Einführung in LDAP und OpenLDAP

TÜBIX,

13. Juni 2015, Tübingen

Peter Gietz,
DAASI International GmbH

Agenda

- Einführung in Verzeichnisdienste
- Unterschiede zu relationalen Datenbanken
- Einführung in das LDAP Datenmodell
- Einführung in das LDAP-Protokoll
- Einführung in OpenLDAP
- Warum OpenLDAP?
- Implementierungsbeispiele

DAASI International

Einführung in Verzeichnisdienste

DAASI International

Was ist ein Verzeichnisdienst?

- *Informationen in einem hierarchischen System, z.B.:*
  - Dateiverzeichnis im Betriebssystem (MS/DOS, Unix)
  - Domain Name Service (DNS)
  - Network Information System (NIS)
  - X.500 – *das* Verzeichnis
  - Lightweight Directory Access Protocol (LDAP)
  - NetIQ (ehem. Novell) eDirectory
  - Microsoft Active Directory (AD)

DAASI International

X.500/LDAP Historie

- *X.500: ITU-T-Standard, der auf ISO/OSI-Stack aufsetzt,*
- *Erste Version 1988: X.500v1*
  - *X.509 war Sicherheitsmodell für X.500*
- *Kontinuierlich weitere Versionen*
  - *X.500 is not dead, it just smells funny*
  - *X.500v7 2012*
- *LDAP war ursprünglich ein leichtgewichtiges Zugriffsprotokoll zu X.500-Server:*
- *X.500 <-> LDAP-Gateway <->LDAP Client*
- *Erst später ist man auf die Idee gekommen einen LDAP-Server zu bauen*

DAASI International

Konzept von X.500/LDAP

- *Eine Datenbank*
  - *Hierarchische Datenstruktur*
  - *Optimiert für schnelles lesen*
  - *Einfache Updatemechanismen – keine Transaktionen*
- *Netzwerkprotokoll*
  - *Verteilung der Daten im Netz*
  - *Spiegelung der Daten im Netz*

X.500/LDAP

- *LDAP übernimmt vollständig das X.500-Datenmodell*
- *Die Funktionalität des Protokolls ist fast gleich. Wesentliche Unterschiede:*
  - *Nicht mehr OSI-Stack, sondern direkt über TCP-IP*
  - *Kein Chaining sondern Referrals*

DAASI International

Relationale Datenbanken vs. Verzeichnisdienste

Vergleich RDBMS vs. LDAP 1/3

Aspekt	Relationale Datenbank	LDAP
Informations-organisation	Entitätsinformationen werden logisch auf verschiedene Tabellen aufgeteilt und Relationen zwischen den Tabellen erstellt.	Entitätsinformationen bleiben logisch an einem Platz, nämlich in einem Informationsobjekt, das einen Knoten in einem hierarchischen Baum darstellt.
Mehrfachwerte	Mehrfachwerte eines Datenfelds erzwingen eine neue Tabelle (Normalisierung) oder verschiedene Datenfelder, zum Beispiel Telefonnummer_1, Telefonnummer_2.	Die Spezifizierung eines Datenattributs legt fest, ob es singlevalue oder multivalue ist. Im letzteren Fall lassen sich beliebig viele Mehrfachwerte speichern.
Schema	Es gibt keine standardisierten Tabellenschemata.	Es gibt international standardisierte Daten-schemata insbesondere zur Abbildung von Organisations- und Personendaten, inklusive Gruppen- und Rollenkonzepten.
Authentifizierung	Proprietäre Authentifizierungsmechanismen	Verschiedene standardisierte und über das Netz funktionierende Authentifizierungs-mechanismen.

DAASI International

Vergleich RDBMS vs. LDAP 2/3

Aspekt	Relationale Datenbank	LDAP
Vergleichs-regeln	Vergleichsregeln sind nicht Teil des Datenmodells, sondern müssen in der Anwendungslogik implementiert werden. Manche Datenbanken unterstützen es, Groß- und Klein-schreibung nicht zu unterscheiden.	Vergleichsregeln sind Teil des Daten-modells. Man kann also bei einer Schema-definition bestimmen, dass bei Wertver-gleichen etwa von Telefonnummern nur die Ziffern, nicht aber Trenn- und Leerzeichen unterschieden werden. Es gibt mehrere Vergleichsregeln für Gesamtstring- und Teilstring-Vergleiche.
Flexibilität	Änderungen des Datenschemas, also der Tabellenstruktur, sind nachträglich nur schwer möglich, da meist die Anwendungslogik, mindestens aber die SQL-Abfragen anzupassen sind.	Änderungen des Datenschemas sind ein-fach möglich: Man fügt einem Informations-objekt eine neue Objektklasse hinzu und kann dann entsprechende neue Attribute speichern. Änderungen betreffen jeweils nur die gewünschten Informationsobjekte. Die Anwendungslogik, die die neuen Attribute nicht verwendet, darf bleiben, wie sie ist.

Vergleich RDBMS vs. LDAP 2/3

Aspekt	Relationale Datenbank	LDAP
Netzwerkzugriff	Es gibt kein standardisiertes Netzwerkprotokoll, jede Implementierung kann also nur über einen passenden Client angesprochen werden. Allerdings wurde ODBC als standardisierte Datenbankschnittstelle entwickelt.	Das Netzprotokoll ist wesentlicher Bestandteil des LDAP-Standards. Unterschiedliche Clients können auf denselben Datenbestand zugreifen. Eine Verteilung der Daten im Netz ist einfach möglich.
Abfragesprache	Mächtige und standardisierte Abfragesprache SQL, die auch Joins erlaubt, die Schnittmengen aus verschiedenen Tabellen bilden können.	Mächtige Suchmöglichkeiten mit standardisierter Syntax für LDAP-Filter. Joins sind nicht möglich, weshalb Schnittmengen in der Anwendungslogik implementiert werden müssen.
Transaktionen	Es gibt Transaktionen, das heißt, mehrere Operationen an mehreren Tabellen werden als Ganzes nur durchgeführt, wenn alle Operationen fehlerfrei waren.	Es gibt keine Transaktionen. Allerdings lassen sich mehrere Änderungen an einem Eintrag als eine Operation durchführen, wodurch sie ebenfalls nur durchgeführt wird, wenn alle Änderungen fehlerfrei waren.

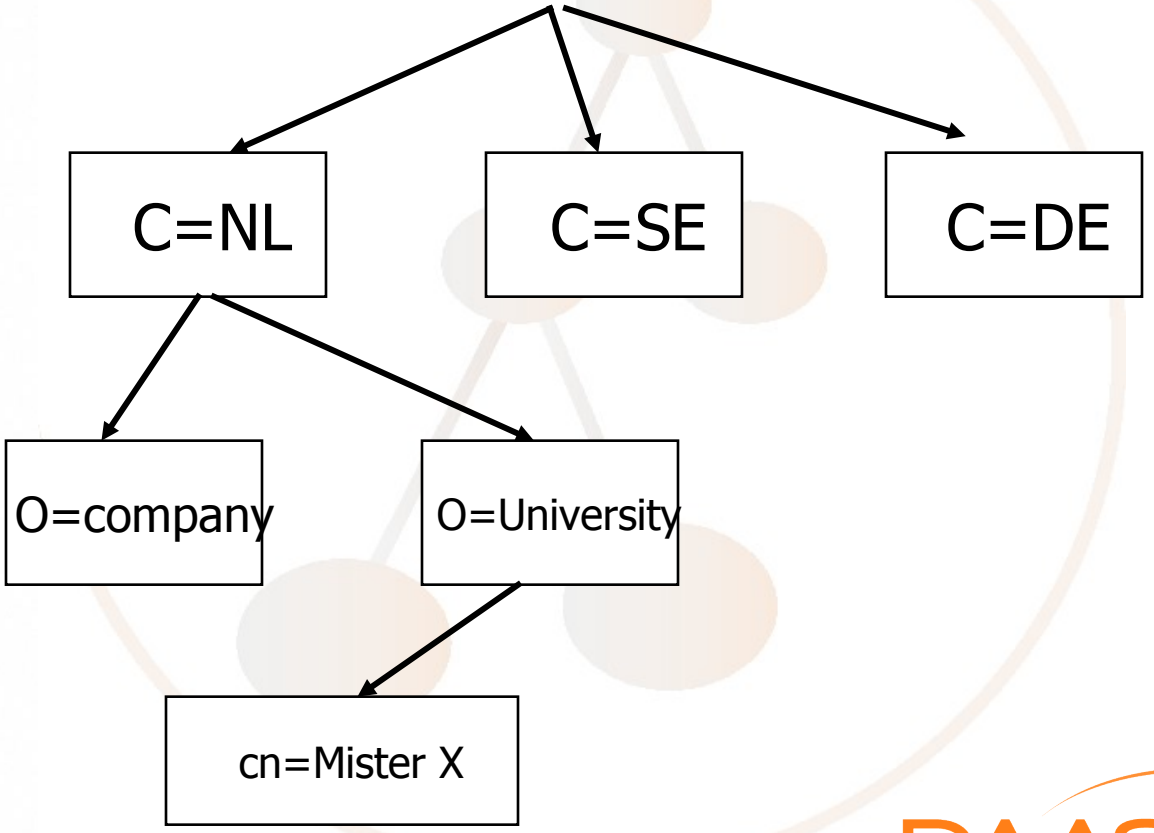
Das LDAP-Datenmodell

Directory Information Tree (DIT)

- Daten werden in Einträgen gespeichert
- Einträge werden als Baumknoten gespeichert
  - Jeder Knoten hat 0 bis n Kinderknoten
  - Jeder Knoten hat genau 1 Elternknoten
  - Mit Ausnahme des Wurzelknotens

DAASI International

Directory Information Tree (DIT)

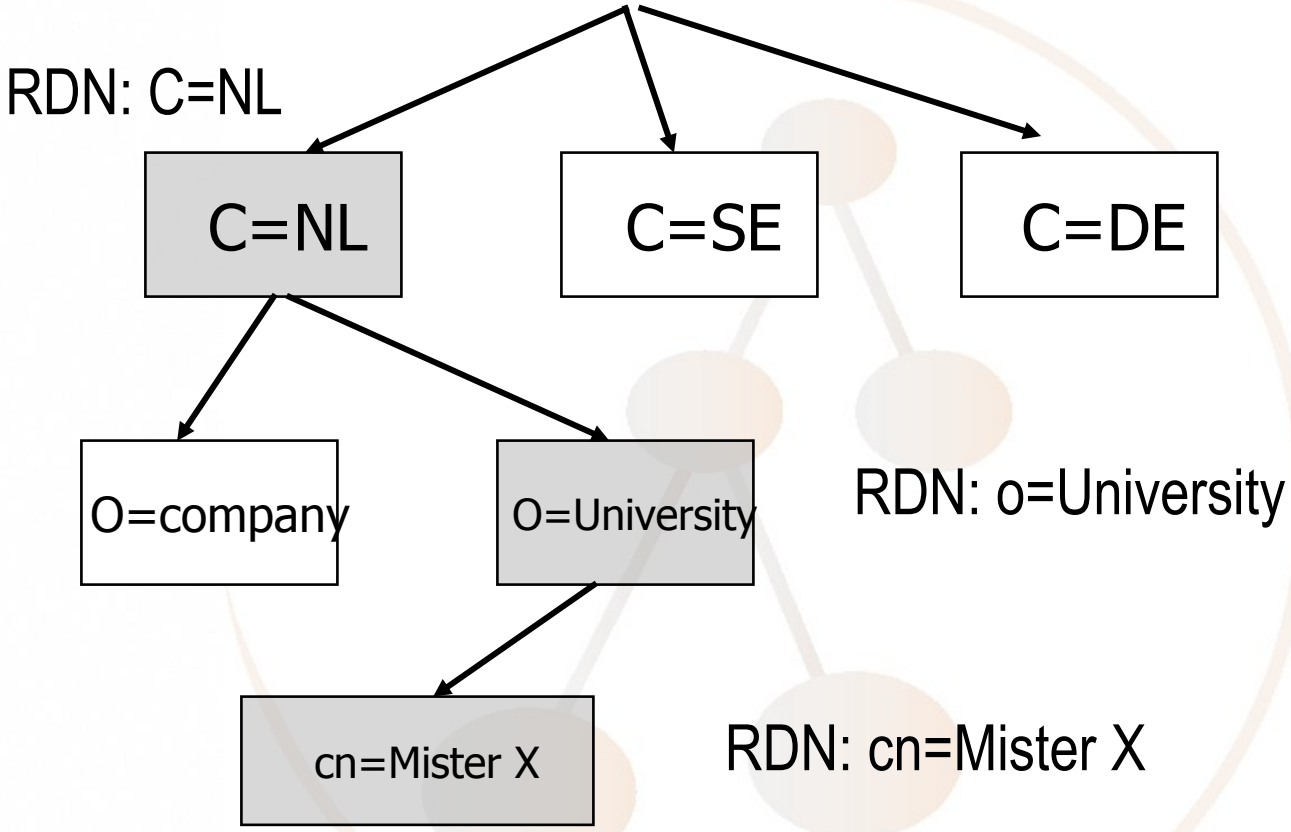


Distinguished Name (DN)

- Jeder Eintrag hat einen eindeutigen Namen
  - In der eigenen Hierarchieebene: Relative Distinguished Name (RDN)
  - Alle RDNs auf dem Pfad von der Wurzel zum Eintrag bilden zusammen den Distinguished Name (DN)
- Keine zwei Geschwistereinträge (also mit gemeinsamen Elternknoten) dürfen den gleichen RDN haben
- Demnach hat kein Eintrag im gesamten Baum einen gleichen Namen

DAASI International

Relative Distinguished Name (RDN)
Distinguished Name (DN)

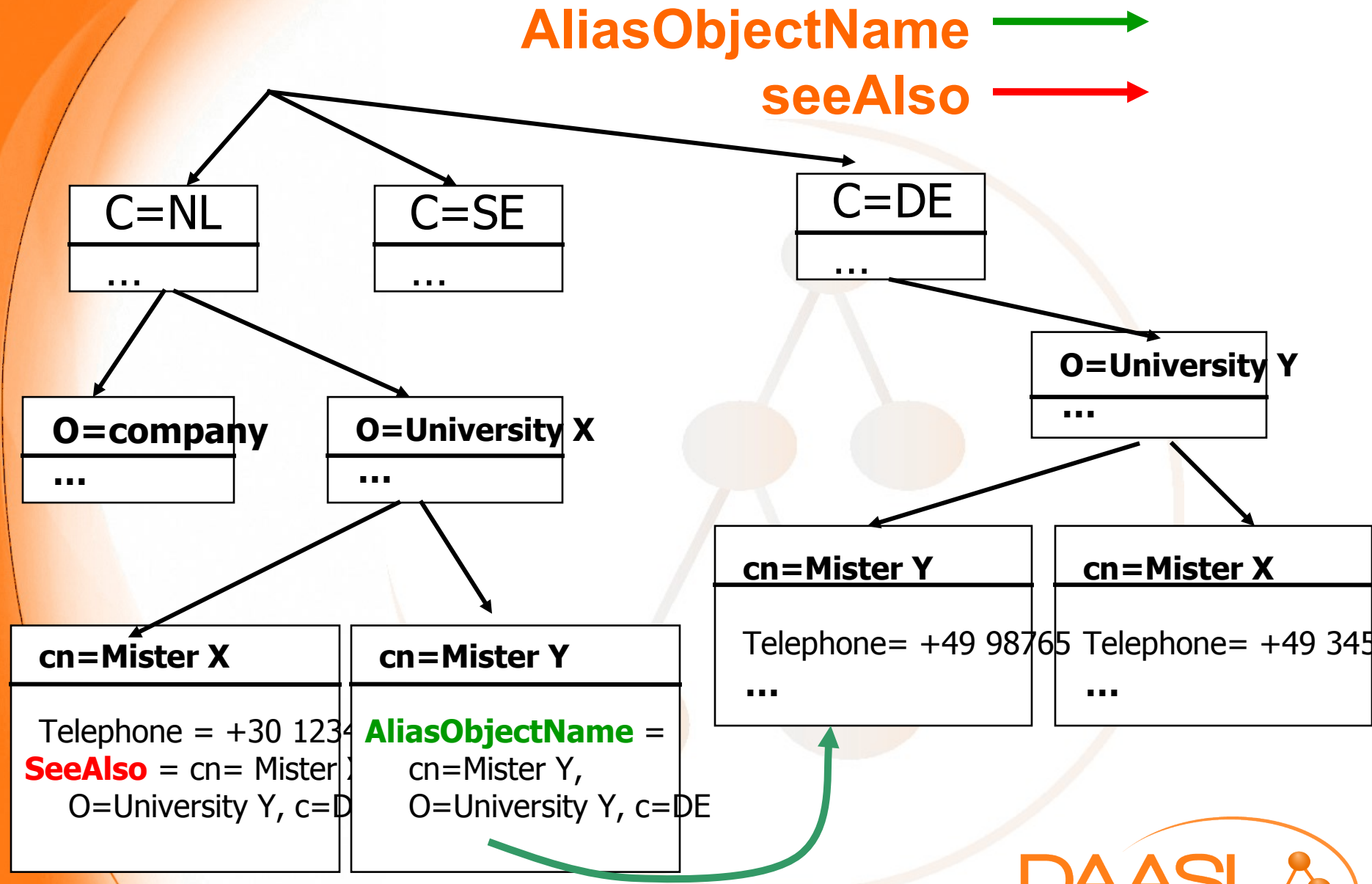


DN: c=NL;o=University;cn=Mister X

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

DN Pointer

- Alias Einträge haben einen DN zeigen auf einen weiteren DN
- seeAlso Einträge enthalten eigene Daten und zusätzlich einen DN Zeiger auf einen weiteren Eintrag



LDAP Informationsmodell

- Ein Datensatz wird Eintrag (entry) genannt
- Ein Eintrag besteht aus Attributen
- Ein Attribut besteht aus Attributtyp und Attributwert
- Es kann als Single- oder Multivalued definiert werden
- Ein Attributtyp hat eine zugehörige Attributsyntax
- Der Attributwert unterliegt dieser Syntax
- Zusätzlich kann ein Attributtyp verschiedene Vergleichsregeln (Matching Rules) haben:
  - Equality
  - Substring
  - Ordering
  - Extensible (selbstdefiniert)

DAASI International

Spezielle Attribute

- Ein oder mehrere Attribut-Typ-Wert-Paare bilden den RDN
  - *Naming Attribute* oder
  - *Distinguished Attribute*
- Jeder Eintrag muss mindestens ein *Objektklassen-Attribut* haben, welches
  - Den gesamten Eintrag charakterisiert
  - Einen Satz zu verwendender Attributtypen spezifiziert (*Must und May-Attribute)*
- Objektklassen können Attributtypen von übergeordneten Objektklassen ererben

Operationale Attribute

- Operationale Attribute werden vom Server selbst verwaltet
- Z.B.:
  - modifiersName
  - modifyTimestamp
  - creatorsName
  - createTimestamp
  - hasSubordinates

LDAP-Objektklassen

- Eine Objektklassendefinition spezifiziert eine LDAP-Objektklasse
- Die Objektklasse spezifiziert v.a. welche Attribute in einem von der Objektklasse modellierten Eintrag vorhanden sein müssen (MUST-Attribute) und welche vorhanden sein dürfen (MAY-Attribute)
- Man kann eine Objektklasse von einer anderen ableiten
  - die abgeleitete Klasse erbt alle Eigenschaften (Attribute) der Oberklasse, z.B.:
    - objectclass person enthält surname
    - organizationalPerson hat zusätzliche Attribute, wie RoomNumber. Surname wird aus person geerbt

DAASI International

Attributsyntaxen

Name	OID	Beschreibung
Binary	1.3.6.1.4.1.1466.115.121.1.5	BER/DER Daten
Boolean	1.3.6.1.4.1.1466.115.121.1.7	Boolscher Wert
DistinguishedName	1.3.6.1.4.1.1466.115.121.1.12	DN
DirectoryString	1.3.6.1.4.1.1466.115.121.1.15	UTF-8 string
Generalized Time	1.3.6.1.4.1.1466.115.121.1.24	Zeitabgabe YYYMMDDhhmmssZ
IA5String	1.3.6.1.4.1.1466.115.121.1.26	ASCII string
Integer	1.3.6.1.4.1.1466.115.121.1.27	Integer
NumericString	1.3.6.1.4.1.1466.115.121.1.36	Numerischer String
OID	1.3.6.1.4.1.1466.115.121.1.38	Object Identifier
Octet String	1.3.6.1.4.1.1466.115.121.1.40	Beliebige Octets
PrintableString	1.3.6.1.4.1.1466.115.121.1.44	Einfacher String mit alphanumerischen Zeichen und den Sonderzeichen " ( ) + - , . / : ? und blank

Matching Rules

Name	Context	Description
booleanMatch	equality	Boolean
objectIdentiferMatch	equality	OID
distinguishedNameMatch	equality	DN
uniqueMemberMatch	equality	DN with optional UID
numericStringMatch	equality	numerical
numericStringOrdering	ordering	numerical
numericStringSubstringsMatch	substrings	numerical
caseIgnoreMatch	equality	case insensitive, space insensitive
caseIgnoreOrderingMatch	ordering	case insensitive, space insensitive
caseIgnoreSubstringsMatch	substrings	case insensitive, space insensitive
caseExactMatch	equality	case sensitive, space insensitive
caseExactOrderingMatch	ordering	case sensitive, space insensitive
caseExactSubstringsMatch	substrings	case sensitive, space insensitive
caseIgnoreIA5Match	equality	case insensitive, space insensitive
caseIgnoreIA5OrderingMatch	ordering	case insensitive, space insensitive
caseIgnoreIA5SubstringsMatch	substrings	case insensitive, space insensitive
caseExactIA5Match	equality	case sensitive, space insensitive
caseExactIA5OrderingMatch	ordering	case sensitive, space insensitive
caseExactIA5SubstringsMatch	substrings	case sensitive, space insensitive

Aus: Adam Tauno Williams, LDAP and OpenLAP
ftp://kalamazoolinux.org/pub/pdf/ldapv3.pdf

DAASI International

3 Objektklassen Typen

- ABSTRACT
  - wird nur für die Vererbungshierarchie verwendet
  - darf nicht allein instanziiert werden
  - ein Eintrag darf nicht nur von abstrakten Objektklassen modelliert werden
- STRUCTURAL
  - definiert die Hauptcharakteristik eines Eintrags, wie z.B. Person, Organisation, etc.
  - ein Eintrag darf nur eine strukturelle Objektklasse, bzw. deren Vererbungshierarchie enthalten
- AUXILIARY
  - Hilfsklasse, die einem Eintrag zusätzliche Eigenschaften modelliert
  - z.B.: PKIUser mit Attribut userCertificate

DAASI International

Objektklassendefinition nach RFC 2252

```
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 ")"
```

Objektklassendefinition Beispiele

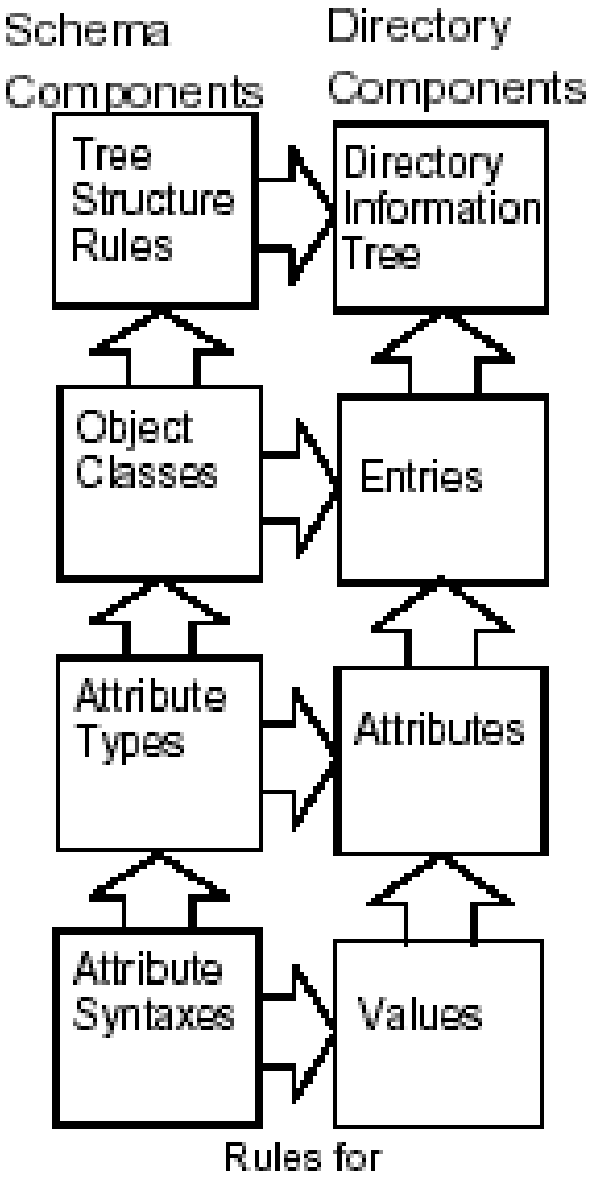
```
( 2.5.6.0 NAME 'top' ABSTRACT MUST objectClass )

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

( 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 )
)
```

telephoneNumber wird doppelt referenziert!

Schemaregeln und Verzeichniskomponenten



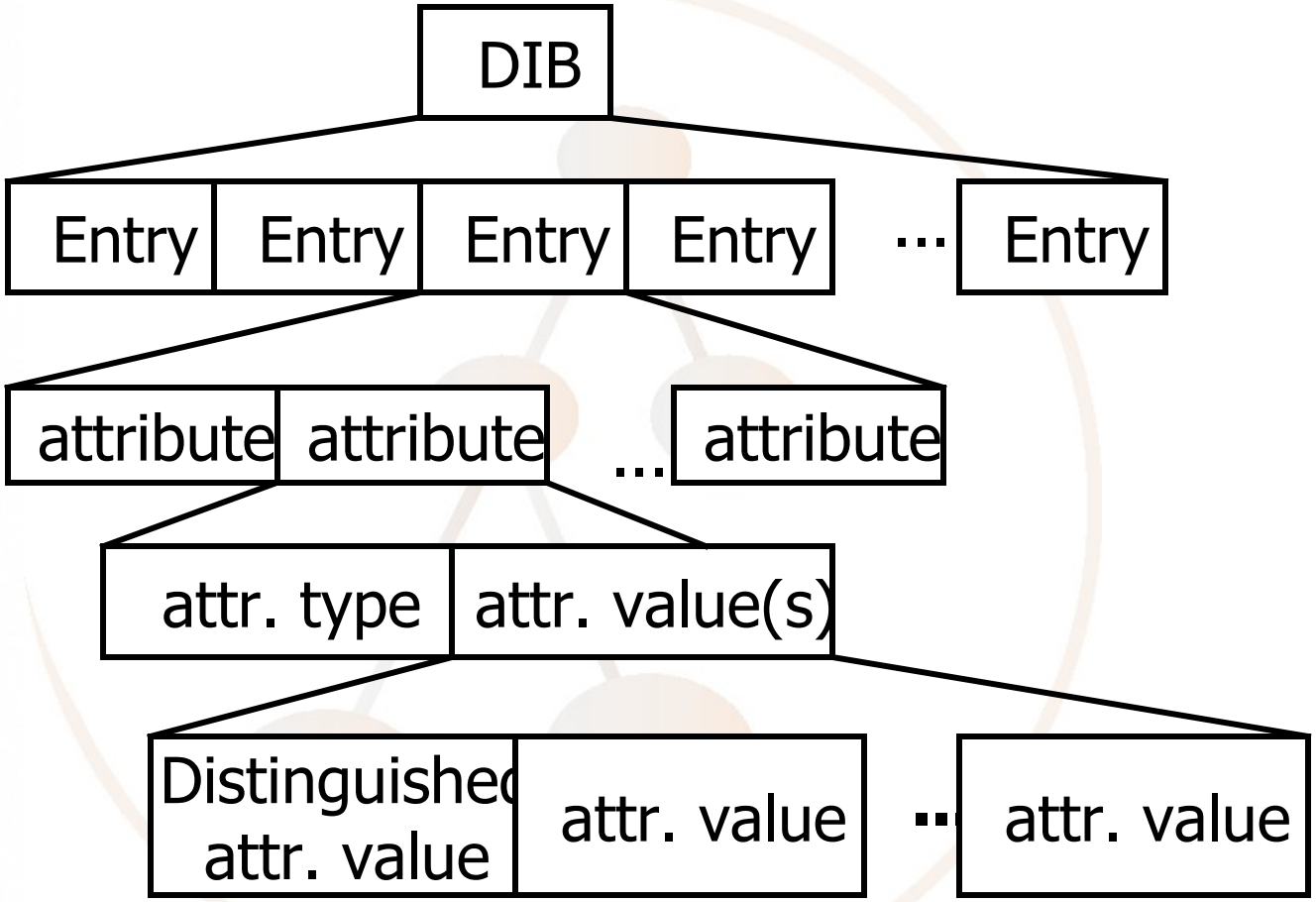
Aus: Michael MacIsaac - IBM,
Directory Solutions Using OpenLDAP,
http://linuxvm.org/present/SHARE102/s9207mma.pdf

Schema

- Eine Ansammlung von Objektklassen, Attributen, Syntaxen und Matching Rules, die für einen bestimmten Zweck definiert wurden, werden *Schema* genannt
- Jedes Schemaelement (Attributtypen, Objektklassen, Syntaxen, Matchingrules, etc.) hat eine weltweit eindeutige Nummer (Object Identifier, OID) mit der es identifiziert werden kann

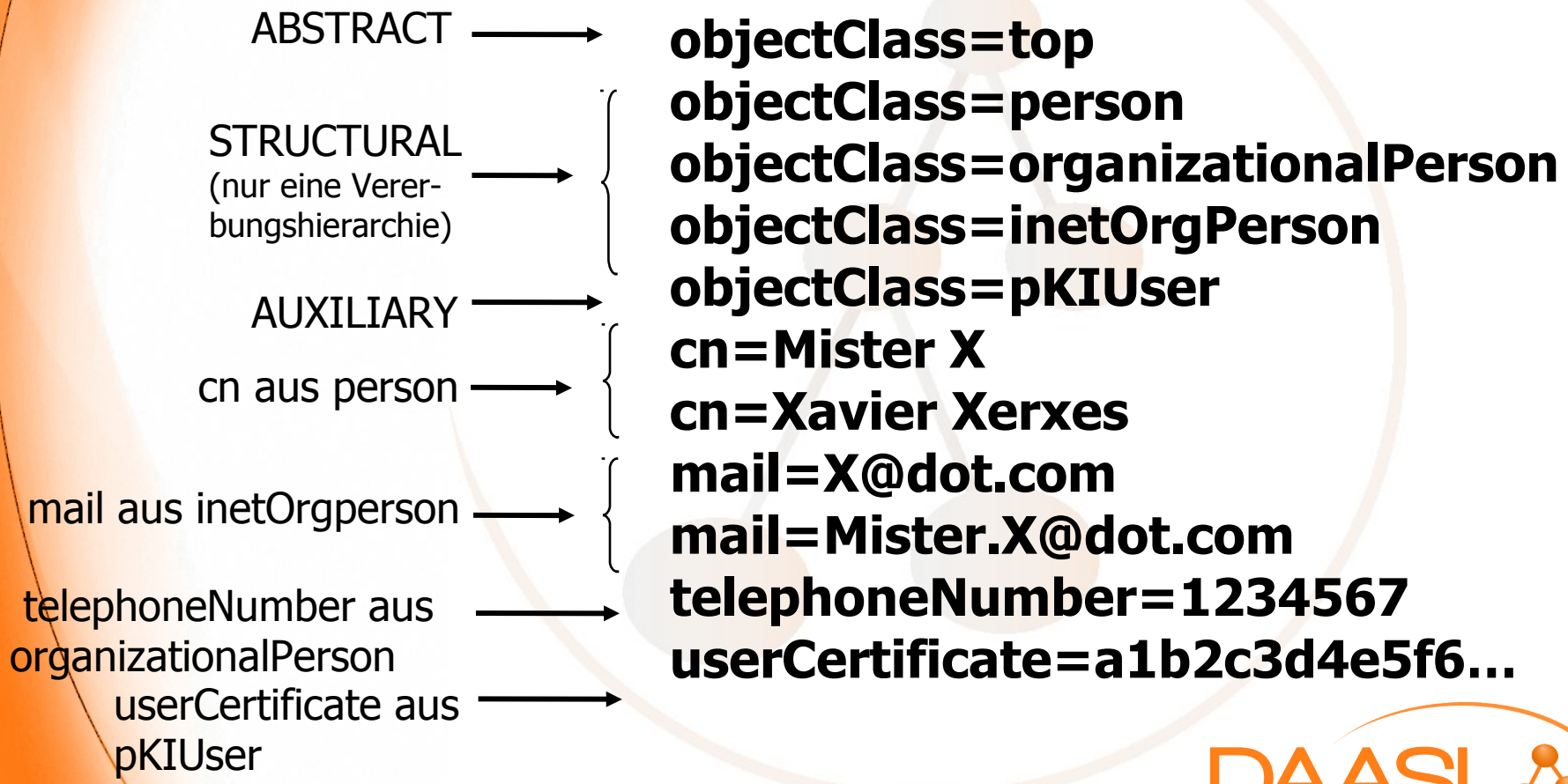
DAASI International

Directory Information Base



Beispiel

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



LDIF (RFC 2849)

- LDAP Data Interchange Format
- ASCII-Format zum Datenaustausch
  - auch für delete und modify
- Beispiel:

```
dn: cn=Mister X, o=University, c=NL
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Mister X
cn: Xavier Xerxes
mail: X@dot.com
mail: Mister.X@dot.com
telephoneNumber: 1234567

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

Offene Struktur

- Man kann eigenes Schema definieren
  - Objektklassen
  - Attribute
  - [Syntaxen]
  - [Matching Rules]
- Lokal kann man selbstdefiniertes Schema einfach verwenden
- Wenn das Schema global genutzt werden soll muss man es
  - Standardisieren (IETF-RFC)
  - Oder wenigstens registrieren (s.u.)

Das LDAP Protokoll

Funktionsmodell

- Authentifizierungs-Operationen:
  - bind
  - unbind
  - abandon
- Abfrage-Operationen:
  - search
  - compare
- Update-Operationen:
  - add
  - delete
  - modify
  - modifyDN

LDAP-Search-Parameter 1/3

1. base object or base DN (Basiseintrag)
   - Knoten im DIT von wo aus die Suche durchgeführt werden soll
2. scope (Suchbereich)
   - base (Lese nur den Eintrag der durch den BaseDN spezifiziert wurde)
   - onelevel (Suche nur direkt unterhalb des durch den BaseDN spezifizierten Knotens)
   - subtree (Suche im gesamten durch den BaseDN spezifizierten Teilbaum)
3. derefAliases (Aliasobjekte verfolgen)
   - neverDerefAlias (Dereferenziere keine Aliases bei der Suche oder dem Lesen des Base-Eintrags)
   - derefInSearching (Dereferenziere Aliases nur bei Suchen unterhalb des Base-Eintrags)
   - derefFindingBaseObject (Dereferenziere Aliases nur beim Lesen des Base-Eintrags)
   - derefAlways (Dereferenziere Aliase immer)

LDAP-Search-Parameter 2/3

1. size limit
   - Begrenzung der Anzahl der Einträge, die vom Server zurückgegeben werden sollen
2. time limit
   - Zeitbegrenzung nach der der Server alle bis dahin gefundenen Einträge zurückgeben soll
3. attrsOnly
   - Boolscher Wert. Wenn gesetzt, werden nur die gefüllten Attributnamen angezeigt, nicht aber die Werte.

LDAP-Search-Parameter 3/3

1. Filter
   - Suchfilter; in der einfachsten Form: "(<Bedingung>)"
   - wobei Bedingung aus einem Attributtyp einem Operator und einem Wert besteht
   - (s.u.)
2. attributes
   - eine Liste kommaseparierter Attributtypen, die zurückgegeben werden sollen
   - z.B.: cn, telephoneNumber
   - Attributtypen können auch als OID spezifiziert werden, z.B.: 2.5.4.3, 2.5.4.20
   - * bedeutet hierbei alle User-Attribute
   - 1.1 (OID die sicherlich kein Attributtyp bezeichnet) bedeutet: keine Attributtypen zurückgeben (also nur DN der gefunden Einträge)

Search-Filter-Operatoren 1/2

- Equality: "="
  - Voraussetzung: Attributdefinition hat eine Equality Matching Rule
  - " (cn=Mister X)" alle Einträge, die ein CN-Attribut mit dem Wert “Mister X” haben
- Negationsoperator: "!"
  - " (!(cn=Mister X))" alle Einträge, die kein CN-Attribut mit dem Wert “Mister X” haben
- Substring "*"
  - Voraussetzung: Attributdefinition hat eine Substring Matching Rule
  - " (cn=Mister*)"” alle Einträge, die ein CN-Attribut haben, dessen Wert mit “Mister” anfängt
- Approximate
  - Achtung: Implementierungsabhängig
  - " (cn~=Mister)" alle Einträge, die ein CN-Attribut haben, dessen Wert ähnlich wie “Mister” klingt (nur fürs Englische)

Search-Filter-Operatoren 2/2

- Greater than or equal to und less than or equal to
  - Voraussetzung: Attributdefinition hat eine Ordering Matching Rule
  - "(sn<=Smith)" alle Einträge, die ein SN-Attribut haben, dessen Wert alphabetisch vor oder als "Smith" einsortiert wird (z.B. von sn=Adam bis sn=smith)
  - "(age>21)" ist falsch es funktioniert nur: " (!(age<=21))"
- Presence
  - "(telephoneNumber=*)" alle Einträge die eine Telefonnummer enthalten
  - "(objectclass=*)" *alle* Einträge, da jeder Eintrag mindestens ein objectclass-Attribut enthalten muss.

DAASI International

Search Filter Extensions

- LDAPv3 definiert einen *extensible matching filter*
  - syntax: attr [“:dn”] [“:” matchingrule] “:=“ value
    - attr ist ein Attributname
    - “:dn” bedeutet, dass auch DN-Attribute mit durchsucht werden sollen
    - Die Matching Rule kann durch eine OID oder durch einen Namen angegeben werden
  - Beispiele:
    - (cn:1.2.3.4.5.6:=Mister X) verwende Matching Rule 1.2.3.4.5.6 für den Vergleich
    - (o:dn:=company) suche nach o=company in den Attributes und im DN

DAASI International

Search-Filter-Kombinationen

- Filter können logisch verknüpft werden
  - AND-Operator: &
    - " (& (cn=Mister X) (mail=*dot.com))" alle Einträge, deren cn-Attribut einen Wert "Mister X" enthält und deren mail-Attribut einen Wert enthält, welcher mit "dot.com" endet.
  - OR-Operator: |
    - " (| (cn=Mister X) (sn=Xerxes))" alle Einträge, die entweder ein cn-Wert "Mister X" oder einen sn-Wert "Xerxes" haben.

DAASI International

Search-Filter-Metazeichen

- Fünf Zeichen haben in LDAP-Search-Filtern eine besondere Bedeutung und müssen deshalb als Hexadezimal-Escape-Sequence geschrieben werden, wenn danach gesucht werden soll:
  - ´*´ (dec. 42, hex 0x2A) als : ´\2a´
  - ´(´ (dec. 40, hex 0x28) als: ´\28´
  - ´)´ (dec. 41, hex 0x29) als: ´\29´
  - ´\´ (dec. 92, hex 0x5C) als: ´\5c´
  - NUL (dec. 0, hex 0x00) als: ´\00´
- Beispiel
  - Der Wert “A*Star” muss in Filtern folgendermaßen geschrieben werden:
    "(cn=A\2AStar)"

LDAP URL (RFC 4516)

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

Client-Server-Kommunikaton in LDAP

- LDAP ist ein Request-Response Protokoll mit ASN.1 (DER/BER) Encoding
- Jede LDAP-Operation (bind, search, add, modify, modifyDN, delete) unterteilt sich in einen Client-Request und einen Server-Response
- LDAP Operationen können durch einen Extension-Mechanismus erweitert werden
- Neue Operationen können ebenfalls spezifiziert werden
- Viele solche Erweiterungen wurden standardisiert und werden zunehmend in den verschiedenen Produkten implementiert
- Server Response enthält Daten, Verweise, und/oder Error-Codes

DAASI International

LDAPv3 Erweiterungsmechanismen 1/2

- LDAP controls
  - RFC 2251, Par. 4.1.12
  - alle 9 LDAP-Operationen (bind, search, add, ...) können mit Controls erweitert werden
  - Controls modifizieren das Verhalten einer ldap-Operation
  - besteht aus controlType, criticality, [controlValue]
  - Client und Server müssen das Control unterstützen
  - es wurden bereits mehrere Controls standardisiert

DAASI International

LDAPv3 Erweiterungsmechanismen 2/2

- LDAP extended operations
  - RFC 2251, Par. 4.12
  - neu definierte LDAP-Operation (zusätzlich zu den 9)
  - ExtendedRequest: requestName, [requestValue]
  - ExtendedResponse: LDAPresult,[responseName, response]
- SASL Mechanismen
  - durch die Unterstützung von der generischen SASL-Authentifizierung kann LDAP durch weitere neue SASL-Mechanismen erweitert werden

Der Root_DSE-Eintrag

- Der Root_DSE-Eintrag ist ein spezieller Eintrag im LDAP-Server direkt unterhalb vom DN ""
- Er enthält operationale Attribute, die den Server beschreiben:
  - namingContext (Namensraum, multivalued)
  - subschemaSubentry (DN des Eintrags der das vom Server unterstützte Schema spezifiziert)
  - AltServer (Host:port eines weiteren LDAP-Servers, der die gleichen Daten enthalten sollte)
  - supportedLDAPVersion (Unterstützte LDAP-Version)
- Er enthält Attribute, die die unterstützten Erweiterungen spezifizieren:
  - supportedExtensions
  - supportedControls
  - supportedSASLMechanisms
- Mit dem OpenLDAP-Command-line-Tool ldapsearch kann man den Root_DSE-Eintrag folgendermaßen auslesen:
  - `ldapsearch -x -b "" -s base +`

subSchema

- Unterhalb des im RootDSE-Attribut subschemaSubentry referenzierten DN veröffentlicht der Server, welche Schemaelemente er kennt:

```
# base <cn=Subschema> with scope base filter: (objectclass=*)
dn: cn=Subschema
structuralObjectClass: subentry
createTimestamp: 20050723111603Z
modifyTimestamp: 20050723111603Z
ldapSyntaxes: ( 1.3.6.1.4.1.1466.115.121.1.50
 DESC 'Telephone Number' ) ...
matchingRules: ( 1.3.6.1.4.1.4203.1.2.1 NAME
 'caseExactIA5SubstringsMatch' SYNTAX
 1.3.6.1.4.1.1466.115.121.1.26 ) ...
attributeTypes: ( 2.16.840.1.113730.3.1.241 NAME
 'displayName' DESC 'RFC2798: preferred name to be
 used when displaying entries' EQUALITY
 caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE ) ...
objectClasses: ( 1.3.6.1.4.1.1466.344 NAME 'dcObject' DESC
 'RFC2247: domain component object' SUP top
 AUXILIARY MUST dc ) ...
```

StartTLS Extended Operation

- Mit der StartTLS Extended Operation kann der Client die Client-Server-Verbindung nachträglich verschlüsseln
- Sie ist Bestandteil des LDAP-Standards und muss von allen Clients und Servern implementiert sein

LDAPv3 Standard

- Fertige Standards:
  - Das Informationsmodell
  - Zwei Namensräume
  - Ein Netzwerkprotokoll (Client-Server)
  - Sichere Authentifizierungs- und Verschlüsselungsmechanismenn
  - Ein Referierungsmodell (Referral)
  - Erweiterungsmechanismen
  - LDAP URL
  - Datenaustauschformat (LDIF)
  - APIs für C und Java (de facto)
- Nicht standardisiert
  - Replikationsmodell
  - Zugriffskontrolle

DAASI International

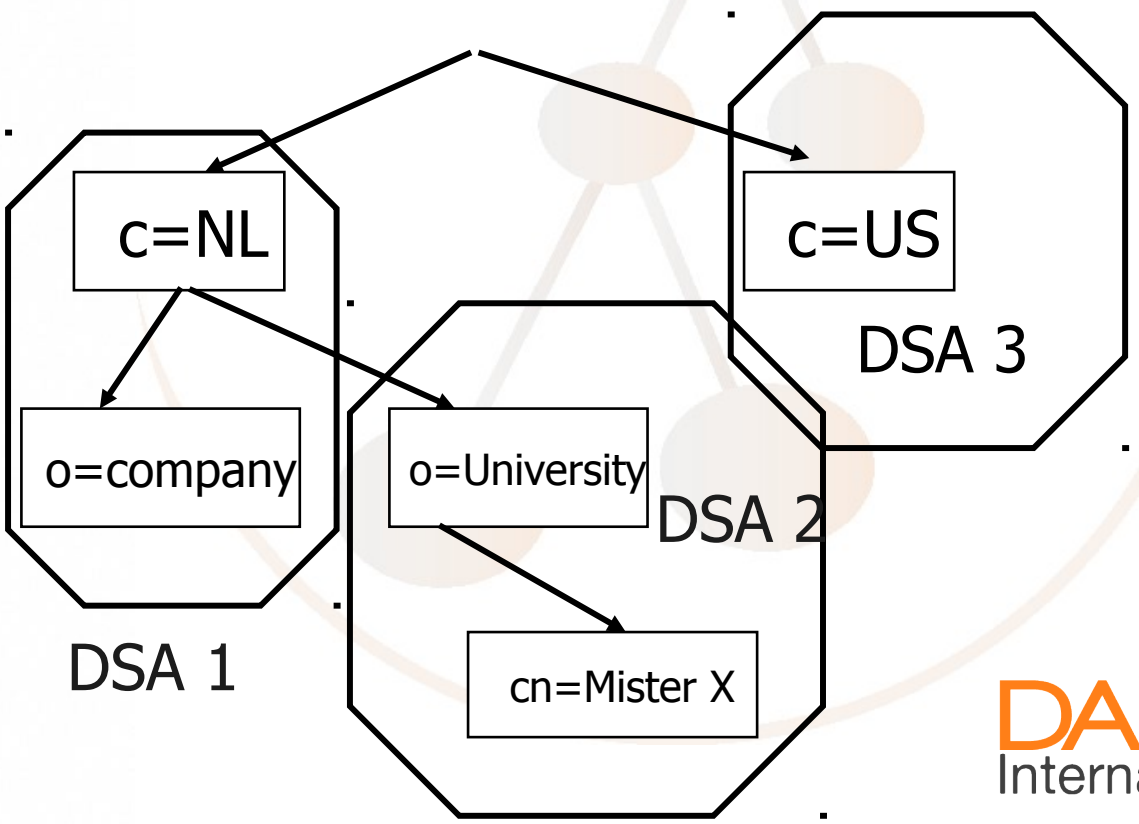
Endgültige Version des LDAPv3-Standards

- LDAP: The Protocol [RFC4511]
- LDAP: Directory Information Models [RFC4512]
- LDAP: Authentication Methods and Security Mechanisms [RFC4513]
- LDAP: String Representation of Distinguished Names [RFC4514]
- LDAP: String Representation of Search Filters [RFC4515]
- LDAP: Uniform Resource Locator [RFC4516]
- LDAP: Syntaxes and Matching Rules [RFC4517]
- LDAP: Internationalized String Preparation [RFC4518]
- LDAP: Schema for User Applications [RFC4519]

DAASI International

Verteilung der Daten

- Daten können auf verschiedene Server, sog. *Directory Service Agents (DSA)* verteilt werden
  - In LDAP mittels Referrals:



Referrals

- Mit Hilfe von Referrals kann von einem Server auf einen anderen Server verwiesen werden
- Ein Referral besteht aus einer LDAP-URL
- Referrals sind im Client-Server-Protokoll integriert.
  - Erhält der Client neben bzw. anstatt Daten ein Referral, kann er sich an den entsprechenden Server wenden, um von dort die gesuchten Daten zu erfragen
- Beispiel:

```
dn: o=University,c=nl
objectclass: referral
objectclass : extensibleObject
ref: ldap://dsa2.org:3891/o=University,c=nl
```

Replikation

- Standardisierungsbemühungen seit 1998
- IETF WG LDUP
  - LDAP Duplication / Replication / Update Protocols
- Ohne Standard keine Implementierungs-übergreifende Replikation möglich
- Augenblickliche Lösungen:
  - Datenaustausch via LDIF-Dateien
  - Defacto Standard der Open Source Lösung (slurpd, syncrepl, s.u.)
  - XML-Ansätze
  - Client Update Mechanismen
  - Proprietäre Lösungen

DAASI International

LDAP-Authentifizierungsmethoden

LDAP-Authentifizierung

- Simple Bind
  - Man authentifiziert sich über einen Eintrag mittels DN und Passwort
  - Passwort geht ungeschützt über das Netz!
- Simple Bind + TLS (Transport Layer Security ~= SSL)
  - Vor dem Bind-Vorgang wird die gesamte Session verschlüsselt
  - StartTLS-Operation
- Alternative Authentifizierung mittels SASL
  - Simple Authentication and Security Layer
  - Vorgeschrieben: Digest MD5 (challenge response)
  - Andere SASL-Mechanismen möglich

SASL

- 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)
    - Kerberos IV
    - GSSAPI (and thus Kerberos V)
    - EXTERNAL (e.g. X.509 certificate used in the underlying SSL / TLS)

DAASI International

GSSAPI

- GSSAPI (Generic Security Service Application Program Interface)
  - Security framework that abstracts from underlying protocols
  - Includes a Kerberos mechanism

NSS und PAM

- Name Service Switch (NSS)
  - Layer in Unix C bibliotheken, die verschiedene Möglichkeiten bietet user, groups, IP services, networks, etc., aufzulisten oder zu suchen:
    - Flat files (etc/passwd, etc.) = schwer zu administrieren
    - NIS (Network Information Service) = hat Sicherheitslöcher
    - LDAP = ☺
- Pluggable Authentication Modules (PAM)
  - Framework für Login-Dienste
  - Verwaltet authentication, accounts, sessions and passwords
  - Module existieren für LDAP, Kerberos, etc.

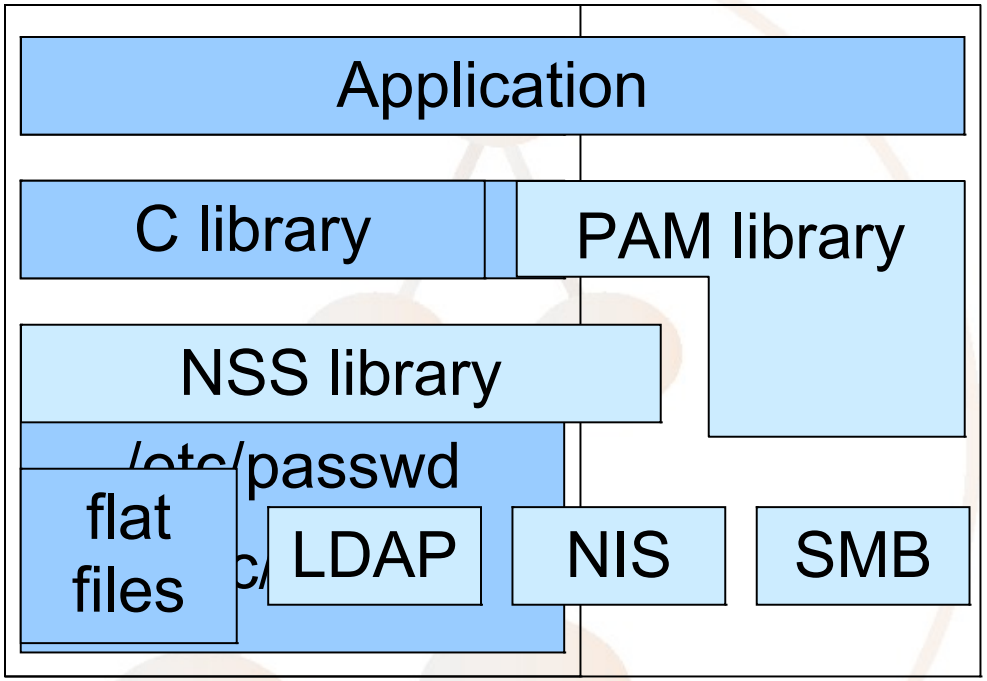
DAASI International

LDAP für NIS

- RFC 2307: An Approach for Using LDAP as a Network Information Service, L. Howard, March 1998
  - Definiert Mechanismen um Dinge, die zu TCP/IP oder zu UNIX gehören in LDAP abzubilden
  - Ermöglicht die Anwendung von LDAP als zentraler Namensdienst
  - Software unter: http://www.padl.com/nss_ldap.html

DAASI International

Unix Authentifizierung



Unix-Benutzerverwaltung

- Standardisierte LDAP Objektklassen zur Abbildung von NIS (RFC 2307)
  - 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

Authentifizierungsdienst (1/4)

- Problem:
  - Benutzer haben Zugriff auf viele Rechner
  - Auf jedem Rechner eigene LoginID und Passwort
  - Benutzer muss sich viele Passwörter merken
  - Unterschiedliche Password-Policies
  - ➔ sehr hoher Administrationsaufwand

Zentraler verzeichnisdienstbasierter Authentifizierungsdienst

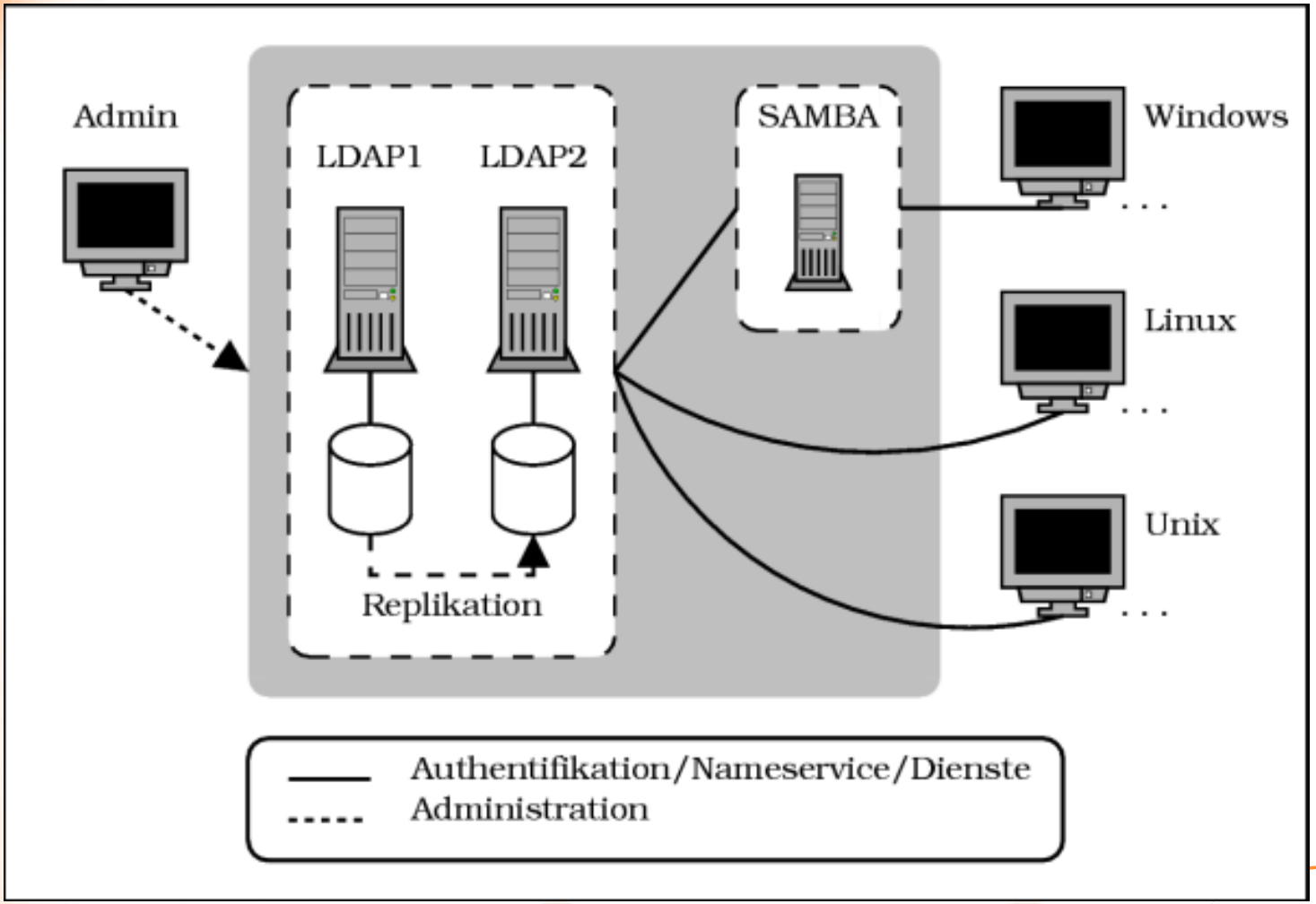
- Unix-Clients
  - Können mittels NSS / PAM-LDAP direkt auf LDAP-Server zugreifen
  - Kann gecashed werden: nscd (Name Service Caching Daemon)
  - Aber auch Anbindung an MS Active Directory (AD) möglich mit Kerberos
- Windows-Clients
  - Einfache Integration in AD
  - Aber auch über SAMBA Anbindung an LDAP-Server möglich
    - NT4 Domäne (Samba 2.x)
    - AD-Simulation (Samba 3.0)

DAASI International

OpenLDAP/Samba Kochrezept

- Man nehme eine Linux-Box mit Minimalinstallation
- Füge Folgendes hinzu
  - binutils
  - gcc
  - glibc-devel
  - make
  - nss_ldap
  - openldap2
  - openldap2-client
  - openldap2-devel
  - openssl-devel
  - Pam und pam-devel
- Nicht zu vergessen Samba 3
- Sehr nützlich sind die IDEALX smbldap-tools

Architektur der OpenLDAP-Lösung

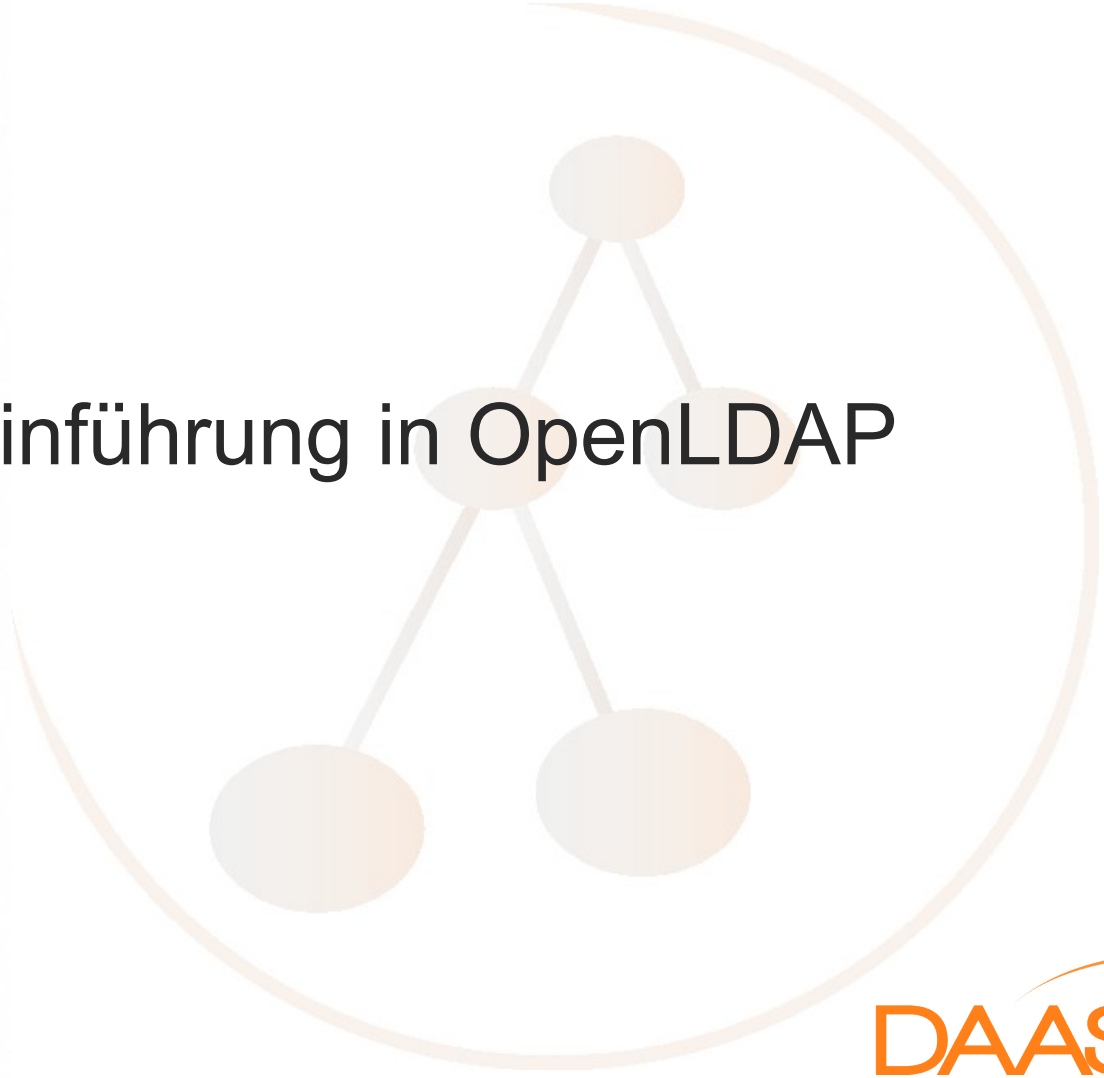


Technologie

- Referenz-Implementierung des RFC [TODO]
- Impelentiert in C
- Aktuelle Version 2.4.40
- Speichert Daten in verschiedenen Backends
  - BDB / HDB (Berkeley)
  - MDB (Memory Mapped)
  - Text-Dateien (LDIF)
- Konfiguration
  - Datei
  - LDIF (cn=config)

DAASI International

Einführung in OpenLDAP



DAASI International

Geschichte des Projekts 1/2

- University of Michigan (Tim Howes & Co.) erstellte die erste Open-Source-LDAP-Implementierung
  - LDAP (1992)
  - U-Mich LDAP 3.3 (April 1996)
  - Grundlage für eine ganze Reihe kommerzieller Implementierungen
- Critical Angle *LDAPworld* patches
- *Boolean LDAP* (July 1998)
  - U-Mich 3.3 + critical angle patches + misc. patches + bug fixes
- Net Boolean Incorporated sponsors the OpenLDAP Foundation and Project in August 1998
  - “to provide open source LDAP software and information”
- OpenLDAP 1.0 released August 1998
  - *Artistic License*

DAASI International

Geschichte des Projekts 2/2

- OpenLDAP 1.1 (Dec 1998)
- OpenLDAP 1.2 (Feb 1999)
- OpenLDAP 2.0 alpha (July 1999)
- ISC Sponsors OpenLDAP (1999)
- Kurt attends first IETF (Nov 1999)
- SuSE hires Kurt (Jan 2000)
- Howard Chu joins the “core”
- OpenLDAP 2.0 released (Aug 2000) (erste LDAPv3-Implementierung)
- LDAPBIS chartered / LDAPEXT shuts down
- SuSE lays Kurt off (Mar 2001)
- OpenLDAP 2.1 alpha (Feb 2001)
- IBM/LTC hires Kurt (May 2001)
- OpenLDAP 2.1 (July 2001)
- OpenLDAP 2.2 (December 2003)
- OpenLDAP 2.3 (June 2005) current: 2.3.41
- OpenLDAP 2.4 (October 2007) current 2.4.40

DAASI International

Vorteile

- Sehr gute Performanz auch bei großen Datenmengen (Mehrere millionen Einträge)
- Einfaches Erstellen und Einbinden eigener Schema
- Vielzahl fertiger Overlays (Erweiterungen)
  - Referential integrety
  - MemberOf
  - AccessLog
- Eigene Erweiterungen (in C) möglich
- Protokollierung von Änderungen über Overlay AccessLog im Server
- Feingranulare ACLs

Technologie

- Referenz-Implementierung der RFCs (4510-4519)
- Implementiert in C
- Speichert Daten in verschiedenen Backends
  - BDB / HDB (Berkeley)
  - MDB (Memory Mapped)
  - Text-Dateien (LDIF)
- Flexible Overlay-Technologie

DAASI International

Aufbau eines Attributs im Schema

```
attributetype ( 2.16.840.1.113730.3.1.241
        NAME 'displayName'
        DESC '[...]'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )
```

Umfang der OpenLDAP-Distribution

- OpenLDAP Release enthält:
  - C-Bibliotheken für LDAP
  - Slapd LDAP-Server
    - mit eine ganzen Reihe von Daten-Backend-Modulen, die auch parallel eingesetzt werden können.
    - mit sog. Overlays kann das Verhalten des Servers ebenfalls modifiziert werden
  - Client tools
    - Ldapsearch
    - Ldapmodify
    - Ldapdelete
    - Ldapmodrdn
    - ldappasswd

DAASI International

OpenLDAP Installieren

- Alle Linux-Distributionen enthalten einen (manchmal etwas veralteten) Openldap2-Server
- Der kann mit den entsprechenden Bordmitteln installiert/aktiviert werden
  - RPM, emerge, etc.

OpenLDAP per Hand installieren

- Tgz-File von http://www.openldap.org/software/download/ downloaden
  - Aktuelle Stable Release ist 2.4.21
- tar -xzf openldap-stable-20050429.tgz
- cd openldap-2.4.21
- ./configure
- make depend
- make
- make test
- su
- make install

Vorausgesetzte Software

- BerkeleyDB als Database Management System
- Cyrus-SASL als Authentifizierungs-System
- OpenSSL für sicheren und verschlüsselten Datentransport.
- Gegebenenfalls Kerberos zur Anwender-Authentifzierung.

DAASI International

Configure-Optionen 1: Pfade

- ./configure --help:

```
--prefix=PREFIX         install architecture-independent files in PREFIX
                        [/usr/local]
--exec-prefix=EPREFIX   install architecture-dependent files in EPREFIX
                        [same as prefix]
--bindir=DIR            user executables in DIR [EPREFIX/bin]
--sbindir=DIR           system admin executables in DIR [EPREFIX/sbin]
--libexecdir=DIR        program executables in DIR [EPREFIX/libexec]
--sysconfdir=DIR        read-only single-machine data in DIR [PREFIX/etc]
--localstatedir=DIR     modifiable single-machine data in DIR [PREFIX/var]
--libdir=DIR            object code libraries in DIR [EPREFIX/lib]
--includedir=DIR        C header files in DIR [PREFIX/include]
```

Configure-Optionen 2: allgemeine Features

```
--enable-debug          enable debugging [yes]
--enable-dynamic        enable linking built binaries with dynamic libs [no]
--enable-syslog         enable syslog support [auto]
--enable-proctitle      enable proctitle support [yes]
--enable-ipv6           enable IPv6 support [auto]
--enable-local          enable AF_LOCAL (AF_UNIX) socket support [auto]
--with-cyrus-sasl       with Cyrus SASL support [auto]
--with-fetch            with fetch(3) URL support [auto]
--with-threads          with threads [auto]
--with-tls              with TLS/SSL support [auto]


--enable-slapd          enable building slapd [yes]
--enable-slurpd         enable building slurpd [auto]
```

Configure-Optionen 3: slapd Features

```
--enable-aci            enable per-object ACIs (experimental) [no]
                        [Access Control Instructions im DIT]
--enable-cleartext      enable cleartext passwords [yes]
--enable-crypt          enable crypt(3) passwords [no]
--enable-lmpasswd       enable LAN Manager passwords [no]
--enable-spasswd        enable (Cyrus) SASL password verification [no]
                        [um Passworte für Cyrus-SASL im DIT abzulegen,
                        sasldb2 wird dann nicht mehr benötigt]
--enable-modules        enable dynamic module support [no]
                        [Backends und Overlays die als Modul bei Bedarf
                        nachgeladen werden können]
--enable-rlookups       enable reverse lookups of client hostnames [no]
--enable-slapi          enable SLAPI support (experimental) [no]
--enable-slp            enable SLPv2 support [no]
--enable-wrappers       enable tcp wrapper support [no]
                        netzwerkseitige Zugangskontrollen über
                        /etc/hosts.allow, hosts.deny
```

Configure-Optionen 4: slapd backends

```
--enable-bdb              enable Berkeley DB backend no|yes|mod [yes]
--enable-dnssrv           enable dnssrv backend no|yes|mod [no]
--enable-hdb              enable Hierarchical DB backend no|yes|mod [no]
--enable-ldap             enable ldap backend no|yes|mod [no]
--enable-ldbm             enable ldbm backend no|yes|mod [no]
  --with-ldbm-api         with LDBM API auto|berkeley|bcompat|mdbm|gdbm [auto]
  --with-ldbm-type        use LDBM type auto|btree|hash [auto]
--enable-meta             enable metadirectory backend no|yes|mod [no]
--enable-monitor          enable monitor backend no|yes|mod [yes]
--enable-null             enable null backend no|yes|mod [no]
--enable-passwd           enable passwd backend no|yes|mod [no]
--enable-perl             enable perl backend no|yes|mod [no]
--enable-shell            enable shell backend no|yes|mod [no]
--enable-sql              enable sql backend no|yes|mod [no]

--enable-dyngroup         Dynamic Group overlay no|yes|mod [no]
--enable-rewriteenable    DN rewriting in back-ldap and the rwm overlay [auto]
                          [unterstützt regelbasiertes Umschreiben von Daten]
--enable-proxycache       Proxy Cache overlay no|yes|mod [no]
                          Overlay um Datensätze eines definierten Suchmusters im
                          lokalen Cache für eine definierte Zeit zu speichern.
                          Wird in Verbindung mit LDAP und META eingesetzt.
```

Beispiel für configure-Aufruf

```
./configure \
--enable-local=yes \
--enable-aci=yes \
--enable-spasswd=yes \
--enable-modules=yes \
--enable-rewrite=yes \
--enable-bdb=yes \
--enable-hdb=yes \
--enable-ldap=mod \
--enable-meta=mod \
--enable-monitor=mod \
--enable-relay=mod \
--enable-dyngroup=mod \
--enable-proxycache=mod \
--with-tls
```

DAASI International

Probleme bei configure

- Werden die Bibliotheken für BDB, OpenSSL und SASL nicht automatisch gefunden, können entsprechende Pfate gesetzt werden:
  - ```
    /usr/bin/env LD_LIBRARY_PATH="$
    {LD_LIBRARY_PATH}:/usr/local/BerkeleyDB.4.2/lib" CPPFLAGS="-
    I/usr/local/BerkeleyDB.4.2/include" \ LDFLAGS="-
    L/usr/local/BerkeleyDB.4.2/lib" \
    configure ....
    ```
    ```

Openldap konfigurieren

- Traditionell wird Openldap in der Konfigurationsdatei /etc/openldap/slapd.conf konfiguriert
  - Diese Datei besteht aus einem ersten globalen Teil und einem oder mehreren Datenbank-Teilen (database)
  - Einträge in Datenbank-Teilen können globale Einträge überschreiben
- Mittlerweile gibt es auch die Möglichkeit, die Konfiguration direkt in den LDAP-Daten abzulegen, wodurch Neustart des Daemons bei Konfigurationsänderungen entfällt

Konfiguration

- Zwei Möglichkeiten: Datei und „cn=config“
- Globale Parameter:
  - Einbinden von Schema
  - Laden von Modulen (z.B. Overlays)
  - Konfiguration der Verbindungssicherheit
  - Globale ACLs
- Parameter pro Backend:
  - Manager-Account
  - Definieren der Attribute im Index
  - Konfiguration der geladenen Overlays
  - Replikation
  - Backend-ACLs

DAASI International

Konfiguration über Datei

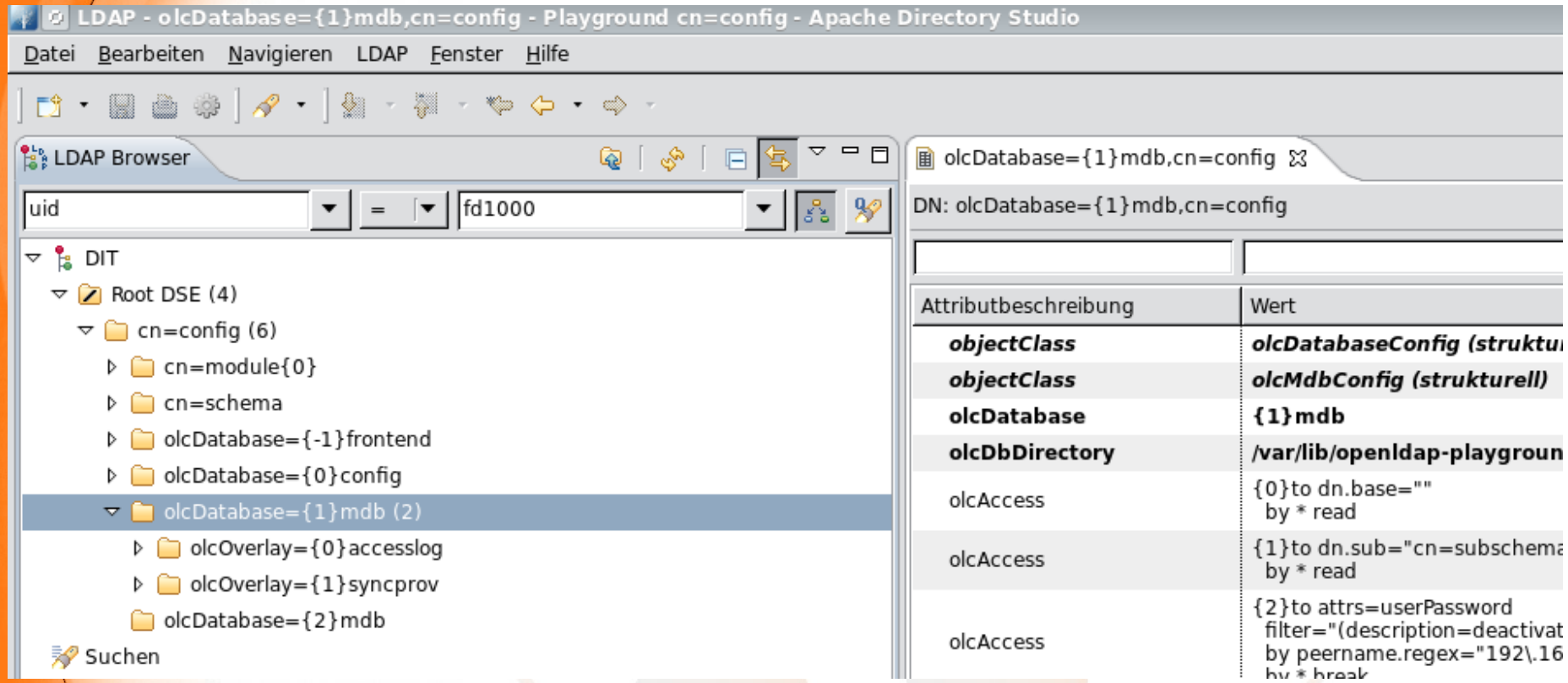
- Kontra
  - Statische Konfiguration → Erfordert Neustart bei Änderungen
  - Kann nicht auf andere Server repliziert werden
- Pro
  - Erlaubt Kommentare in der Konfiguration
  - Einfach mit einem Texteditor Änderbar
  - Im Besonderen sind Schema-Änderungen leicht möglich

DAASI International

Konfiguration über cn=config

- Kontra
  - Keine Kommentare in der Konfiguration möglich
  - Umständliche Konfiguration über LDAP-Befehle
  - Unübersichtliche Darstellung
- Pro
  - Dynamische Konfiguration → Viele Einstellungen werden ohne Neustart übernommen (z.B. ACLs)
  - Replikation z.B. in Multi-Provider-Umgebung möglich

Konfiguration über cn=config



Eine minimale Slapd.conf

```
## Globaler Teil
# Schemadateien laden:
include         /etc/openldap/schema/core.schema

# Prozessinformationsdateien
pidfile         /var/run/slapd/slapd.pid        # Prozessnummer
argsfile        /var/run/slapd/slapd.args       # Argumente beim
                                                # Start von slapd
## erste Datenbankdefinition
database        bdb
suffix          "dc=daasi,dc=de"                # Namensraum
rootdn          "cn=manager, dc=daasi,dc=de" # Superusereintrag
rootpw          {SSHA}3c6fGKuzajVjVqgr          # Passwort

directory       /var/lib/ldap                   # Datenverzeichnis
```

Erstellen des Passwort-Hash

- Auch wenn die Datei slapd.conf nur für Root lesbar sein sollte, ist es sinnvoll, das Root-Passwort als Hash abzulegen
- Hierzu wird bei OpenLDAP das Hilfsprogramm slappasswd mitgeliefert:
  - slappasswd -s secret
  - Es werden neben dem voreingestellten SSHA-Algorithmus noch folgende unterstützt:
    - {SSHA}, {SHA}, {Crypt}, {MD5}, {SMD5}
    - slappasswd -s secret -h {crypt}

DAASI International

Konfigurationsparameter loglevel

- Mit loglevel wird die Ausführlichkeit der Logging-Information spezifiziert, wobei verschiedene Level kombinierbar sind
  - -1 enable all debugging
  - 0 no debugging
  - 1 trace function calls
  - 2 debug packet handling
  - 4 heavy trace debugging
  - 8 connection management
  - 16 print out packets sent and received
  - 32 search filter processing
  - 64 configuration file processing
  - 128 access control list processing
  - 256 stats log connections/operations/results
  - 512 stats log entries sent
  - 1024 print communication with shell backends
  - 2048 print entry parsing debugging

Loglevel 192 # Konfig. u. ACL

Slapd starten

- Linux-Distributionen stellen ein rc-script zum Starten und Stoppen des Slapd zur Verfügung
  - Unter Suse: rcldap
  - Dieses wird über /etc/sysconfig/openldap konfiguriert
- Manchmal macht es Sinn, den Slapd auch per Hand zu starten:
  - /usr/lib/openldap/slapd -d 64
  - Der Prozess wird im Vordergrund gestartet und Debugging-Information auf die Konsole ausgegeben.
  - Kann mit Strg-C abgebrochen werden

DAASI International

Daten eingeben

- Entweder durch ldapadd mit LDIF-Datei
- Oder slapcat aus einem bestehenden Serverein spezielles LDIF-File erstelen, das man mit slapadd in den Server eingeben kann
  - Slapcat und slapadd können bei bestimmten Datenbackends nur bei gestopptem Server verwendet werden

Weitere globale Konfigurationsparameter 1/3

- Mit dem Parameter referral kann man einen Ldap-Server spezifizieren, auf den alle Anfragen, die nicht den Namensraum des Servers betreffen verwiesen werden.

```
referral     ldap://root.openldap.org/
```

- Mit dem Parameter defaultsearchbase kann man einen voreingestellten Namensraum spezifizieren, der abgesucht wird, wenn der Client keinen Search-BaseDN angibt

```
defaultsearchbase dc=daasi,dc=de
```

- Mit dem Parameter threads kann die maximale Anzahl offener Threads spezifiziert werden. Default ist 16

```
threads 32
```

- Mit dem Parameter ucdata-path kann man spezifizieren, wo der Server Unicode-Zeichen-Tabellen sucht.

```
ucdata-path /usr/etc/ucdata
```

Weitere globale Konfigurationsparameter 2/3

- Mit dem Parameter gentlehup kann man das Verhalten des Servers bei Empfang eines SIGHUP-Signals modifizieren. Wird der Parameter auf on gestellt, wird der Server erst terminieren, wenn alle Clients die Session beendet haben. Schreiboprationen werden jedoch in der Zwischenzeit abgelehnt (Error "Unwilling to perform")

```
gentlehup on
```

- Mit dem Parameter idletimeout kann man spezifizieren, wie lange (Anzahl Sekunden) der Server eine aktivitätslose Verbindung zum Client aufrecht erhält. (0= nie abbrechen. Default ist 0)

```
idletimeout 360
```

- Mit dem Parameter modulepath kann eine Liste von Verzeichnissen angegeben werden, in denen der Server nach dynamischen Modulen sucht
- Mit dem Parameter moduleload werden zu ladende dynamische Module angegeben (Voraussetzung --enable-modules in configure)

```
modulepath /usr/lib/openldap:/usr/local/lib

moduleload  back_ldap.la
```

Weitere globale Konfigurationsparameter 3/3

- Mit dem Parameter rootDSE kann eine LDIF-Datei angegeben werden, in der zusätzliche rootDSE-Attribute spezifiziert werden können, die zusätzlich beim Auslesen des rootDSE-Eintrags zurückgegeben werden

```
rootDSE /usr/etc/rootdse.ldif
```

- Mit dem Parameter schemadn kann der DN, unter dem der Server das von ihm gekannte Schema veröffentlicht, spezifiziert werden. Default ist "cn=subschema"

```
schemadn "cn=subschema,dc=daasi,dc=de"
```

Konfigurationsparameter zu Passwort-Hashes

- Mit dem Parameter password-hash kann man einen oder mehrere Hash-Algorithmen angeben, die bei der Speicherung von Passwörtern im Attribut userPassword verwendet werden sollen. Dies wird allerdings nur bei der Verwendung der extended Operation Password Modify berücksichtigt, nicht jedoch bei normalren Add- oder Modify-Operationen. Möglich sind: {SSHA}, {SHA}, {Crypt}, {MD5}, {SMD5}. Default ist {SSHA}

```
password-hash {CRYPT}
```

- Mit dem Parameter password-crypt-salt-format kann das Format des Salt-textes bei der Generierung von crypt-Hashes spezifiziert werden. Dieses wird im sprintf-Format angegeben und darf nur eine Konversion enthalten. Beispiele: "%.2s" erzeugt ein Salt aus zwei Bytes. Default ist "%s", welches ein Salt aus 31 Byte erzeugt.

```
password-crypt-salt-format "%.8s"
```

Konfigurationsparameter conn_max_pending

- Mit conn_max_pending kann man die maximale Anzahl von in einer Warteliste eingetragenen requests für nicht authentifizierte Client-Server-Verbindungen (sessions)
  - Wenn die Anzahl erreicht wird, wird die Session vom Server beendet
  - Defaultwert ist 100
- Mit conn_max_pending_auth kann dasselbe für authentifizierte Sessions spezifiziert werden
  - Default ist 1000

DAASI International

Konfigurationsparameter disallow

- Mit disallow können verschiedene Features abgestellt werden, z.B.:
  - bind_v2 (LDAPv2-Authentifizierung)
  - bind_anon (Anonyme Client)
  - bind_simple (Klartext-Passwort-Authentifizierung)
  - bind_krbv4 (Kerberos IV-Authentifizierung)

```
disallow bind_v2 bind_anon
```

Konfigurationsparameter require

- Mit require kann spezifiziert werden, welche Features ein Client beherrschen muss, um mit dem Server zu kommunizieren
- Es kann global, oder per Datenbank angegeben werden
- Folgende Features können genannt werden:
  - bind (Client muss sich authentifizieren)
  - LDAPv3 (Client muss LDAPv3 unterstützen)
  - SASL (Client muss sich via SASL authentifizieren)
  - strong (Client muss stake Authentifizierung verwenden (=SASL)
  - none (Keine Vorbedingungen)

```
require LDAPv3 bind
```

Size und time limits 1/3

- Mit dem Parameter sizelimit kann global eine Maximalanzahl von Einträgen spezifiziert werden, die der Server bei Anfragen zurückgibt. Default ist 500. -1 oder "unlimited" schaltet diese Limitierung aus.

```
sizelimit 30
```

- Mit dem Parameter timelimit kann global eine Maximaldauer (in Sekunden) angegeben werden, die der Server für die Beantwortung einer Anfrage hat. Bei dessen Überschreitung, gibt der Server die Daten zurück, die er bis dahin gefunden hat. Default ist 3600. -1 oder "unlimited" schaltet diese Limitierung aus

```
timelimit 1200
```

DAASI International

Size und time limits 2/3

- Neben der einfachen Spezifikation der globalen Limits, kann man dies auch je Datenbank abhängig von Authentifizierung definieren, also unterscheiden zwischen:
  - anonymous (nicht authentifizierte Benutzer)
  - users (authentifizierte Benutzer)
  - ein oder mehrere bestimmte authentifizierte Benutzer
    - Genauer dn (exact bzw. base)
    - Gruppe von dns (onelevel, subtree, children,regex)
    - Gruppenmitglieder (group[/oc[/at]), wobei Gruppe durch dn-Werte des spezifizierten Attributs (default ist member) in Gruppenobjeckten (defaultklasse ist groupOfNames) definiert werden

```
dn.exact=cn=administrator1,ou=admins,dc=daasi,dc=de
dn.subtree=ou=admins,dc=daasi,dc=de
group/myGroupclass/myMember=cn=admingroup,ou=groups,...
```

Size und time limits 3/3

- Das Limit kann zusätzlich unterschieden werden in soft und hard limits, wobei soft limit zur Ausführung kommt, wenn der client kein eigenes limit spezifiziert hat und hard, um zu vergleichen ob der vom Client gesetzte limit zu hoch ist
  - -1 oder none schaltet solche limits aus
  - Beispiele:

```
limits anonymous size.soft=3 size.hard=soft
limits users size.soft=50 size.hard=soft
limits dn.subtree=ou=admins,dc=daasi,dc=de size.soft=-1
```

Database-Konfigurationsparameter

```
## Database-Teil


# Mit dem index-Parameter können interne Indices aufgebaut werden,
# die Suchvorgänge erheblich beschleunigen
# Je Matchingrule-Art werden getrennte Indices aufgebaut
# pres (reines Vorhandensein), eq (equality matching rule)
# sub (substring matching rule), approx (approximate matching)

index    objectclass  pres,eq
Index    cn,sn,uid    pres,eq,sub,approx
```

DAASI International

Backend-Konfiguration: ldbm

- Ldbm ist das klassische Datenbackend von OpenLDAP
  - Stabil
  - Performant
  - Keine Transaktionen
  - Heutzutage ist BDB das Datenbackend der Wahl
- Folgende ldbm-spezifische Konfigurationsparameter gibt es:

```
cachesize       # Anzahl der im Speicher vorgehaltenen Einträge
dbcachesize     # Anzahl von Byte, die im Speicher für jeden
                # Index zur Verfügung gestellt werden soll
dbnolocking     # Hiermit kann man Filelocking abstellen
                # performanter aber weniger stabil
dbnosynch       # Schreibzugriffe werden nicht sofort auf Platte
                # geschrieben. performanter aber weniger stabil
mode            # Zugriffsrechte der Datendateien
Index           # interne Indices
```

Backend-Konfiguration: bdb

- bdb (Sleepycat Berkeley DB) ist ab OpenLDAP 2.1.4 das voreingestellte Datenbackend, welches folgende Vorteile gegenüber ldbm hat:
  - Transaktionslogging
  - Pagelevel locking, wodurch slapcat und slapadd auch bei laufendem Server verwendet werden dürfen
  - Mehrere Threads können gleichzeitig auf die gleichen Datenbankdateien zugreifen
  - Performantere Indexerstellung und Indexsuche

```
cachesize       # Anzahl der im Speicher vorgehaltenen Einträge
dbnosynch       # Schreibzugriffe werden nicht sofort auf Platte
                # geschrieben. performanter aber weniger stabil
mode            # Zugriffsrechte der Datendateien
index           # interne Indices
checkpoint      # setzt fest wie oft die Daten auf Platte gesyncht
                # werden. Angabe in Kilobyte und Minuten
lockdetect      # Verhalten bei zwei im Konflikt
                # stehenden Transaktionen
```

Zugriffskontrolle (ACLs)

- Eine ACL besteht aus vier Teilen
  - <what> Für welchen Bereich ist die ACL gültig
  - <who> Für wen ist die ACL gültig
  - <access> Welche Rechte hat <who>
  - <control> Flags zur Kontrolle bei der Verarbeitung der nachfolgenden ACLs
- Eine ACL ist folgendermaßen aufgebaut:

  access to <what>

  [by <who> [<access>] [<control>]]+

Diese Berechtigungen können vergeben werden

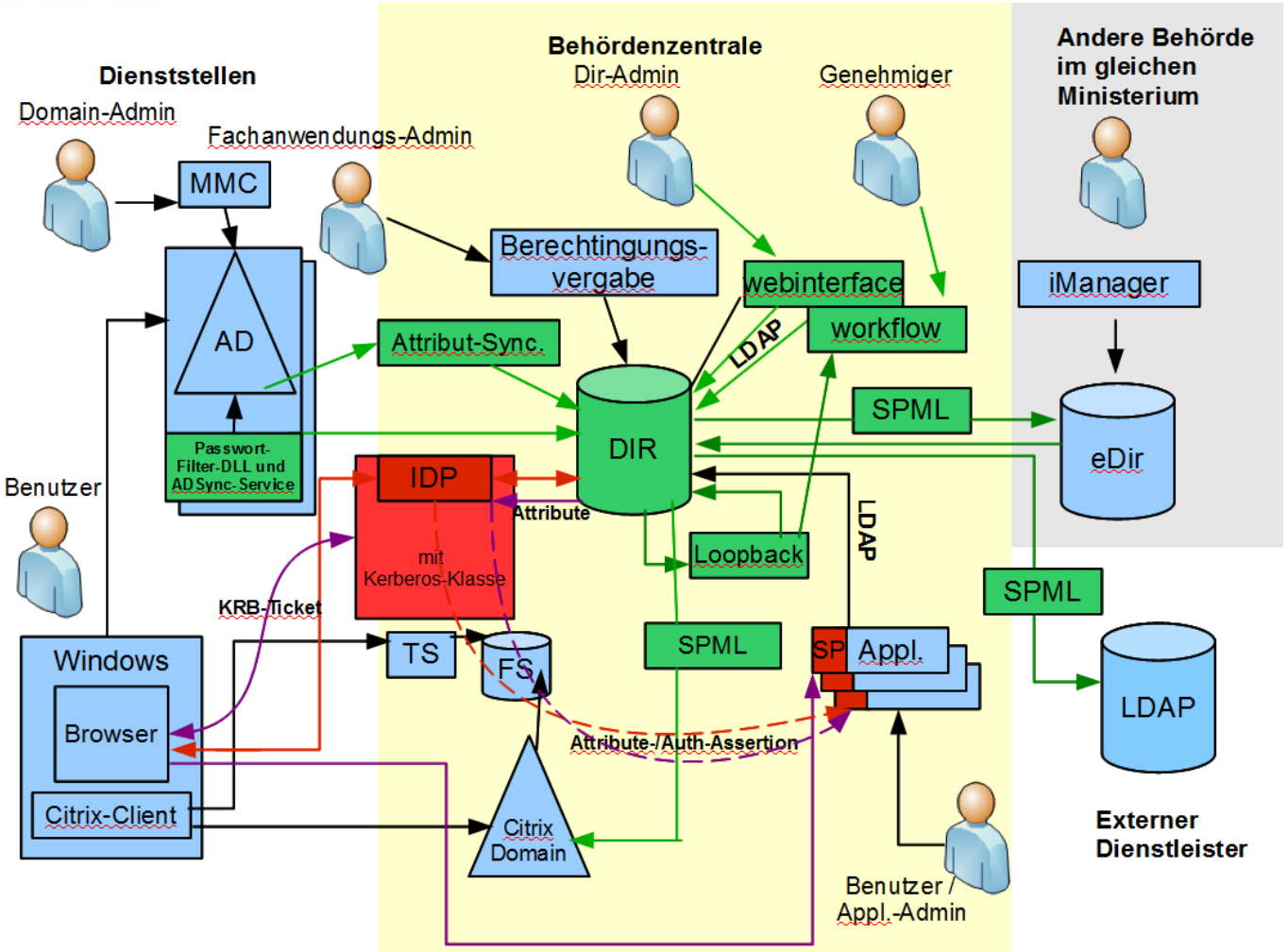
- 0: Keinerlei Zugriff
- d: Fehlermeldungen dürfen angezeigt werden
- x: Authentifizierung ist erlaubt
- c: Das Vergleichen von Attributen ist erlaubt
- s: Das Anwenden von Suchfiltern ist erlaubt
- r: Ein Lesezugriff wird gestattet
- w: Ein Schreibzugriff wird gestattet
- m: Gestattet Schreibzugriff auf operationale Attribute
- Kombinationen sind möglich, z.B. dxcs
- Reihenfolge der Angabe ist nicht relevant

DAASI International

Berechtigungen können auf diese Bereiche vergeben werden

- * alle Daten
- dn.exact=<DN>: Der DN
- dn.regex=<regex>: Alle DNs, die auf den Regulären Ausdruck passen
- dn.base=<DN>: Der DN
- dn.one=<DN>: Alle direkten Untereinträge von DN
- dn.children=<DN>: Alle Untereinträge von DN
- dn.subtree=<DN>: Der DN und alle Untereinträge

Implementierungsbeispiel



Vielen Dank für Ihre Aufmerksamkeit.

DAASI International GmbH

www.daasi.de

Telefon: 07071 4071090

E-Mail: peter.gietz@daasi.de