
  - pGPKey (ASCII-armored key)
  - pGPUserId; pGPKeyID; pGPFingerPrint
  - pGPKeySize; pGPKeyType
- may contain:
  - pGPKeyCreateTime; pGPKeyExpireTime
  - pGPKeyRevoked (0=valid, 1=revoked)
  - pGPKeyUsage
  - pGPUserDN (DN of the directory entry of the person)

D D S

Objectclasses for PGP 2: cAInformationObject

- must contain:
  - cACertKeyLink / cACertKeyURL (DN / URL of the certifying key)
  - cADN / cAURL (DN / URL of the CA, or RA)
  - cAPolicy (URL of the CA's policy)
  - cACRLDN / cACRLURL (DN / URL of the CA's CRL)

D D S

Objectclasses for PGP 3: pGPServerInfo

- must contain:
  - cn (name of the entry, always cn=pGPServerInfo)
  - baseKeySpaceDN (DN of the PGP keyspace subtree)
- may contain:
  - basePendingDN (DN of the keyspace for yet pending keys)

D D S

Current problems

- PGP ServerInfo entry has to be directly underneath the root
- Current model is not similar to the X.509 Key storage model
- Will S/MIME win the race?

D D S

Addresses and Partners

- DFN Directory Services
  - http://www.directory.dfn.de
  - mailto:dirco@directory.dfn.de
- DFN PCA
  - http://www.cert.dfn.de/dfn-pca
  - mailto:dfnpca@pca.dfn.de
- University of Stuttgart CA
  - http://ca.uni-stuttgart.de
  - mailto:info@ca.uni-stuttgart.de

D D S

X.509: The classic (1988)

- "The Directory: Authentication Framework"
- Part of the OSI-Directory standard X.500
- Defines Data model, e.g.:
  - userCertificate; cACertificate
  - crossCertificatePair
  - certificateRevocationList
- Defines mechanisms for authentification
- Certificate includes DN of the user
- Certificate includes DN of the signing CA

D D S

X.509v3 (1997)

- New extension mechanism
- Predefined extensions:
  - Information about key: identifier, usage, ...
  - Policy information: certificate policy, ...
  - User and CA extensions: alternative name, ...
  - Certification path constraints
- Lots of people see X.509v3 as independent from X.500
  - Problem: hypothetical DNs
  - No proof of uniqueness

D D S

X.509v4 (2000)

- Draft version ready (May 11, 2000)
  - ftp://ftp.bull.com/pub/OSIdirectory/4thEditionTexts/X.509_4thEditionDraftV2.pdf
  - Press release: http://www.itu.int/ITU-T/itu-t_news/sg7_x509_press.htm
- Includes verification of certificate chains with CAs from different domains and hierarchies
- Enhancements in the area of certificate revocation
- New features in attribute certificates (AC)
- Defines usage of ACs for access control and authorization

D D S

Applications of X.509 certificates

- Certificate based security on different levels:
  - Network Layer:
    - IPSec (Internet Protocol Security)
  - Transport Layer:
    - SSL (Secure Socket Layer) =
    - TLS (Transport Layer Security)
  - Application Level:
    - S/MIME (Secure Multipurpose Internet Mail Extensions) v3: patent free algorithms, mailing list support
    - PGP (Pretty Good Privacy), since version 6

D D S

IETF WG PKIX

- Defines an Internet PKI on basis of X.509 certificates
- Supports the following IETF security protocols:
  - S/MIME
  - TLS (=SSL)
  - IPSec
- Status:
  - 9 RFCs
  - 21 Internet Drafts
  - Overview: <draft-ietf-pkix-roadmap-05.txt>

PKIX and certificate profiles

- **RFC 2459 redrafted: <draft-ietf-pkix-new-part1-00.txt> defines:**
  - **Certificate (X.509v3 standard fields and standard extensions plus one private extension for authority information access, for e.g. validation service)**
  - **CRL (X.509v2 standard fields, and standard extensions)**
  - **Certificate path validation process, basic and extending**
- **<draft-ietf-pkix-acx509prof-02.txt> defines:**
  - **Attribute certificate profile for standard fields and extensions**
  - **additional attribute types**
  - **Attribute certificate validation**
  - **revocation**

D D S

PKIX and certificate profiles (contd.)

- <draft-ietf-pkix-qc-03.txt> defines:
  - Qualified Certificate
    - as prescribed by some governmental laws
    - owner is natural person
    - unmistakable identity
    - only non-repudiation as key usage
    - ...

PKIX LDAPv2 schema

- RFC 2587 “Internet X.509 Public Key Infrastructure LDAPv2 Schema”, defines:
  - Objectclass pkiUser with attribute userCertificate
  - Objectclass pkiCA with attributes cACertificate, certificateRevocationList, authorityRevocationList, crossCertificatePair
  - Objectclass cRLDistributionPoint with attributes cn, certificateRevocationList, authorityRevocationList, deltaRevocationList
  - Objectclass deltaCRL with attribute deltaRevocationList

D D S

PKIX operational protocols

- LDAPv2: RFC 2559 defines:
  - LDAP repository read
  - LDAP repository search
- LDAPv3: <draft-ietf-pkix-ldap-v3-01.txt> defines:
  - Which v3 features are needed in PKIX
  - attributeCertificate
  - certificate matching rules
- FTP/HTTP: RFC 2585
- Limited Attribute Certificate Acquisition Protocol (LAAP) <draft-ietf-pkix-laap-00.txt>

D D S

PKIX and certificate validation

- Simple Certification Verification Protocol (SCVP) <draft-ietf-pkix-scvp-01.txt>
  - Client can offload certificate validation to a dedicated (trusted) server (validity of certificate and certification path)
- Online Certificate Status Protocol (OCSP) RFC 2560
  - determination of current status of a certificate without the use of CRLs
  - question contains cert id and time
  - answer contains: “revoked”, “notRevoked” or “unknown”
- OCSP Extension <draft-ietf-pkix-ocspx-00.txt>
  - allows client to delegate processing of certificate acceptance functions to a trusted server

D D S

LDAP work on X.509: Data model

- **LDAP Object Class for Holding Certificate Information <draft-greenblatt-ldap-certinfo-schema-02.txt>**
  - **Introduces Objectclass certificateType**
  - **enables client to retrieve only those certificates that it really wants**
  - **contains attributes: typeName, serialNumber, issuer, validityNotBefore, validityNotAfter, subject, subjectPublicKeyInfo, certificateExtension, otherInfo**

D D S

LDAP work on X.509: TLS

- **LDAPv3 Extension for Transport Layer Security** <draft-ietf-ldapext-ldapv3-tls-06.txt>
  - Extended request/response for Start TLS operation
- **Authentication Methods** <draft-ietf-ldapext-authmeth-04.txt>
  - Includes (as SHOULD) certificate-based authentication with TLS
  - Client uses Start TLS operation
  - Server requests client certificate
  - Client sends certificate and performs a private key based encryption
  - Client and server negotiate ciphersuite with encryption algorithm
  - Server checks validity of certificate and its CA
  - Client binds with SASL “EXTERNAL” mechanism

D D S

Questions?

- More Info at:
  - peter.gietz@directory.dfn.de
  - www.directory.dfn.de