

13. Ldap



13. LDAP – Lightweight Directory Access Protocol

Übersicht

Lightweight **D**irectory **A**ccess **P**rotocol

LDAP wurde an der Universität Michigan entwickelt (1993/1995).

- Directory Service (Verzeichnisdienst)
- Oft lesende Zugriffe, selten schreibende Zugriffe
- Organisation im Verzeichnisbaum
- Verteilung des Verzeichnisbaums auf mehrere Rechner möglich
- Administration des Verzeichnisbaums kann auf Teilbäume aufgeteilt werden
- Suchoperation über dem Verzeichnisbaums kann auf Teilbäume eingeschränkt werden
- Wurzeln im X.500, ursprünglich als Ergänzung, jetzt eigenständig.

13. LDAP

Standards

Basis RFCs von LDAPv3

- RFC 2251 – Lightweight Directory Access Protocol (v3)
- RFC 2252 – LDAPv3 Attribute Syntax Definitions
- RFC 2253 – UTF-8 String Representation of Distinguished Names
- RFC 2254 – The String Representation of LDAP Search Filters
- RFC 2255 – The LDAP URL Format
- RFC 2256 – A Summary of the X.500(1996) User Schema for use with LDAPv3
- RFC 2307 – An Approach for Using LDAP as a Network Information Service

Weitere RFCs: 2293, 2294, 2377, 2596, 2649, 2696, 2713, 2714, 2798, 2829, 2830, 2849, 2891, 3045, 3062, 3088, 3112, 3296, 3377, 3383

Kurze, gute Anleitung: <http://www.zytrax.com/books/ldap/>

13. LDAP

Protokoll

Der Verzeichnisdienst besteht aus zwei Teilen:

1. Der Server mit der Datenbank
2. Der Client der auf die Datenbank zugreifen soll

Client und Server müssen sich mittels eines Protokolls verständigen:

Früher X.500 Protokoll Directory Access Protocol (DAP) - sehr kompliziert, jetzt LDAP. LDAP ist eine vereinfachte Form von DAP.

Das LDAP-Protokoll hat folgende Vorteile:

- Basiert auf TCP. z.Z. benutzen fast alle Netzwerke und Betriebssysteme TCP/IP. Damit kann man auf bestehende Netzwerkverbindungen aufsetzen
- Client/Server Anwendungen sind unter TCP leicht zu implementieren
- TCP Verbindungen sind schnell und zuverlässig

13. LDAP

Protokoll

Bestandteile einer LDAP-Aktionen

1. Bindung: Jede LDAP-Aktion wird mit einer Bindung eingeleitet. Dabei muss sich der Client gegenüber dem Server mittels Nutzernamen und Passwort authentifizieren. Es werden die verschiedensten Authentifizierungsverfahren unterstützt. Anhand der Bindung werden Zugriffsrechte verteilt.
2. Lesender Zugriff: Es gibt zwei lesende Zugriffsmöglichkeiten:
 - Suchen im Directory-Baum
 - Vergleichen eines vorgegebenen Wertes mit einem Wert im Baum
3. Schreibender Zugriff: Es gibt folgende schreibende Zugriffsmöglichkeiten:
 - Hinzufügen
 - Löschen
 - Ändern
 - Umbenennen

13. LDAP

Protokoll

Ablauf:

- Bindungsanforderung (Client --> Server)
- Bestätigung der Bindung (Client <-- Server)
- Senden der Anfrage (Client --> Server)
- Antwort – Daten (Client <-- Server)
- Antwort – Code (Client <-- Server)
- Verbindungsende anzeigen (Client --> Server)
- Verbindungsende bestätigen (Client <-- Server)

13. LDAP



Authentifizierung

LDAP unterstützt folgende Authentifizierungsmöglichkeiten:

SASL Simple Authentication and Security Layer (RFC 2222)

SASL dient nur der Authentifizierung nicht der Verschlüsselung

Es gibt mehrere Implementationen:

Kommerzielle SASL-Bibliotheken

Cyrus-SASL (von CMU - Carnegie Mellon University)

Bestandteile: sasldb(2), saspasswd(2), sasldblistusers(2), dbconvert, saslauthd, pwcheck

Kerberos (dreiköpfiger Hund der griechischen Sage)

Dient zur Authentifizierung und Autorisierung von Nutzern und Rechnern.

Unterstützt neben Passwörtern auch Zertifikate (Granting Tickets).

SSL/TLS dient zur Authentifizierung, Autorisierung und Verschlüsselung.

13. LDAP

Datenmodell

LDAP ist ein Verzeichnisdienst - stellt also im wesentlichen ein Verzeichnis dar.

Aufgabe eines Verzeichnisses ist es, Objekte abzubilden und miteinander in Relation zu setzen. Ein Objekt wird im Verzeichnis durch einen Verzeichniseintrag dargestellt.

Ein Verzeichniseintrag besteht aus dem Namen des Objekts (Distinguished Name = DN) und den Eigenschaften (Attributen).

Der DN wird in einen hierarchischem Namensraum eingeordnet. Dadurch entsteht die Verzeichnisstruktur (Directory Information Tree = DIT).

Verzeichniseintragungen und Eigenschaften unterliegen einer gewissen Standardisierung. Sie werden Objektklassen zugeordnet. Die Objektklassen sind standardisiert. Die Objektklassen werden in Schemas zusammengefasst und für einen LDAP-Server definiert (als Files).

13. LDAP

Datenmodell

Schema

Das Schema umfasst alle möglichen Eintragungen im DIT. Dabei wird sowohl die Struktur, die Eigenschaften als auch die Position im DIT der Eintragungen festgelegt. Die Schemas findet man unter:

`/etc/openldap/schema`

folgende elementaren Schemas gibt es:

<code>core.schema</code>	- RFC 2256, 2587, 2079, 1274, 2247, 2459
<code>cosine.schema</code>	- RFC 1274, X.500
<code>inetorgperson.schema</code>	- X.500, LDAPv3, RFC 2256
<code>nis.schema</code>	- NIS, RFC 2307
<code>misc.schema</code>	- RFC 822, Mail
<code>rfc2307bis.schema</code>	- NIS, NFS

13. LDAP

Datenmodell

Objekte

Objekte sind der wesentliche Bestandteil eines DIT. Ihre Eigenschaften werden durch die Objektklassen beschrieben. Objektklassen haben folgende Aufgaben und Eigenschaften:

Eindeutige Namensgebung: jede Objektklasse hat einen eindeutigen Namen, der nicht case-sensitive ist.

Eindeutige Identifikation: Numerische Kennung - OID (Object Identifier)

Art des Objekts: Containerobjekt, kann weitere Objekte enthalten

Blattobjekt, enthält keine weiteren Objekte

Notwendige Attribute

Mögliche Attribute

13. LDAP

Datenmodell

Objekte

Containerobjekte:

Root

Country (c)

Domain (dc)

Organization (o)

Organizationunit (ou)

Blattobjekte:

Common Name (cn)

Objekte innerhalb des DIT können mehreren Objektklassen angehören.

Datenmodell

Attribute

Durch die Zuordnung eines Objektes zu einer Objektklasse während der Erzeugung des Objektes wird festgelegt, welche Attribute für dieses Objekt benutzt werden können. Attribute haben ebenfalls eine fest vorgegebene Struktur und sind für spezielle Aufgaben vorgesehen (durch die Definition in den Schemas):

Eindeutiger Name: wie bei Objektklassen

Eindeutige Identifikation: Nur als Synonym

Syntax: Attribute können Werte zugeordnet werden. Diese Werte unterliegen einer festen Syntax, die durch die Schemas vordefiniert ist.

Werte: Wert des Attributes. In Abhängigkeit von der Definition muss für ein konkretes Objekt der Wert belegt werden oder nicht.

Datenmodell

Regeln

Beim Aufbau eines DIT sollte man gewisse Regeln beachten:

1. Oberstes Element ist ein Root-Element, das immer vorhanden sein muss.
2. Mittels Domainangaben kann der Name dem DNS angepasst werden. z.B. dc=informatik,dc=hu-berlin,dc=de. Unser Server (Namensraum) beginnt erst bei informatik.
3. Unterhalb von Root kann genau ein Country Objekt (c) auftreten, darf fehlen.
4. Nach dem Country Objekt bzw Root Objekt können Domain-Objekte (dc), Organization Objekte (o) oder Organization Unit Objekte (ou) auftreten (mehrere).
5. Unterhalb von (o) stehen Blattobjekte cn oder Containerobjekte ou (mehrere)
6. Blattobjekte können nur in o oder ou Objekten vorkommen.

13. LDAP



Datenmodell

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

Der DN beschreibt ein Objekt mit einem absoluten Namen, der einmalig im DIT ist. Der RDN beschreibt ein Objekt mit einem relativen Namen, der ab einem bestimmten Punkt im DIT, der vor einer Dereferenzierung festgelegt werden muss, beginnt. Damit kann man innerhalb des DIT gleichartige Teilbäume aufbauen. Die gleichartigen Teilbäume dürfen natürlich nicht auf der selben Ebene liegen. Die Objekte innerhalb der Teilbäume müssen bezüglich des DN einen eindeutigen Namen haben.

13. LDAP



Software

Serversoftware

- slapd - LDAP-Server
- slurpd - LDAP-Update-Server (bis Version 2.3.xx)
- slappasswd - Passwortverschlüsselung
- slapacd - Überprüfen der Zugriffsrechte für Objekte
- slapauth - Überprüfen von LDAP-Authentifizierung
- slaptest - Testen des Konfigurationsfiles
- slapadd - Hinzufügen von Eintragungen in die Datenbank (offline)
- slapcat - Ausgabe der Datenbank (Eingabestrom für slapadd)
- slapdn - Prüfprogramm für die Gültigkeit von DN's
- slapindex - Regenerierung der Datenbasis

13. LDAP



Software

Clientsoftware

- Idapsearch - Suchen von Objekten und Attributen im Verzeichnis
- Idapadd - Hinzufügen von Einträgen im Verzeichnis - online
- Idapmodify - Modifizieren von Einträgen im Verzeichnis - online
- Idapdelete - Streichen von Einträgen im Verzeichnis - online
- Idappasswd - Setzen von Passwörtern - online
- Idapcompare - Prüfen von Werten im Verzeichnis - online
- Idapmodrdn - Ändern von RDN Eintragungen - online
- Idapwhoami - Zeigt die eigene Identität an - online

13. Ldap

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers(1)

Konfigurationsfiles:

`/etc/openldap/slapd.conf` - Server

`/etc/openldap/ldap.conf` - Client

Starten des Servers:

`/etc/init.d/ldap start`

Laden der Daten in den Server:

offline:

`slapdadd`

online:

`ldapadd`

Datenwartung:

`ldapadd`, `ldapmodify`, `ldapdelete`, `ldappasswd` - alles online

13. LDAP

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (2)

```
/etc/openldap/slapd.conf  (1)
# See slapd.conf(5)
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/misc.schema
include /etc/openldap/schema/solaris.schema
include /etc/openldap/schema/rfc2307bis.schema
# schemacheck on bis 2.3.xx
pidfile   /var/run/slapd/slapd.pid
argsfile  /var/run/slapd/slapd.args
```

13. LDAP

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (3)

/etc/openldap/slapd.conf (2)

```
modulpath /usr/lib/openldap/modules
```

```
access to attrs=userPassword
```

```
    by self write
```

```
    by * auth
```

```
access to * by * read
```

```
sizelimit 20000
```

```
database bdb
```

```
suffix "dc=informatik,dc=hu-berlin,dc=de"
```

```
rootdn "cn=Manager,dc=informatik,dc=hu-berlin,dc=de"
```

```
rootpw {SSHA} 43tg/FgRco9ULJBFQWERQRa0oS8asdfqr
```

```
directory /var/lib/ldap
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (4)

/etc/openldap/slapd.conf (3)

```
index objectClass eq
TLSCipherSuite      HIGH:MEDIUM:+SSLv2:+SSLv3
TLSCACertificateFile /etc/openldap/CACerts/cacert.pem
TLSCACertificatePath /etc/openldap/CACerts
TLSCertificateFile  /etc/openldap/Certs/c1.crt
TLSCertificateKeyFile /etc/openldap/Certs/c1.key
TLSRandfile         /dev/random
TLSVerifyClient     hard
TLSCRLCheck         none
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (5)

Datenbasis initialisieren:

```
rm /var/lib/ldap/__db.*
rm /var/lib/ldap/*.db
rm /var/lib/ldap/a-lock
rm /var/lib/ldap/a-lock
/var/lib/ldap/DB_CONFIG   anlegen
    # cachsize
    set_cachsize 0 15000000 1
    set_data_dir db
    # transaction log
    set_lg_bsize 2097152
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (6)

Starten des SLAPD-Servers:

```
/etc/init.d/ldap start
```

oder für lokalen start:

```
/usr/lib/openldap/slapd -h ldap:/// -u ldap -g ldap -o slp=on
```

Nun nur noch Daten in den Server laden.

Initialisierung:

offline

Wartung:

online

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (7)

Dateninitialisierung - Hilfsmittel

Übernahme der Standardinformationen mit Migration-Tools von Luke Howard.

Folgende Files werden übernommen:

hosts, ethers, rpc, services, networks, protocols,
auto.home, auto.master, group, netgroup, passwd

Die Grundstruktur der Datenbasis wird im File `migrate_common.ph` festgelegt.

Achtung: Standardmäßig wird eine zweistufig Basis vorausgesetzt. Wenn mehr notwendig wird, ist das Script `migrate_base.pl` zu modifizieren.

Die Daten können aus den Files oder aus dem NIS bezogen werden.

Datenübernahme online wird auch unterstützt, dauert aber sehr lange.

13. LDAP

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (8)

Datenformat für Initialisierung und Update:

LDIF-Files:

Files enthalten die entsprechenden Datensätze in Klartext.

Beispiele:

Eintrag für Host:

```
dn: cn=ejoker,ou=Hosts,dc=informatik,dc=hu-berlin,dc=de
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 141.20.1.53
cn: hugate
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (9)

Datenformat für Initialisierung und Update:

Eintrag für Passwd(1):

```
dn: cn=tbell,ou=People,dc=informatik,dc=hu-berlin,dc=de
objectClass: top
objectClass: person
objectClass: organizational Person
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: tbell
cn: TESTNUTZER Bell
businessCategory: RBG
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (10)

Datenformat für Initialisierung und Update:

Eintrag für Passwd(2):

```
dn: cn=tbell,ou=People,dc=informatik,dc=hu-berlin,dc=de
objectClass: top
```

.....

```
buisnessCategory: RBG
```

```
givenName: Testnutzer
```

```
sn: Bell
```

```
mail: tbell@informatik.hu-berlin.de
```

```
userPassword: {crypt}JxDxjhl513ABkE
```

```
loginShell: /usr/bin/bash
```

```
uidNumber: 509
```

```
gidNumber: 500
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (11)

Datenformat für Initialisierung und Update:

Eintrag für Passwd(3):

```
dn: cn=tbell,ou=People,dc=informatik,dc=hu-berlin,dc=de
objectClass: top
```

...

```
gidNumber: 500
```

```
homeDirectory: /vol/home-vol1/unixsoft/tbell
```

```
gecos: TESTNUTZER Bell, RBG
```

/etc/passwd:

```
tbell:JSDxjh123ABCkE:509:500:TESTNUTZER Bell, RBG:
/vol/home-vol1/unixsoft/tbell:/usr/bin/bash
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (11)

Ergebnis Anschauen:

```
ldapsearch -x -b ou=People,dc=informatik,dc=hu-berlin,dc=de uid=tbell
```

```
ldapsearch -x -D cn=Manager,dc=informatik,dc=hu-berlin,dc=de -W \  
-b ou=People,dc=informatik,dc=hu-berlin,dc=de uid=tbell
```

```
ldapsearch -x -D uid=tbell,ou=People,dc=informatik,dc=hu-berlin,dc=de -W \  
-b ou=People,dc=informatik,dc=hu-berlin,dc=de uid=tbell
```

```
ldapsearch -x -b ou=Hosts,dc=informatik,dc=hu-berlin,dc=de cn=neu
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (12)

Datenbestand ändern

Datensatz hinzufügen

```
ldapadd -x -D cn=Manager,dc=informatik,dc=hu-berlin,dc=de -W -f host.ldif
```

host.ldif:

```
dn: cn=neu,ou=Hosts,dc=informatik,dc=hu-berlin,dc=de
```

```
objectClass: top
```

```
objectClass: ipHost
```

```
objectClass: device
```

```
ipHostNumber: 141.20.23.12
```

```
cn: neu
```

Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (13)

Datenbestand ändern:

Datensatz ändern:

Datensatz löschen

```
ldapdelete -x -D cn=Manager,dc=informatik,dc=hu-berlin,dc=de -W \  
cn=neu,ou=Hosts,dc=informatik,dc=hu-berlin,dc=de
```

13. LDAP



Konfigurationsbeispiel

Konfiguration eines LDAP-Servers (14)

Datenbestand ändern:

Datensatz ändern:

Datensatz modifizieren

```
ldapmodify -x -D cn=Manager,dc=informatik,dc=hu-berlin,dc=de -W <<END
dn: cn=neu,ou=Hosts,dc=informatik,dc=hu-berlin,dc=de
replace: ipHostNumber
ipHostNumber: 141.20.20.23
END
```

13. LDAP



Probleme

Zugriffskontrolle

Replica

Sicherheit

Partitionierung

13. LDAP

Probleme

Zugriffskontrolle

TCP-Wrapper:

/etc/hosts.allow

slapd: LOCAL informatik.hu-berlin.de cms.hu-berlin.de

slapd: 141.20.20.0/255.255.255.0 141.20.21.0/255.255.255.0

SLAPD:

/etc/openldap/slapd.conf

access-Anweisungen: Manuals: slapd.conf, slapd.access

13. LDAP

Probleme

Zugriffskontrolle

SLAPD:

/etc/openldap/slapd.conf

Beispiel:

access to *

by dn.base="cn=Manager,dc=cs,dc=hu-berlin,dc=de" write

by dn="uid=samba,ou=People,ou=all,dc=cs,dc=hu-berlin,dc=de" read

by * break

.....

13. LDAP

Probleme

Zugriffskontrolle

SLAPD:

/etc/openldap/slapd.conf

Beispiel:

access to *

.....

access to attrs=userPassword

by dn="uid=admsar,ou=People,ou=sar,dc=cs,dc=hu,dc=de" none break

by dn="uid=admall,ou=People,ou=all,dc=cs,dc=hu,dc=de" auth break

by dn="cn=proxyagent,ou=profile,dc=cd,dc=hu,dc=de" auth

by self read

by * auth

.....

13. LDAP



Probleme

Zugriffskontrolle

SLAPD:

/etc/openldap/slapd.conf

Beispiel(2)

```
access to dn.subtree="ou=all,dc=cs,dc=hu,dc=de"
```

```
by dn="uid=admsar,ou=People,ou=sar,dc=cs,dc=hu,dc=de" none break
```

```
by dn="uid=admall,ou=People,ou=all,dc=cs,dc=hu,dc=de" write
```

```
by peername.ip=141.20.20.20.0%255.255.255.0 read
```

```
by peername.ip=141.20.20.20.0%255.255.255.0 read
```

```
by peername.ip=127.0.0.0%255.255.255.0 read
```

```
by * none
```

```
access to dn.subtree="ou=sar,dc=cs,dc=hu,dc=de"
```

....

13. LDAP

Probleme

Replizierung

Master-Ldap-Server kann repliziert werden – Replica ist ebenfalls ein slapd.
Entsprechende Eintragungen müssen in den slapd.conf-Files vorgenommen werden. Der Update der Replica erfolgt zeitnah mit der Änderung im Master-Ldap-Server.

Es gibt verschiedene Verfahren für die Replizierung:

- Bis Version 2.3.xx - slurpd. Änderungen werden auf dem Master-Server gespeichert und bei entsprechendem Kontakt zum Replica nachgezogen. Für den Update der Replica ist der Daemon slurpd zuständig. Auf dem Masterserver muss für jeden Replica ein Eintrag vorhanden sein. Ein Entsprechenden Eintrag für den Master-Server muß auf dem Replica vorhanden sein. Achtung: Die Passwörter für die Replica liegen beim Master-Server im Klartext vor!!
- Ab Version 2.3.xx – syncrepl. Der Replica holt die Änderungen vom Master (refreshOnly und refreshAndPersist)

13. LDAP

Probleme

Replizierung

slurpd-Konfiguration(1)

slapd.conf (master):

```
replicationinterval 300
replica uri=ldaps://repl201.informatik.hu-berlin.de
  bindmethod=simple
  binddn="cn=updater,dc=cs,dc=hu-berlin,dc=de"
  credentials=geheim
repllogfile /var/log/replogs
```

13. LDAP

Probleme

Replizierung

slurpd-Konfiguration(2)

slapd.conf (replica):

```
rootdn cn=updater,dc=informatik,dc=hu-berlin,dc=de
```

```
rootpw {SSHA} qewrqewrfdslasjfqelwfr
```

```
updatedn „cn=updater,dc=cs,dc=hu-berlin,dc=de“
```

```
access to *
```

```
by dn.base="cn=updater,dc=cs,dc=hu-berlin,dc=de" write
```

```
by * break
```

13. LDAP

Probleme

Replizierung

syncrepl-Konfiguration(1)
refreshOnly

slapd.conf (master):

```
moduleload syncprov.1a
```

```
index entryCSN eq
```

```
index entryUUID eq
```

```
access to *
```

```
by dn.base="uid=rupdater,ou=People,ou=all,dc=cs,dc=hu-berlin,dc=de" read
```

```
by * break
```

```
overlay syncprov
```

13. LDAP



Probleme

Replizierung

syncrepl-Konfiguration(2)

refreshOnly

slapd.conf (replica):

index entryCSN eq

index entryUUID eq

syncrepl rid=101

provider=ldaps://master.cs.hu-berlin.de

type=refreshOnly

retry="120 5 100 +"

searchbase="dc=cs,dc=hu-berlin,dc=de"

bindmethode=simple

13. LDAP

Probleme

Replizierung

syncrepl-Konfiguration(3)

refreshOnly

slapd.conf (replica):

syncrepl rid=101

...

binddn="cn=rupdater,ou=People,ou=all,dc=cs,dc=hu-berlin,dc=de"

credentials=geheim

13. LDAP

Probleme

Replizierung

syncrepl-Konfiguration(4)

RefreshAndPersist

slapd.conf (master):

```
moduleload syncprov.1a
```

```
index entryCSN eq
```

```
index entryUUID eq
```

```
access to *
```

```
    by dn.base="cn=rupdater,ou=People,ou=all,dc=cs,dc=hu-berlin,dc=de" read
```

```
    by * break
```

```
overlay syncprov
```

```
syncprov-checkpoint 1 120
```

13. LDAP



Probleme

Replizierung

syncrepl-Konfiguration(5)
refreshAndPersist

slapd.conf (replica):

index entryCSN eq

index entryUUID eq

syncrepl rid=102

provider=ldaps://master.cs.hu-berlin.de

type=refreshAndPersist

retry="120 5 100 +"

searchbase="dc=cs,dc=hu-berlin,dc=de"

bindmethode=simple

13. LDAP

Probleme

Replizierung

syncrepl-Konfiguration(6)
refreshAndPersist

slapd.conf (replica):

syncrepl rid=102

...

binddn="cn=rupdater,ou=People,ou=all,dc=cs,dc=hu-berlin,dc=de"
credentials=geheim

13. LDAP

Probleme

Sicherheit

ACLs (schon behandelt)

TLS, SSL

CA-Infrastruktur notwendig

Server-Authentifizierung (möglich, empfehlenswert)

Client-Authentifizierung (möglich, aufwendig)

13. LDAP



Probleme

Partitionierung

13. LDAP



Beispiel