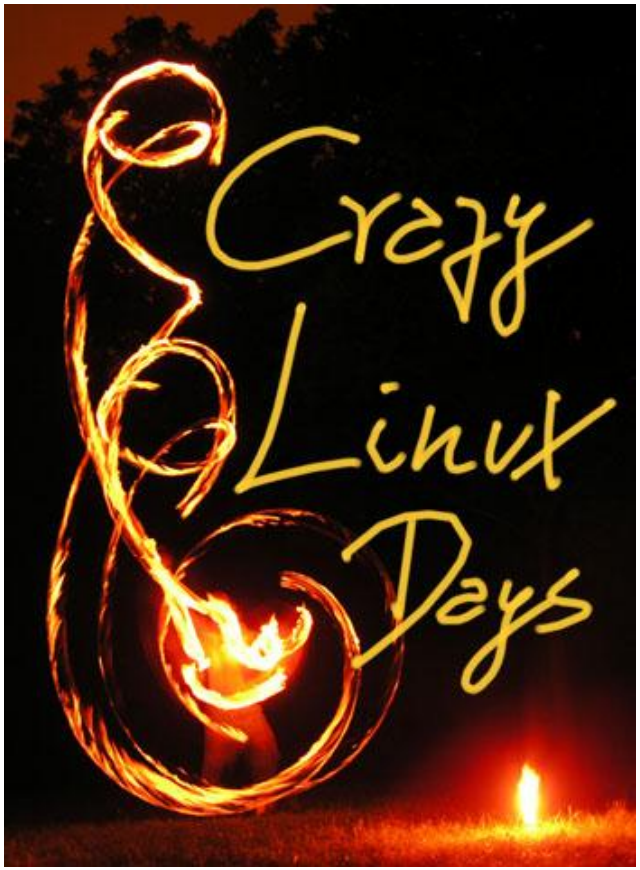


OpenLDAP Administration

- Grundlagen
- LDAP Schema
- Administration
- Linux Services
- Overlays

Peter Jahn, MSc
Jahn@LinuxCampus.net

Grundlagen Verzeichnisdienste

Traditionelle Benuzerverwaltung

- Traditionelle Benutzerverwaltung in Linux
  - Benutzer, Gruppen und Passwörter werden in einfachen und flachen Textdateien verwaltet

```
peter@tux> cat /etc/passwd
peter:x:1001:100:Peter Jahn:/home/peter:/bin/bash
```

```
peter@tux> cat /etc/shadow
peter:iXcPgb12JPLMq:12518:0:99999:7:::
```

```
peter@tux> cat /etc/group
users:x:100
audio:x:17:peter
power:x:110:peter,dafina,harald
```

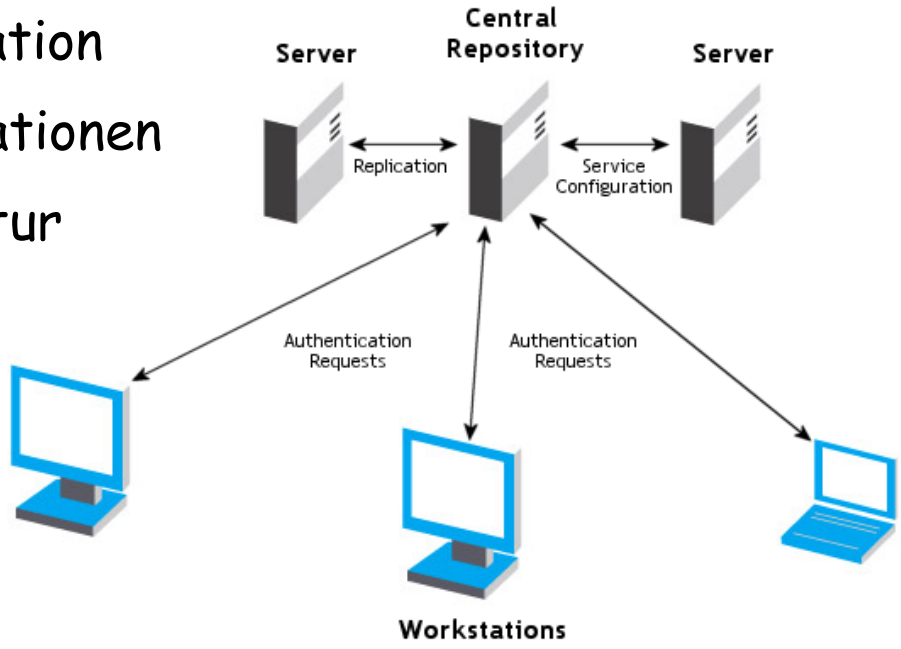
Traditionelle Benuzerverwaltung

- Nachteile dieses Ansatzes
  - Flaches Format ohne Hierarchie
  - Keine Erweiterungen der Attribute (Spalten) möglich
  - Dateien werden im lokalen Dateisystem gespeichert und nicht auf andere Server repliziert
  - Linux spezifisch und nicht ideal für Heterogene Umgebungen

```
peter@tux> cat /etc/passwd
peter:x:1001:100:Peter Jahn:/home/peter:/bin/bash
```

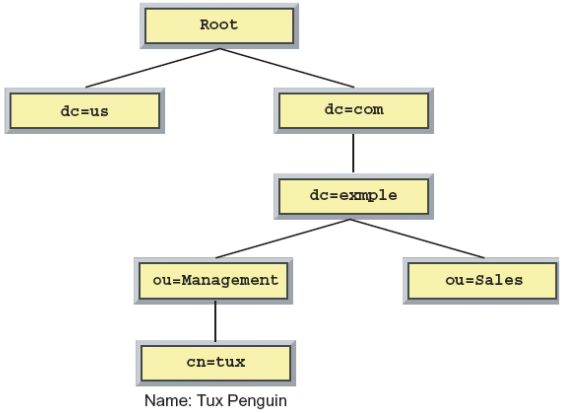
Verzeichnis Dienste

- Vorteile eines Verzeichnisdienstes
  - Unterstützung von unterschiedlichen Betriebssystemen
  - Möglichkeit zur Erweiterung des Schemas
    - Objekte und Attribute
  - Single Point of Administration
  - Unterstützung von Replikationen
  - Hierarchische Baumstruktur



LDAP Anwendungsgebiete

- Benutzer Verwaltung
  - Linux Login
  - Samba
  - Apache
  - Squid
- Service Integration
  - DHCP/DNS
  - Kerberos
- Single SignOn
  - Kerberos

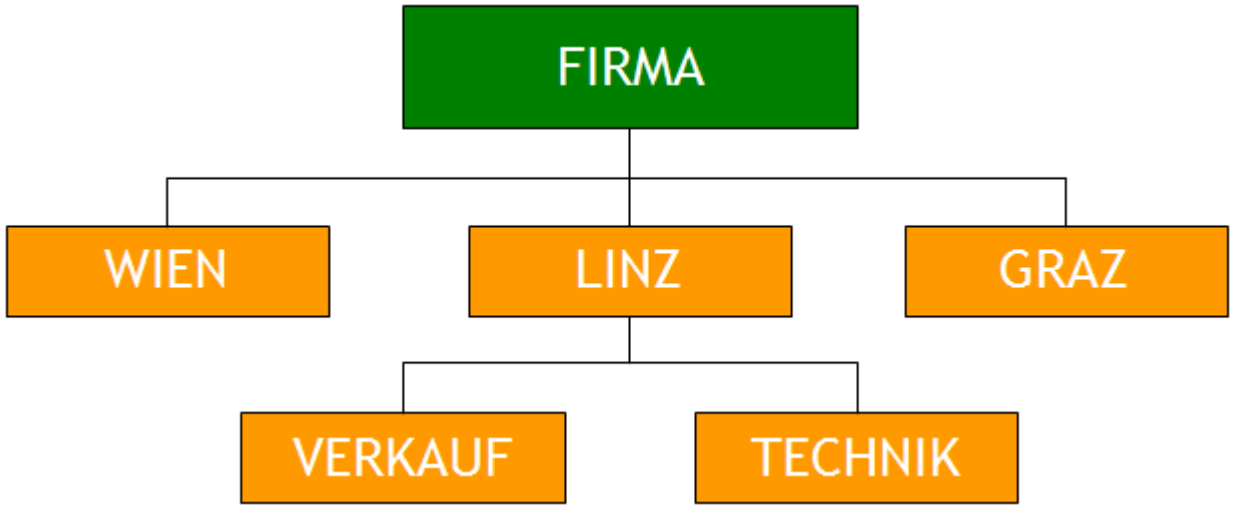


LDAP

- Lightweight Directory Access Protocol (LDAP)
  - ist eine Sammlung von Protokollen und wurde entwickelt um auf Objekte in einem Directory zugreifen zu können
    - -> LDAP ist keine Datenbank
  - Der Zugriff auf das Directory kann abhängig von der Konfiguration verschlüsselt oder im Klartext erfolgen
    - Destination Port 389 oder 636
  - Aktuell gibt es LDAPv3 RFC 4510
    - http://tools.ietf.org/html/rfc4510

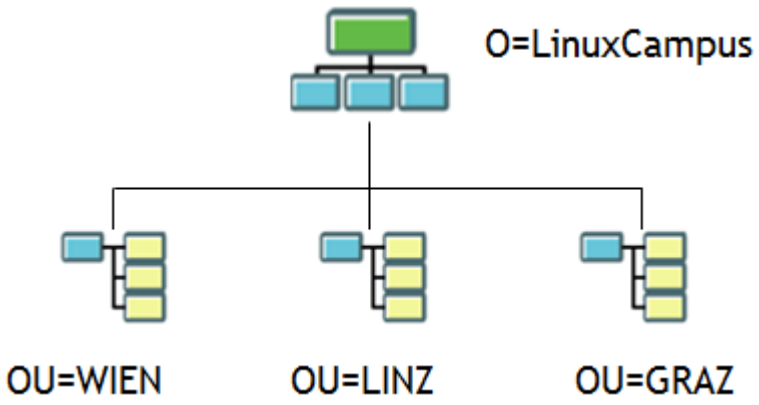
Firmenstruktur

- Hierarchische Firmenstruktur
  - Jede Firma ist in Standorte und Abteilungen aufgeteilt
  - Die Struktur wird in einem Organigramm dargestellt

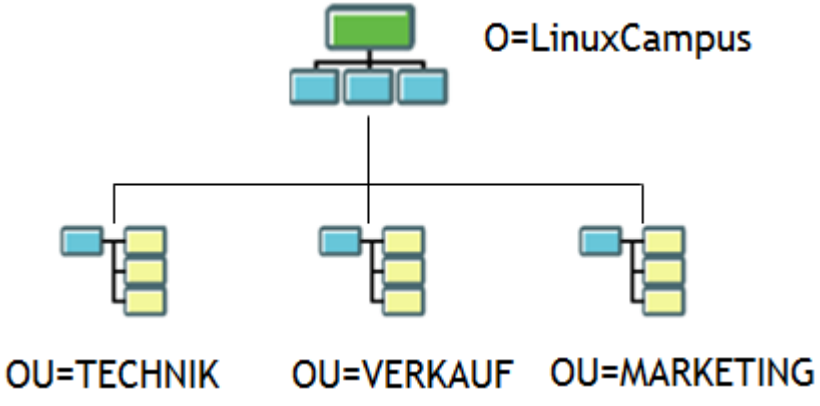


Standort- vs. Abteilungsdesign

- Struktur im Verzeichnisdienst
  - basiert meistens auf dem Organigramm des Unternehmens



Standort basierendes Design

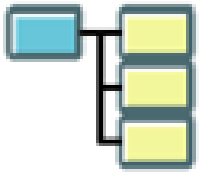


Abteilung basierendes Design

Unterschied zwischen O und OU

O=Organisation

- Zwingend erforderlich
- Kann nicht unter einer OU erstellt werden



OU=Organisation Unit

- Ist optional
- Kann unter einer OU erstellt werden

Beide Objekte dienen dazu um eine logische Struktur aufzubauen, welche die Verwaltung, Replikation, etc. erleichtert.

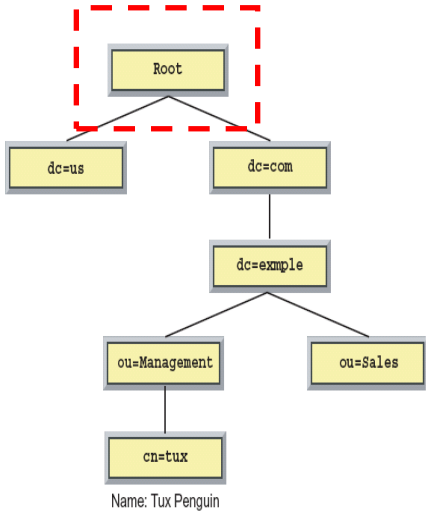
LDAP Directory Tree Struktur

- LDAP benutzt eine hierarchische Baumstruktur
  - Alle Objekte haben eine eindeutige Position in der Hierarchie
  - Der komplete Pfad vom Root bis zum jeweiligen Objekt wird Distinguished Name oder DN bezeichnet

- Distinguished Name
  - ein DN muss eindeutig im Directory Tree sein
    - dn: cn=peter,ou=wien,o=linuxcampus
    - dn: cn=peter,ou=graz,o=linuxcampus

Definition der Base DN

- Basis besteht aus Domain Components
  - dc=domain,dc=land
    - dc=triples,dc=at



- Basis besteht aus einer Organisation
  - o=organisation
    - o=triples

Installation

OpenLDAP

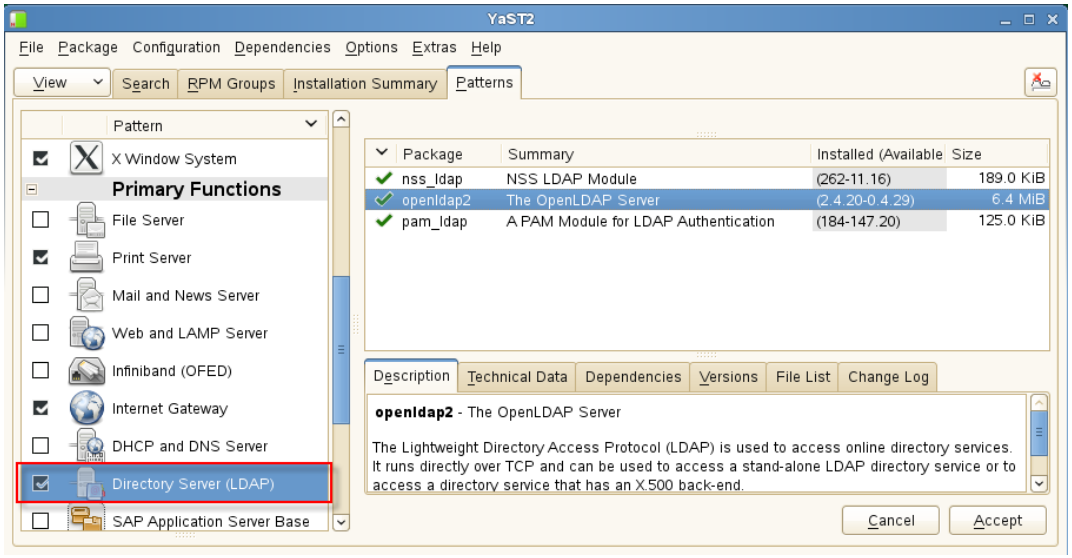
- Was ist OpenLDAP
  - eine Implementierung von LDAP, die als freie Software unter der der BSD-Lizenz ähnlichen OpenLDAP Public License veröffentlicht wird.
  - OpenLDAP ist Bestandteil der meisten aktuellen Linux-Distributionen und läuft auch unter verschiedenen Unix-Varianten, Mac OS X und verschiedenen Windows-Versionen.
  - Da OpenLDAP den LDAP-Standard verfolgt, ist es mit OpenLDAP möglich, eine zentrale Benutzerdatenverwaltung aufzubauen und zentral zu warten.

Bestandteile

- Bestandteile von OpenLDAP
  - lapd –LDAP daemon
  - backends – über diese wird der eigentliche Zugriff auf die Daten realisiert
  - overlays – ermöglichen das Verhalten der backends und damit des slapd zu modifizieren, ohne diese(n) selbst zu ändern
  - syncrepl – Synchronisation und Replikation gemäß RFC 4533
    - slurpd – stand-alone LDAP update replication daemon wird ab v2.4 nicht mehr mitgeliefert
  - Bibliotheken, die das LDAP-Protokoll bereitstellen
  - Werkzeuge, Hilfsmittel und Beispiele

Installation auf SUSE

- Installationspakete für SLES11SP1-64Bit
  - nss_ldap
  - openldap2
  - pam_ldap
  - *pam_ldap-32bit*
  - *nss_ldap-32bit*

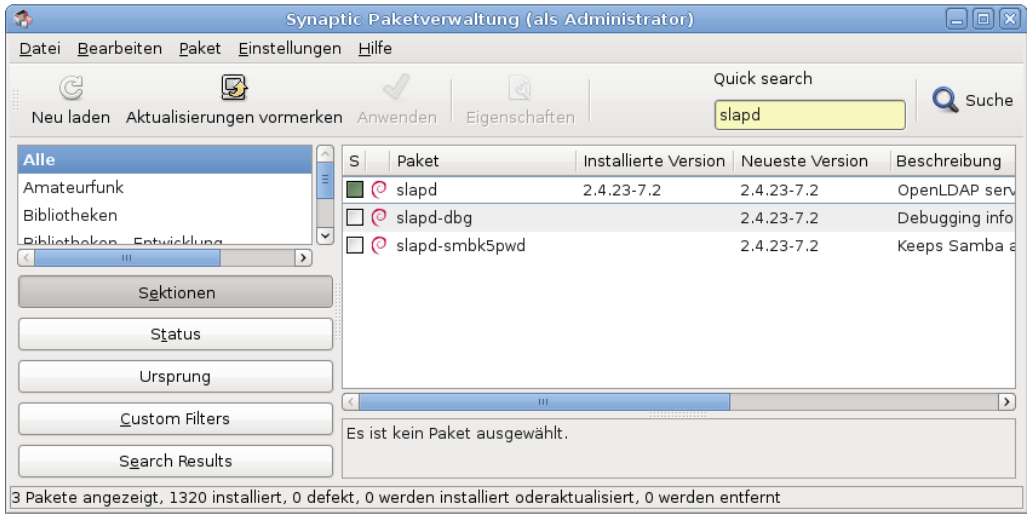


SUSE

Konfigurations-, Schema und Datenbankdateien	
/etc/sysconfig/ldap /etc/sysconfig/openldap	SUSE spezifische Konfigurationsdateien für YAST und SuSEconfig
/etc/openldap/slapd.conf	OpenLDAP Server Konfigurationsdatei
/etc/openldap/slapd.d/*	Online Konfigurationsordner ab v2.3
/etc/openldap/ldap.conf	Systemweite Einstellungen für LDAP Clients
/etc/openldap/schema/*	Schemadateien
/etc/ldap.conf	LDAP Konfigurationsdatei für die PADL Module pam_ldap und libnss-ldap
/var/lib/ldap/*	Datenbankdateien

Installation auf DEBIAN

- Installationspakete für DEBIAN
  - slapd
  - ldap-utils
  - libldap
  - libnss-ldap
  - libpam-ldap
  - ldapvi



www.LinuxCampus.net

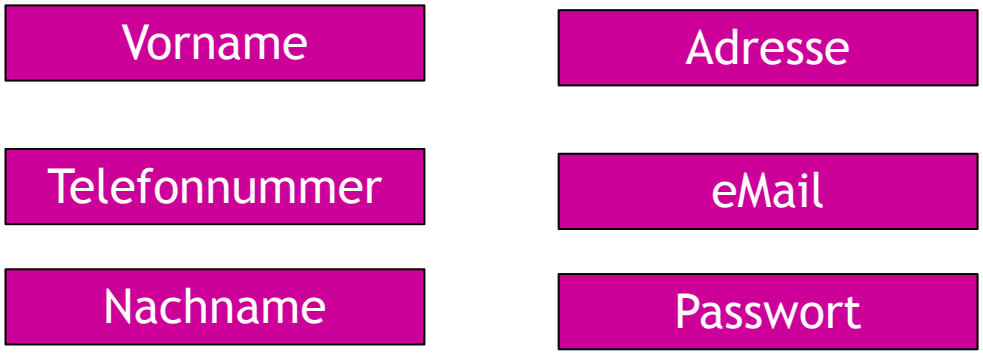
DEBIAN

Konfigurations-, Schema und Datenbankdateien	
/etc/ldap/slapd.conf	OpenLDAP Server Konfigurationsdatei
/etc/ldap/slapd.d/*	Online Konfigurationsordner ab v2.3
/etc/ldap/ldap.conf	Systemweite Einstellungen für LDAP Clients
/etc/libnss-ldap.conf	Konfigurationsdatei für libnss-ldap
/etc/pam_ldap.conf	Konfigurationsdatei für pam_ldap
/etc/ldap/schema/*	Schemadateien
/etc/default/slapd	Debian spezifische Konfigurationsdatei für ldap
/var/lib/ldap/*	Datenbankdateien
/usr/lib/ldap/*	Standard Modulpfad bei Debian

Schema Grundlagen

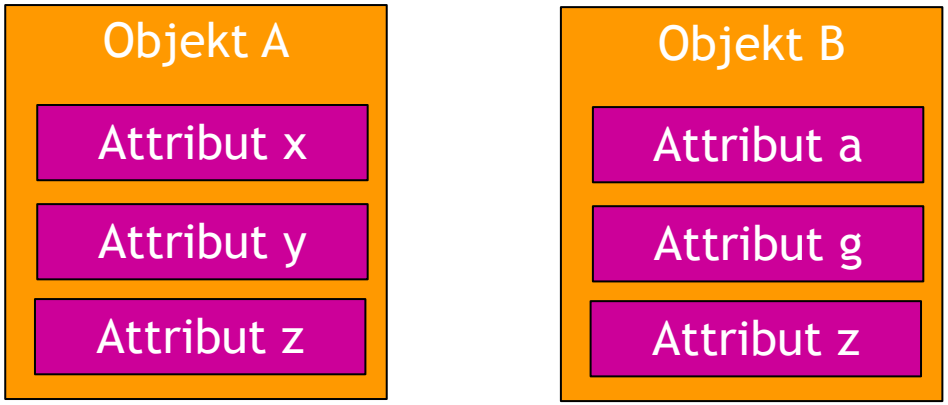
Objekte und Schema

- Attribute
  - sind einzelne Felder welche als Speicherorte dienen
  - Bei der Definition eines Attributes wird genau bestimmt welche Werte sie beinhalten dürfen

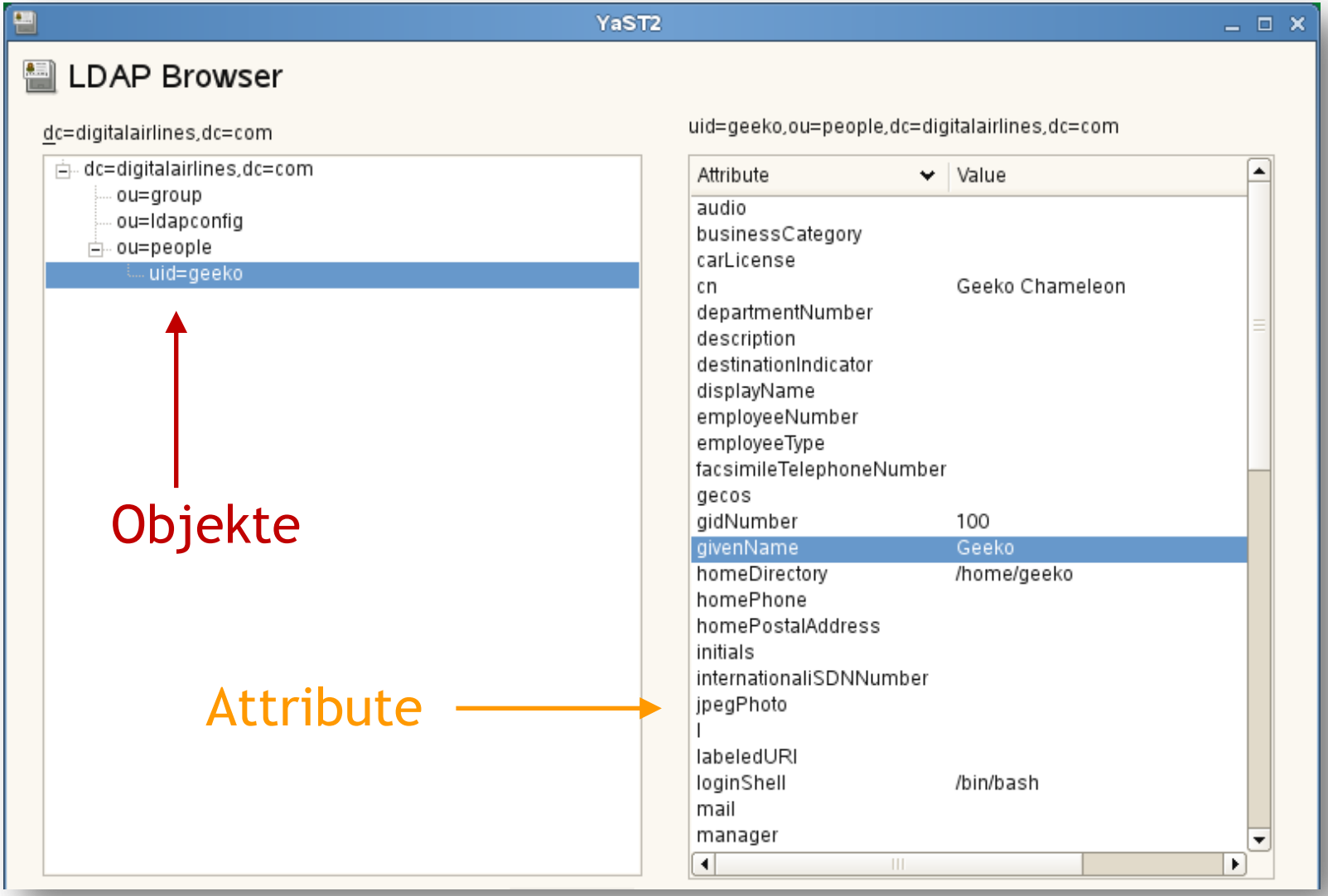


Objekte und Schema

- Objektklassen
  - wenn man eine Ansammlung von Attributen definiert hat kann man diese dann zu Objekten zusammenfügen
  - Die Beschreibung von Objekten wird als Objektklasse bezeichnet

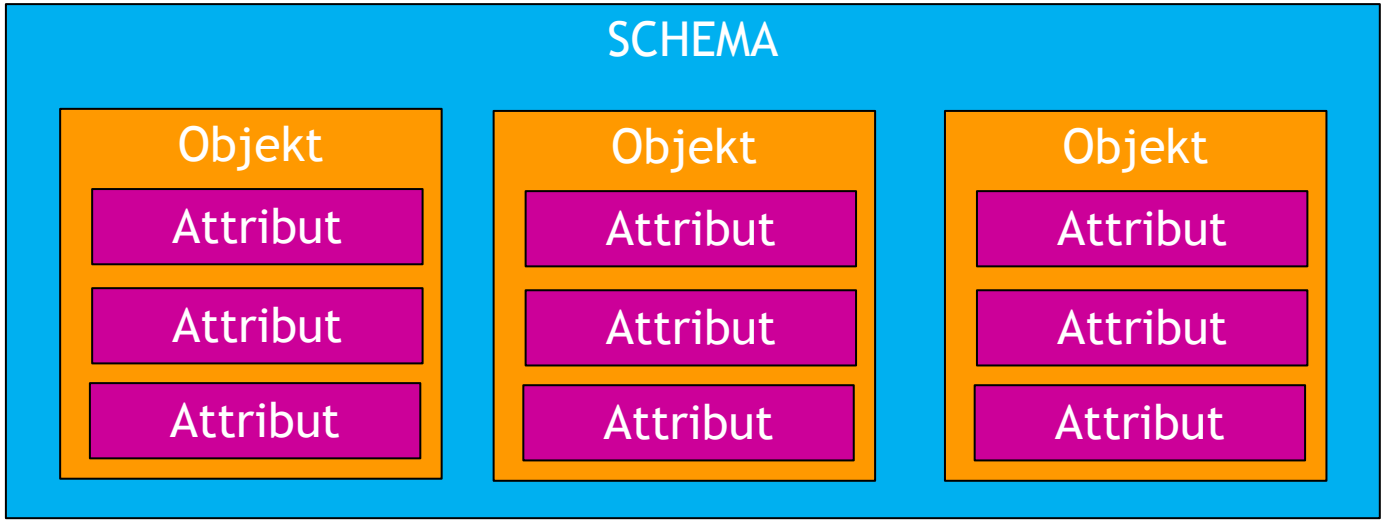


Objekte und Attribute



Objekte und Schema

- Schema
  - eine Sammlung von Objektklassen und Attributen wird als Schema bezeichnet
  - Eine Schema ist wichtig damit die Benutzer gezwungen werden sich an Syntax Regeln zu halten und um Wildwuchs zu vermeiden



Schema Dateien

- OpenLDAP Schema Dateien
  - liegen unter /etc/(open)ldap/schema/

```
sles11sp1: # ls /etc/openldap/schema
collective.schema    dyngroup.schema     openldap.ldif
corba.schema         inetorgperson.ldif  openldap.schema
...
```

  - benötigte Schemadateien werden via slapd.conf aktiviert

```
sles11sp1: # cat /etc/openldap/slapd.conf
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
...
```

Die wichtigsten Schemas

Schema	Funktion
core.schema	Grundlegende Klassen und Attribute die gemäß X.501 Standard eingebunden werden müssen
cosine.schema	Standardattribute von LDAPv3
inetorgperson.schema	Organisationsorientierte Attribute Abhängig von core und cosine
nis.schema	Attribute von NIS, Standard für Linux User
openldap.schema	Zusatzattribute für OpenLDAP Abhängig von core, cosine und inetorgperson
samba3.schema	Samba spezifische Attribute
dyngroup.schema	dynamische Gruppen Objekte

Aufbau einer Schemadatei

- /etc/openldap/schema/nis.schema
  - Als erstes werden die Attribute und danach die Objektklassen definiert. Die Reihenfolge ist extrem wichtig!

```
attributetype (a)
attributetype (b)
attributetype (...)

objectclass (A
        ...
        MUST a
        MAY b
)
```

/etc/openldap/schema/nis.schema

```
# Als erstes kommen mehrere Attribute Definitionen
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
    DESC 'An integer uniquely identifying a user in an administrative domain'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

# Danach kommen eine oder mehrere Objektklassen Definitionen
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount'
      DESC 'Abstraction of an account with POSIX attributes'
      SUP top AUXILIARY
      MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
      MAY ( userPassword $ loginShell $ gecos $ description ) )
```

Attributdefinitionen Syntax

```
attributetype ( OID NAME ( 'Name'  'Aliasname' )
    [ DESC 'Beschreibung des Objektes' ]
    [ SUP  Name oder OID der Mutterklasse ]
    [ EQUALITY Prüfverfahren für Gleichheitsprüfung ]
    [ ORDERING  Prüfverfahren für das Sortieren ]
    [ SUBSTR Prüfverfahren für Teilvergleiche ]
    [ SYNTAX Syntax-OID {Datenlänge in Bytes} ]
    [ SINGLE-VALUE ]
    [ NO-USER-MODIFICATION ]
    [ USAGE Verwendungszweck ]
    [ MUST verplichtende Attribute ]
    [ MAY  optionale Attribute ] )
```

Attribute in Objektklassen

- objectClass: top (core.schema)
  - hat nur das Attribut objectclass,
    - alle Objekte werden von top abgeleitet
- objectClass: posixAccount (nis.schema)
  - hat die Attribute eines UNIX Accounts
    - cn, uid, uidNumber, gidNumber, homeDirectory,...
- objectClass: inetOrgPerson (inetorgperson.schema)
  - Organisationsorientierte Attribute
    - eMail, departmentNumber, givenName, homePhone, manager, photo, roomNumber, mobile, carLicense,...

MUST und MAY Attribute

```
dn: uid=tux,ou=verkauf,dc=local,dc=site
objectClass: posixAccount
objectClass: inetOrgPerson
cn: Linux Tux
uidNumber: 901
gidNumber: 100
sn: Kent
homeDirectory: /home/tux
telephoneNumber: 080098769901
loginShell: /bin/bash
mail: tux@verkauf.local.site
title: admin
```

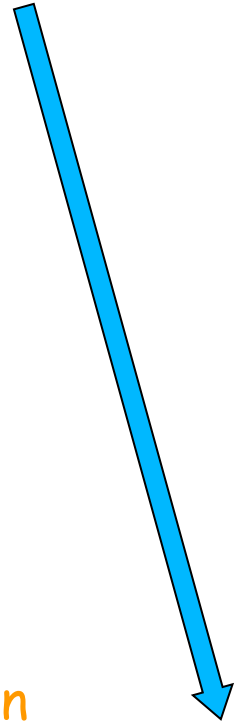
MUST Attribute
müssen definiert werden
-> Mandatory Attributes

MAY Attribute
können definiert werden
-> Optional Attributes

Objektklassen Vererbung

- Die Vererbung im Detail

```
objectClass:  top
# ABSTRACT, nur zur Vererbung
objectClass: person
# STRUCTURAL, abgeleitet von top
objectClass: organizationalPerson
# STRUCTURAL, abgeleitet von person
objectClass: inetOrgPerson
# STRUCTURAL, abgeleitet von organizationalPerson
```



Objektklassen Vererbung

```
# core.schema
objectclass ( 2.5.6.6 NAME 'person'
        DESC 'RFC2256: a person'
        SUP top STRUCTURAL
        MUST ( sn $ cn )
        MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

- SUP top
  - besagt, dass dieses Objektklasse von der übergeordneten (SUPerior) abstrakten Objektklasse top abgeleitet wird und automatisch ihre Attribute erbt
- STRUCTURAL
  - besagt, dass person eine strukturelle Objektklasse darstellt

OpenLDAP Basis Konfiguration

/etc/openldap/slapd.conf

- slapd.conf
  - Bis v2.3 war das die einzige Form wie ein OpenLDAP Server verwaltet wurde. Ab v2.3 gibt es zusätzlich auch noch die Konfiguration zur Laufzeit (/etc/openldap/slapd.d/)
    - Leider stehen noch nicht alle Funktionen der sladp.conf auch in der Online Variante zur Verfügung
  - Es kann also sein, dass eine Mischkonfiguration via slapd.conf + der neuen Variante gemacht wird, oder nur noch über die neue Variante ohne einer slapd.conf

Sektionen in slapd.conf

- slapd.conf wird üblicherweise in 3 Sektionen unterteilt

```
# /etc/openldap/slapd.conf
#Globale Konfigurationsdirektiven
....

#Backend Konfigurationsdirektiven
backend <backend-typ>
...

#Datenbank spezifische Direktiven
database <database-typ>
...
```

- Die Direktiven in slapd.conf bilden eine strenge Hierarchie
  - Globale Direktiven können durch Backend und Database Direktiven überschrieben werden
  - Backend Direktiven können durch Database Direktiven überschrieben werden

Wichtige Syntax Regeln

- #
  - Zeilen die mit einer Raute beginnen werden als Kommentar interpretiert. Befindet sich die Raute nicht am Zeilenanfang sondern ist eingerückt wird diese Zeile NICHT als Kommentar interpretiert!
    - Richtig: |#... Falsch: | #...

```
# Sample access control policy
access to dn.base=""
        by * read
```

Wichtige Syntax Regeln

- Leerzeichen oder Einrückungen
  - Zeilen die mit einem Leerzeichen beginnen oder eingerückt sind werden vom slapd immer als Folgezeile der zuvor verwendeten Direktive interpretiert
    - |hier beginnt die erste Zeile
    - | das ist die Fortsetzung der ersten Zeile

```
# Sample access control policy
access to dn.base=""
        by * read
```



```
# Sample access control policy
access to dn.base="" by * read
```

Übung

- Vorbereiten der Server Konfiguration
  - Austauschen der Konfigurationsdateien
    - /etc/openldap/ldap.conf ...rootpw=linux
    - /etc/openldap/slapd.conf
  - Überprüfen der Rechte und Besitzer

```
sles11sp1:/etc/openldap # ll
total 16
-rw-r--r-- 1 root root  264 2010-05-05 16:23 ldap.conf
drwxr-xr-x 2 root root 4096 2012-01-02 08:25 schema
-rw-r----- 1 root ldap 2538 2010-05-05 16:27 slapd.conf
drwxrwx--- 2 ldap ldap 4096 2010-05-05 16:28 slapd.d
```

Überprüfen der Konfiguration

- slaptest
  - Mit Hilfe dieses Tools kann überprüft werden ob der Syntax der slapd.conf Datei in Ordnung ist

```
# Überprüfen der Konfigurationssyntax
peter@tux~> slaptest
bdb_db_open: database "dc=my-domain,dc=com":
db_open(/var/lib/ldap/id2entry.bdb) failed: No such file or directory (2).
backend_startup_one (type=bdb, suffix="dc=my-domain,dc=com"):
bi_db_open failed! (2)
slap_startup failed (test would succeed using the -u switch)
```

Übung

- Starten des LDAP Servers
  - /etc/init.d/ldap start ...SUSE
  - /etc/init.d/slapd start ...Debian

- Falsches Dateiformat
  - falls beim starten Fehlermeldungen kommen kann es sein dass die Konfigurationsdateien vorher mit dem Befehl dos2unix dateiname konvertiert werden müssen

```
sles11sp1:/etc/openldap # rcldap start
: No such file or directorycore.schema
: No such file or directorycore.schema
: No such file or directorycosine.schema
```

Einträge in slapd.conf

- include /etc/openldap/schema/core.schema
  - Über include Direktiven werden am Anfang der Konfiguration Schemadateien oder weitere Konfigurationsdateien eingebunden
  - Die Reihenfolge der eingebunden Schemadateien ist sehr wichtig und muss auch auf dem Master und Slave Server gleich sein

```
# /etc/openldap/slapd.conf
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
```

Einträge in slapd.conf

- access to dn.base="" by * read
  - Die Access Direktiven dienen der Zugriffsregelung auf den Directory Information Tree (DIT)
  - ACLs bestimmen WER (welches Objekt) WIE (lesend, schreibend, authentifizierend) auf WAS (welche Objekte) im Verzeichnis zugreifen darf
  - access to <what> [by <who> [<accesslevel>] [<control>]]+
  - Standardmäßig darf zunächst nur der rootdn schreibend auf den DIT zugreifen
    - rootdn darf implizit immer alles tun

Einträge in slapd.conf

- defaultsearchbase dc=local,dc=site
  - kümmert sich um Client Anfragen bei denen keine BaseDN übermittelt wurde = Einstiegspunkt im DIT
  - Analog dazu gibt es auch in der LDAP Client Konfigurationsdatei /etc/openldap/ldap.conf dem Eintrag base dc=local,dc=site

```
# /etc/openldap/ldap.conf
BASE dc=local, dc=site
```

Einträge in slapd.conf

- database hdb
  - diese Direktive definiert die Schnittstelle zur eigentlichen Datenbank unseres LDAP Servers
    - suffix "dc=local,dc=site"
    - rootdn "cn=ldapadmin,dc=local,dc=site"
    - directory /var/lib/ldap

Einträge in slapd.conf

- cachesize 10000
  - legt die Größe eines Objekt bezogenen Cache fest der vom jeweiligen Backend im Speicher des slapd Hosts verwaltet wird
  - dieser Wert sagt aus wie viele Objekte des DIT der slapd im Cache für einen schnellen Zugriff vorbehalten soll

Einträge in slapd.conf

- suffix "dc=local,dc=site"
  - definiert die BaseDN also den höchsten sichtbaren Punkt in unserem DIT von dem aus auf alle weiteren Bereiche verzweigt wird
  - Der Suffix besteht üblicherweise aus einem oder mehreren Komponenten
    - o=linuxcampus
    - dc=linuxcampus,dc=net

Einträge in slapd.conf

- checkpoint 1024 5
  - legt die Frequenz fest in welchen Intervallen Wiederherstellungs-/Prüfpunkte im Transaktionslog der unterliegenden BDB-Datenbank gesetzt werden
  - Im Falle eines Systemausfalls bestimmt dieser Eintrag wie viel an Daten im schlimmsten Fall verloren gehen könnten
    - 1024 KByte oder 5 Minuten was zuerst eintrifft

Einträge in slapd.conf

- rootdn "cn=ldapadmin,dc=local,dc=site"
  - legt hartverdrahtet den LDAP Administrator fest
  - dieser User darf jederzeit alle Änderungen am Server vornehmen

- rootpw {SSHA}rawwGsSrghV56gsfhbatah
  - dieser Eintrag bestimmt entweder im Klartext oder noch besser in verschlüsselter Form das Kennwort für dem rootdn
  - Ein verschlüsseltes Kennwort kann wie folgt erstellt werden
    - slappasswd -s [-h {encrypttype}] klartextpasswort

Einträge in slapd.conf

- /var/run/slapd/slapd.args
  - diese Datei beinhaltet Parameter mit denen der slapd Prozesse gestartet wird

- pidfile /var/run/slapd/slapd.pid
  - die PID Datei des Daemon die für das beenden des Prozesses verwendet wird

Einträge in slapd.conf

- loglevel 256
  - Normalerweise loggt unser slapd über syslogd nach /var/log/messages. Optional kann über loglevel und logfile das Verhalten angepasst werden
  - http://www.openldap.org/doc/admin24/slapdconfig.html

Alle rechts genannten Level können miteinander kombiniert werden

32+128=160 würde alle Suchfilter und alle ACL-Zugriffe loggen

Level	Keyword	Description
-1	any	enable all debugging
0		no debugging
1	(0x1 trace)	trace function calls
2	(0x2 packets)	debug packet handling
4	(0x4 args)	heavy trace debugging
8	(0x8 conns)	connection management
16	(0x10 BER)	print out packets sent and received
32	(0x20 filter)	search filter processing
64	(0x40 config)	configuration processing
128	(0x80 ACL)	access control list processing
256	(0x100 stats)	stats log connections/operations/results
512	(0x200 stats2)	stats log entries sent
1024	(0x400 shell)	print communication with shell backends
2048	(0x800 parse)	print entry parsing debugging
16384	(0x4000 sync)	syncrepl consumer processing
32768	(0x8000 none)	only messages that get logged whatever log level is set

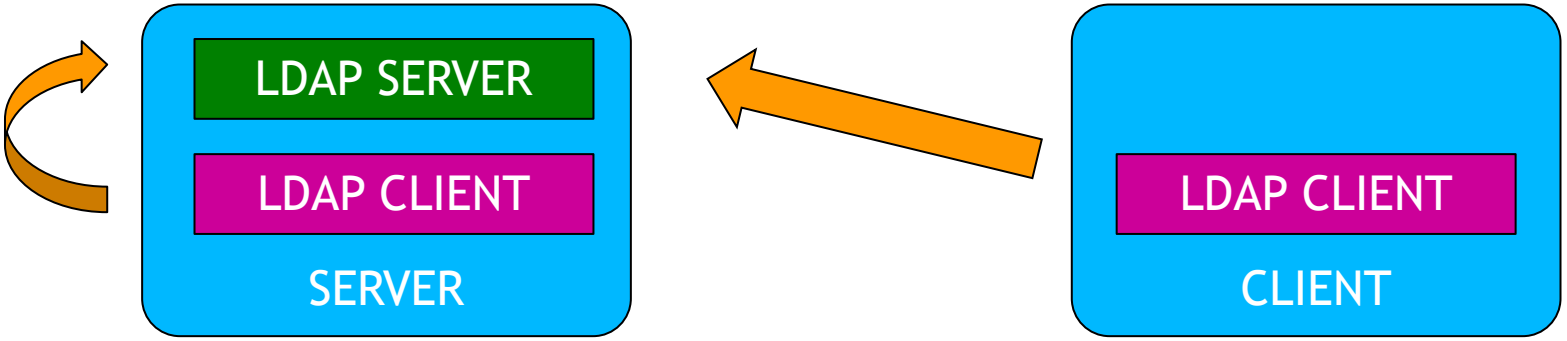
Einträge in slapd.conf

- sizelimit 500
  - regelt über einen Integer Wert die maximale Anzahl der gefundenen Objekte die bei einem Suchmuster zurück geliefert werden. Die Standardeinstellung ist unlimited
- index objectClass eq
  - bestimmt welche Indexes der Server beim erstmaligen laden des Servers erstellen soll. Spätere Änderungen können nur noch durch das Tool slapindex erstellt werden
  - Auch wenn Attribute nicht indexiert sind kann nach Ihnen gesucht werden jedoch zum Nachteil der Performance
  - http://www.zytrax.com/books/ldap/apa/indeces.html

LDAP Client Konfiguration

LDAP Client

- Authentifizierung am LDAP Server
  - Obwohl ein LDAP Server auf dem System läuft wird dieser noch nicht für die Verwaltung der lokalen Accounts verwendet
  - Um einen LDAP Server (lokal oder remote) als Authentifizierungsserver verwenden zu können muss zuerst der LDAP Client (ldap.conf) dementsprechend angepasst werden



/etc/(open)ldap/ldap.conf

- ldap.conf
  - dient zur Konfiguration der systemweiten Client Einstellungen für alle Produkte die auf den libldap*.so* Bibliotheken basieren
    - /etc/openldap/ldap.conf ...SUSE, Red Hat
    - /etc/ldap/ldap.conf ...DEBIAN

- ~/.ldaprc
  - Anwenderspezifische Konfigurationsdatei für Clients

/etc/(open)ldap/ldap.conf

- BASE dc=local,dc=site
  - definiert die Searchbase des Clients d.h den Punkt im DIT ab dem der Client suchen darf
  - Die Searchbase kann auch auf einen tieferen Punkt im DIT zeigen solange es keine Referral gibt
  - Existiert dieser Eintrag nicht, muss bei den LDAP-Tools die Searchbase via -b explizit angegeben werden

```
# /etc/openldap/ldap.conf
BASE  dc=local,dc=site
URI ldap://ldapmaster.local.site
```

/etc/(open)ldap/ldap.conf

- URI ldap://ldapmaster.local.site:389
  - definiert an welchen Host der Client seine Anfrage über das LDAP Protokoll schicken soll
  - Optional kann auch noch der Port angegeben werden
  - Mehrere Einträge als Fallback Lösung sind möglich
  - URI ldap://ldapmaster.local.site ldap://ldapslave.local.site

```
# /etc/openldap/ldap.conf
BASE dc=local, dc=site
URI ldap://ldapmaster.local.site
```

Aufgabe

- Konfiguration des LDAP Clients auf Rechner ldapmaster

```
# /etc/openldap/ldap.conf
BASE dc=local,dc=site
URI ldap://ldapmaster.local.site
```

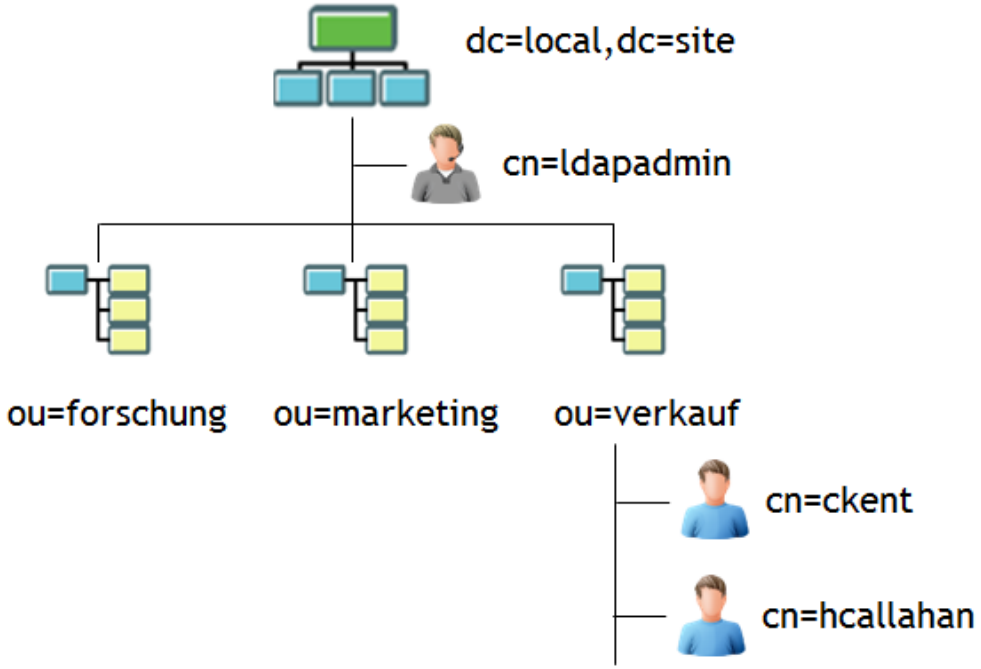
```
# /etc/hosts
127.0.0.2   ldapmaster    ldapmaster.local.site
```

```
# /etc/HOSTNAME
ldapmaster.local.site
```

Erstellen der Basis Struktur

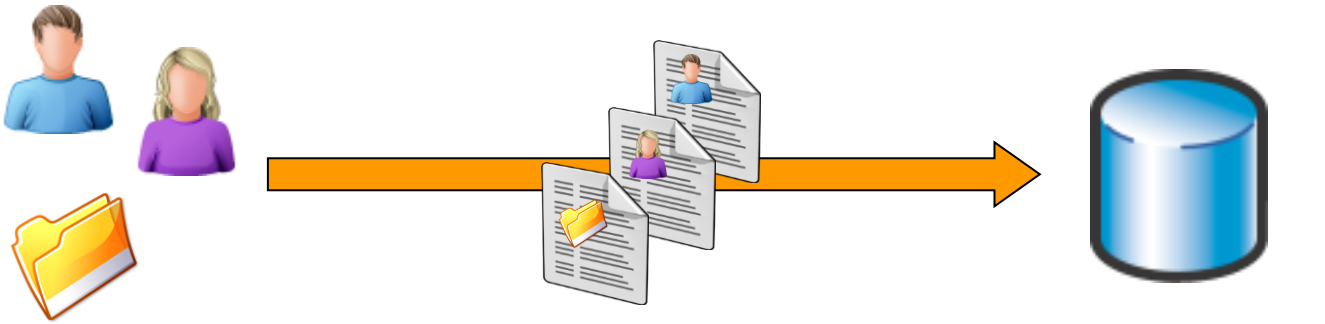
Basis Struktur

- Erstellen der Basis Struktur
  - unsere Datenbank beinhaltet noch keine Objekte
  - Die einzelnen Objekte müssen manuell erstellt werden



LDIF

- Was ist LDIF?
  - LDIF ist eine Textdatei im LDAP Data Interchange Format
    - RFC 2849 - The LDAP Data Interchange Format (LDIF)
  - LDIF wird dazu genutzt um Daten in einem LDAP fähigen Verzeichnisdienst zu importieren, exportieren oder zu verändern.



LDIF Datei

- Eine LDIF Datei muss mindestens beinhalten
  - Einen Distinguished Name (dn) des Objektes

    dn: cn=Peter,ou=verkauf,dc=local,dc=site
  - Eine oder mehrere Objektklassen des neuen Objektes

    objectClass: organization

    objectClass: inetOrgPerson
  - Einen oder mehrere Attribute

    givenName: Peter

    sn: Jahn

LDIF Datei für Initial Setup

```
# /root/ldif/base
#Basis dc=local,dc=site
dn: dc=local,dc=site
objectClass: dcObject
objectClass: Organization
dc: local
o: Brainstorm

# User ldapadmin
dn: cn=ldapadmin,dc=local,dc=site
objectClass: organizationalRole
cn: ldapadmin
```

Erstellt die Basis Struktur

Erstellt passend zum rootDN ein echtes LDAP Objekt

Syntax

- Syntax LDIF Datei
  - Zuerst kommt der Distinguished Name (DN) der immer genau 1 Objekt innerhalb des DIT beschreibt
  - Danach kommen die benötigten Objektklassen
  - Danach die Attribute
  - Zwischen 2 Objekten muss immer mindestens eine Leerzeile vorhanden sein
  - ein Leerzeichen am Anfang der Zeile bedeutet das diese Zeile zur vorherigen Zeile gehört (Fortsetzungszeile)

cn vs. uid

- dn: cn=Peter Jahn,ou=verkauf,dc=local,dc=site
  - Das Attribut commonName(cn) entspricht dem vollständigen Namen des Benutzers
  - Kann Leerzeichen und Umlaute beinhalten (Problematisch!)

- dn: uid=pjahn,ou=verkauf,dc=local,dc=site
  - Das Attribut uid entspricht dem UNIX Account Namen
  - sollte klein geschrieben werden, erlaubt keine Abstände, Umlaute oder Sonderzeichen (Ideal :-)

LDAP spezifische Werkzeuge

- LDAP Kommandozeilenprogrammen
  - Das openldap2-client Paket liefert auch einige LDAP spezifische Tools für die Befehlszeile

Tool	Aufgabe
ldapsearch	Suchen/Exportieren von Objekten
ldapadd	Hinzufügen von Objekten. Entspricht ldapmodify -a
ldapmodify	Ändern vorhandener Objekte
ldapdelete	Löschen vorhandener Objekte
ldappasswd	Passwort für ein Objekt setzten
ldapexop	Extended Operations (ManageDIT,...)
ldapcompare	Prüft ob der Datensatz existiert (Rückgabe: Boolean)
ldapwhoami	meine aktuelle LDAP Identität
ldapmodrdn	Modifizieren des "realtiven dn" rdn

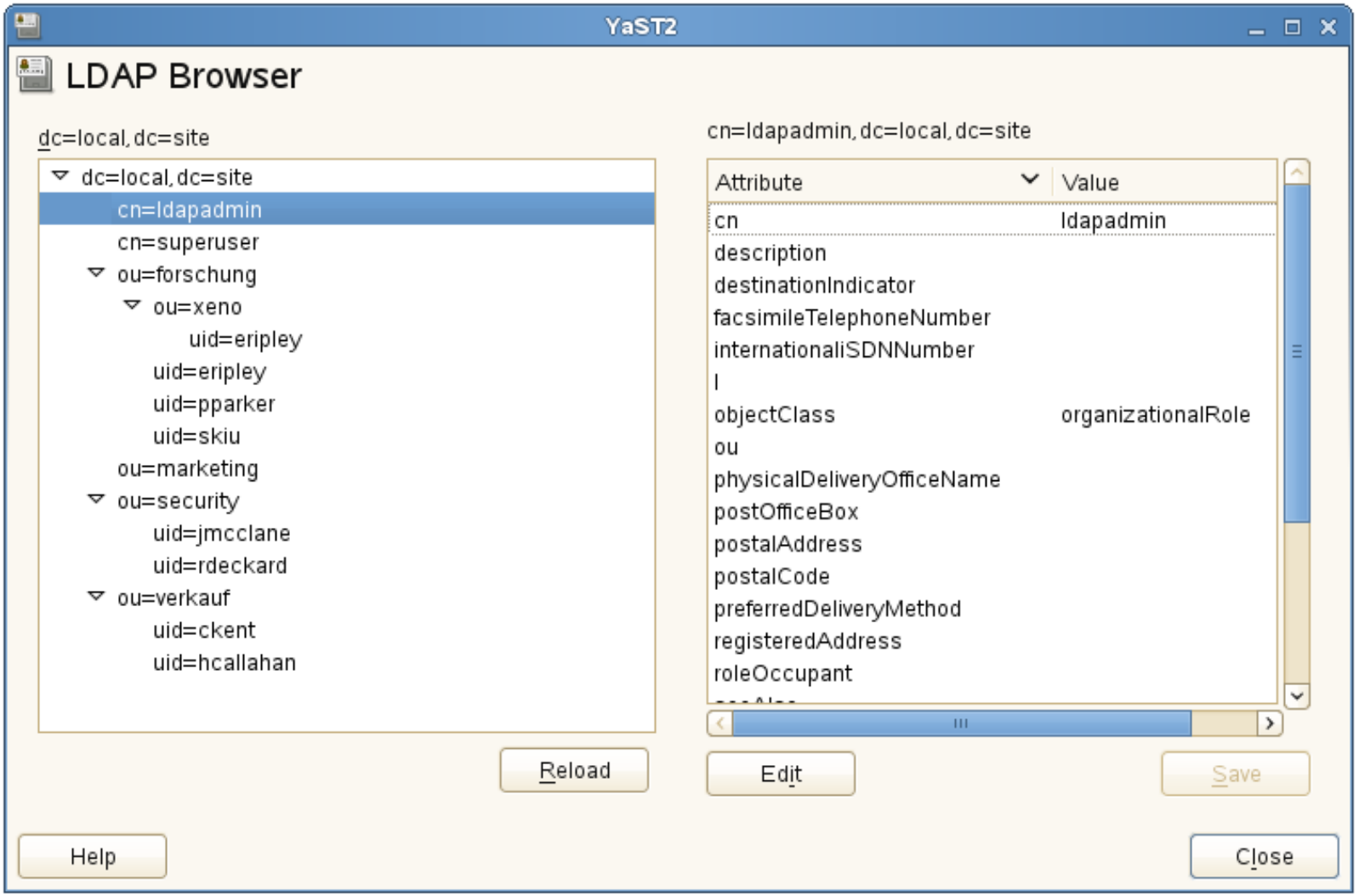
Aufgabe: Erstellen der Basisstruktur

```
ldapadd -x -W -D cn=ldapadmin,dc=local,dc=site  -f struktur.ldif
```

Option	Bedeutung
-x	einfache unverschlüsselte Authentifizierung (simple bind)
-W	anschließend ist eine Passworteingabe erforderlich
-D	Der Benutzer mit dem wir auf das DIT zugreifen wollen
-f	die zu verwendende LDIF Datei

```
# Importieren der Basisstruktur
peter@tux~> ldapadd -xWD cn=ldapadmin,dc=local,dc=site -f struktur.ldif
Enter LDAP Password:
adding new entry "dc=local,dc=site"
adding new entry "cn=ldapadmin,dc=local,dc=site"
```

Struktur im Yast LDAP Browser



Suchen nach einem Objekt

ldapsearch -x "(objectClass=*)"

- liefert uns alle im DIT befindlichen Objekte
  - ldapsearch -x würde das selbe Ergebnis liefern

```
# extended LDIF
#
# LDAPv3
# base <dc=local,dc=site> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# local.site
dn: dc=local,dc=site
objectClass: dcObject
objectClass: organization
dc: local
o: Brainstorm
```

Search Beispiele

- ldapsearch -x -LLL
  - eine bereinigte Ausgabe ohne Kommentare
    - -L ...Ausgabe im LDIF Format
    - -LL ...Deaktivieren von Kommentaren
    - -LLL ...Deaktivieren der LDIF Versionnummer

```
sles11sp1:~/ldif # ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=local,dc=site> (default) with
# filter: (objectclass=*)
# requesting: ALL
#

# local.site
dn: dc=local,dc=site
objectClass: dcObject
objectClass: organization
dc: local
o: Brainstorm
```

```
sles11sp1:~/ldif # ldapsearch -x -LLL
dn: dc=local,dc=site
objectClass: dcObject
objectClass: organization
dc: local
o: Brainstorm
```

Search Beispiele

- ldapsearch -x -LLL uid=ckent sn mail title
  - Alle User mit der UID ckent ausgeben
    - Aber nur die Attribute sn, mail und title anzeigen

```
dn: uid=ckent,ou=verkauf,dc=local,dc=site
sn: Kent
mail: ckent@verkauf.local.site
title: admin
```

Search Beispiele

- ldapsearch -x -LLL "sn=*riple*" mail telephoneNumber
  - alle User die im Nachnamen (sn) den String "riple" haben,
    - aber nur die Attribute mail und telephoneNumber
  - Wildcards müssen in Anführungszeichen oder Klammern stehen

```
dn: uid=eripley,ou=forschung,dc=local,dc=site
mail: eripley@forschung.local.site
telephoneNumber: 0800-forschung-905

dn: uid=eripley,ou=xeno,ou=forschung,dc=local,dc=site
mail: eripley@xeno.forschung.local.site
telephoneNumber: 0800-xenoforschung-907

dn: uid=eripley,ou=marketing,dc=local,dc=site
mail: eripley@marketing.local.site
telephoneNumber: 0800-marketing-909
```

ldapmodify ohne LDIF

- ldapmodify
  - kann von der Befehlszeile ausgeführt werden oder bequemer via LDIF Datei (-f file.ldif)

```
# Arbeiten ohne LDIF Datei
peter@tux~> ldapmodify -xWD cn=ldapadmin,dc=local,dc=site
Enter LDAP Password:
dn: uid=ckent,ou=verkauf,dc=local,dc=site
changetype: modify
replace: description
description: Superman, who else
<STRG> + <D>
modifying entry "uid=ckent,ou=verkauf,dc=local,dc=site"
```

Aufgabe

- Erstellen Sie eine LDIF Datei kent2.ldif

```
dn: uid=ckent,ou=verkauf,dc=local,dc=site
changetype: modify
add: title
title: admin
```

- Importieren sie die Datei
  - ldapmodify -xWD cn=ldapadmin,dc=local,dc=site -f kent2.ldif

- Überprüfen der Änderung
  - ldapsearch -x -LLL uid=ckent

ldapdelete

- ldapdelete
  - dient dazu um einzelne Objekte aus dem DIT zu löschen
  - Es gibt auch die Option -r (rekursiv) die alle Objekte im Subtree löscht -> Also Vorsicht!
  - Vor dem löschen ist es immer praktisch mit ldapsearch sich zur Sicherheit ein Backup des Objektes zu erstellen

```
ldapsearch -xWD cn=ldapadmin,dc=local,dc=site \
    uid=ckent -LLL > user.ldif        ...PW nicht vergessen
ldapdelete -xWD cn=ldapadmin,dc=local,dc=site  \
    uid=ckent,ou=verkauf,dc=local,dc=site
```

Suchfilter

- ldapsearch und Suchfilter
  - Die Verwendung von Suchfiltern ermöglicht es uns selektiv Einträge aus dem DIT zu lesen
    - Objekte in bestimmten Bereichen
    - Objekte mit speziellen Attributen
    - Objekte mit speziellen DN
    - und vieles mehr

Bereich einschränken

- ldapsearch -s <scope>
  - Der Scope definiert die Suchtiefe innerhalb des DIT

Scope	Bereich
base	Nur innerhalb der BaseDN (local.site) suchen
one	Nur eine Ebene unterhalb von BaseDN suchen
sub	Sucht im kompletten DIT (Standardeinstellung)

- ldapsearch -s <scope> -b <searchbase>
  - definiert eine neue Searchbase ab der die Suche starten soll

Beispiele

- ldapsearch -x -LLL cn="Ellen Ripley*" -s base
  - liefert kein Ergebnis weil es das Objekt zwar mehrmals gibt aber nicht in der BaseDN

- ldapsearch -x -b ou=forschung,dc=local,dc=site \
  -LLL cn="Ellen Ripley*" -s one description
  - liefert einen Eintrag zurück

- ldapsearch -x -b ou=forschung,dc=local,dc=site \
  -LLL cn="Ellen Ripley*" -s sub description
  - liefert zwei Einträge zurück, sub ist die Standardeinstellung

RFC 1960 - LDAP Search Filters

3. String Search Filter Definition

The string representation of an LDAP search filter is defined by the following grammar. It uses a prefix format.

```
<filter> ::= '(' <filtercomp> ')'
<filtercomp> ::= <and> | <or> | <not> | <item>
<and> ::= '&' <filterlist>
<or> ::= '|' <filterlist>
<not> ::= '!' <filter>
<filterlist> ::= <filter> | <filter> <filterlist>
<item> ::= <simple> | <present> | <substring>
<simple> ::= <attr> <filtertype> <value>
<filtertype> ::= <equal> | <approx> | <greater> | <less>
<equal> ::= '='
<approx> ::= '~='
<greater> ::= '>='
<less> ::= '<='
<present> ::= <attr> '=*'
<substring> ::= <attr> '=' <initial> <any> <final>
<initial> ::= NULL | <value>
<any> ::= '*' <starval>
<starval> ::= NULL | <value> '*' <starval>
<final> ::= NULL | <value>
```

<attr> is a string representing an AttributeType, and has the format defined in [1]. <value> is a string representing an AttributeValue, or part of one, and has the form defined in [2]. If a <value> must contain one of the characters '*' or '(' or ')', these characters should be escaped by preceding them with the backslash '\' character.

Filterverknüpfungen

Filter	Bedeutung
'(& (filter1) (filter2) ...)'	UND Verknüpfung
'(| (filter1) (filter2) ...)'	ODER Verknüpfung
'(! (filter1))'	Filter Negierung (nur ein Filter ist erlaubt)

- Filterverknüpfungen
  - Filter können beliebig oft miteinander verknüpft werden
  - wichtig dabei ist die richtige Klammersetzung
  - Abstände in den Klammern sind nicht notwendig
    - '(& (filter1) (filter2))' oder '(&(filter1)(filter2)'

Passwörter

- slappasswd
  - erstellt offline ein verschlüsseltes Passwort welches über eine LDIF Datei importiert werden kann

- ldappasswd
  - dient dazu um zur Laufzeit Passwörter im DIT zu verändern ohne eine LDIF Datei erstellen zu müssen
    - -s secret ...neues Passwort
    - -S ...Prompt für neues Passwort

Beispiel Passwortänderung

```
# Ändern eines Passwortes
sles11sp1:~# ldappasswd -xWD cn=ldapadmin,dc=local,dc=site -S
uid=ckent,ou=verkauf,dc=local,dc=site
New password:
Re-enter new password:
Enter LDAP Password:
```

- Erklärung
  - -S Prompt für User ckent Passwort
    - New password, Re-enter new password
  - -W Prompt für ldapadmin Passwort
    - Enter LDAP Password

Ändern des RDN

- ldapmodrdn
  - dient dazu um den Relative Distinguished Name eines Objektes zu verändern z.b uid=Peter -> uid=Franz

ldapmodrdn -xWD -r cn=ldapadmin,dc=local,dc=site
uid=skiu,ou=forschung,dc=local,dc=site uid=skiu1

-r löscht den alten Eintrag innerhalb des Objektes

Verschieben eines Users

- ldapmodify ... -f usermove.ldif
  - wollen wir ein User Objekt in eine andere OU verschieben benötigen wir den ldapmodify Befehl mit einer speziellen LDIF Datei

```
# /root/ldif/skiu1_modrdn.ldif
dn: uid=skiu1,ou=forschung,dc=local,dc=site
changetype: moddn
newrdn: uid=skiu1
deleteoldrdn: 1
newSuperior: ou=verkauf,dc=local,dc=site
```

Auszug aus RFC 2849 LDIF

- changetype: [add | delete | modify | modrdn/moddn]
  - bei changetype modify geht zusätzlich:
    - add:, replace:, delete:
- deleteoldrdn
  - 0 ...Eintrag behalten (=copy), 1 ...Eintrag löschen (=move)
- newsuperior
  - der neue DN-Eintrag des Objektes, nur in Verbindung mit moddn/modrdn möglich
- newdrn
  - der neue RDN-Eintrag des Objektes, nur in Verbindung mit moddn/modrdn möglich

Verschieben eines Users

- Verschieben eines Benutzers
  - ldapmodify -xWD cn=ldapadmin,dc=local,dc=site -f skiu1_modrdn.ldif

```
# User Objekt via LDIF Datei im DIT verschieben
peter@tux~> ldapmodify -xWD cn=ldapadmin,dc=local,dc=site
-f skiu1_modrdn.ldif
Enter LDAP Password:
modifying rdn of entry "uid=skiu1,ou=forschung,dc=local,dc=site"
```

LDAP-Tools

slap* Tools

- slap* Tools
  - Zusätzlich zu den normalen ldap* Tools gibt es auch noch einige slap* Tools.

Tool	Aufgabe
slapadd	Importieren der Datenbank
slapcat	Auslesen der Datenbank
slapindex	Indexes neu erstellen
slapacl	Vorabtests von ACLs
slaptest	Syntaktischer Check der slapd.conf
slapdn	Konformitätsprüfung von DNs zum aktuellen Schema
slapauth	Testen der Authentification von DNs

slap* Tools

- slapcat und slapadd
  - Im Gegensatz zu ldap* Tools welche für den Online Modus gedacht sind verwendet man die meisten slap* Tools wenn der LDAP Server gestoppt ist
  - Der große Unterschied ist das die slap* Tools direkt auf die Datenbankdateien zugreifen und darum können sie auch nicht verwendet werden um remote auf das System zuzugreifen
    - Ein Export via slapcat und ein anschließender Import mit ldapadd ist nicht möglich!
  - Typische Anwendungsbeispiele
    - Backup/Restore, Klonen eines DIT, Replikation,...

Backup eines LDAP Servers

- Ein sehr einfaches Backup Script
  - professionelleres Script gibt es unter

http://blog.pregos.info/2010/03/23/ldap-backup-erstellen/

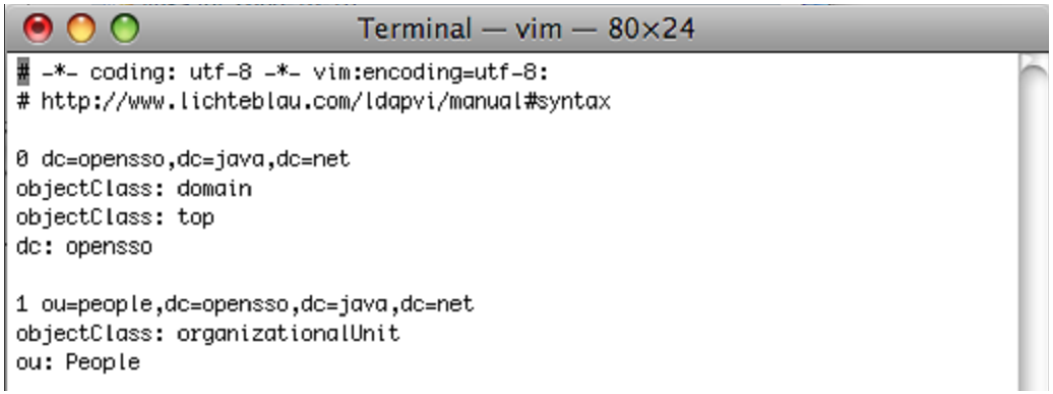
```
# backup.sh
killall slapd &> /dev/null
if  [ $(ps aux | grep slapd | grep -v grep) ]
 then
        echo "Service is still running"; exit 2
 else
        slapcat > /tmp/dbdump_$(date +%F).ldif
fi
/etc/init.d/ldap start
```

Wiederherstellen des Backups

- Einspielen des slapcat Backups
  1. Sicherstellen dass kein SLAPD läuft
  2. löschen aller Datenbankdateien unter /var/lib/ldap/ außer der DB_CONFIG Datei (rm /var/lib/ldap/*.*)
  3. slapadd -l /tmp/dbdump_<datum>.ldif
  4. Überpüfen der Besitzer und Gruppe auf den neuen Dateien
  5. /etc/init.d/ldap start

weitere LDAP Tools

- ldapvi
  - lässt sich ohne Parameter aufrufen wenn die ldap.conf richtig konfiguriert wurde und erlaubt das direkte editieren des DIT mit dem VIM
    - Wird auf Debian standardmäßig mitgeliefert und andere Distributionen können das Tool nachträglich installieren
    - http://www.lichteblau.com/ldapvi/

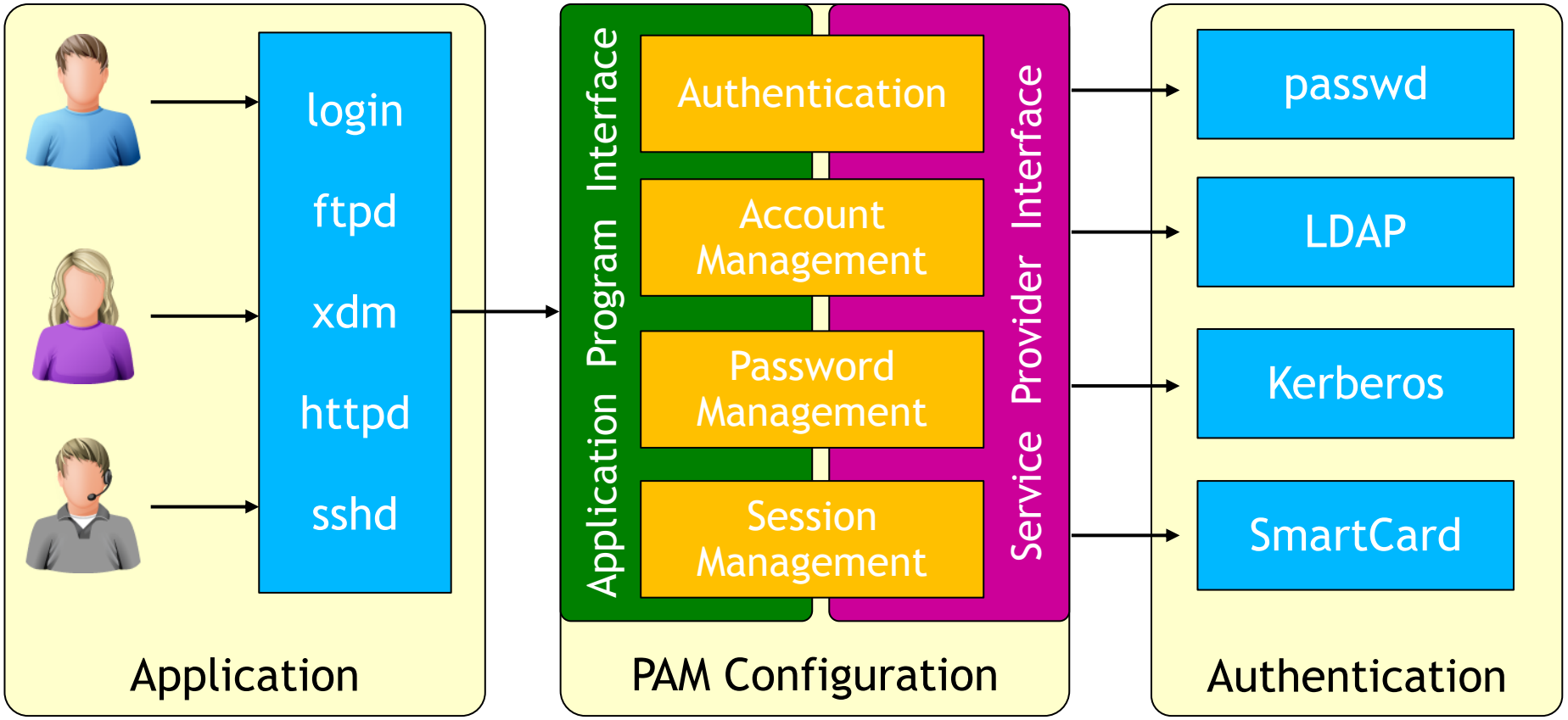


Linux Benutzer Authentifizierung

Linux Benutzer Authentifizierung

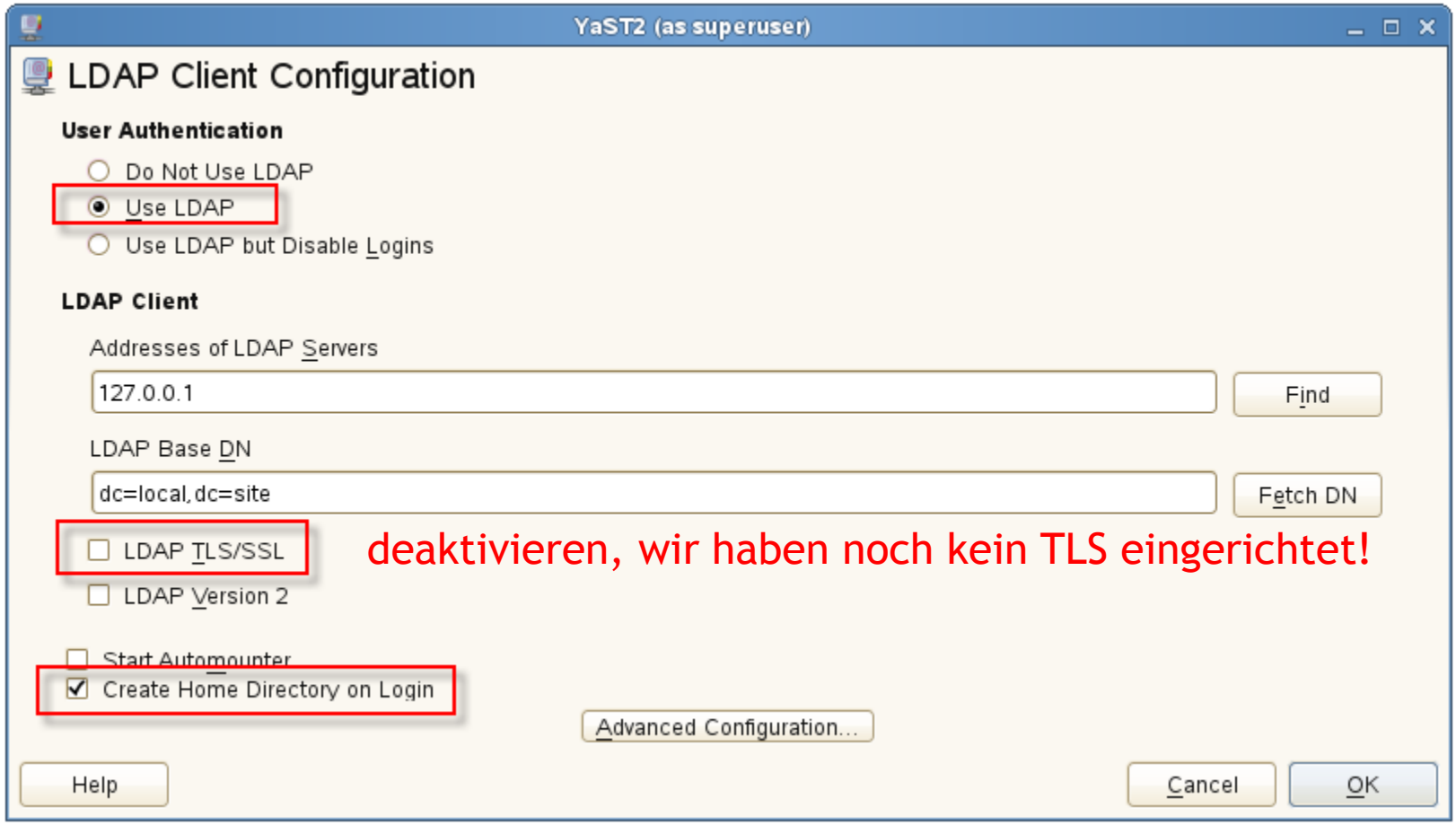
- Aktuelle Situation
  - Wenn sich ein Benutzer am Linux System via GUI, SSH, etc. anmeldet dann wird nur in den lokalen Dateien wie /etc/{passwd, shadow, group} gesucht
- Gewünschte Situation
  - Zusätzlich zu den lokalen Dateien soll auch geprüft werden ob es den Benutzer nicht auch im DIT gibt.
    - -> zentrale Benutzerdatenbank für alle Systeme

Pluggable Authentication Modules



Konfiguration des LDAP Clients - SUSE

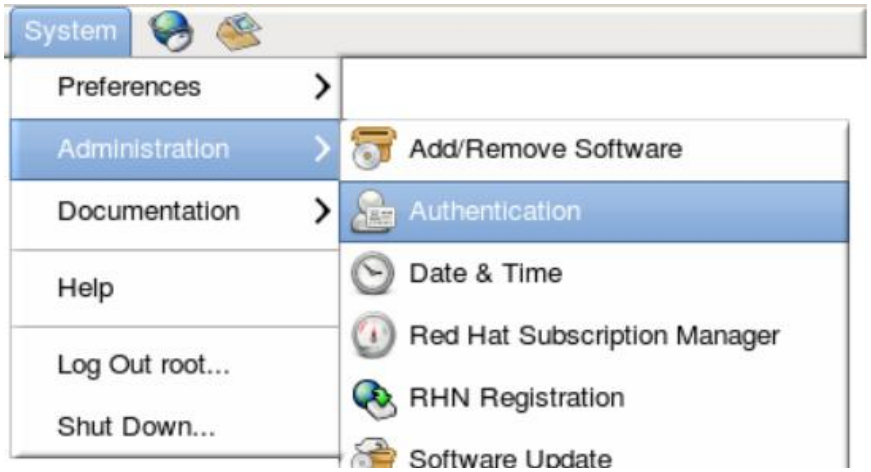
- Im Yast, Network Services > LDAP Client



www.LinuxCampus.net

Konfiguration des LDAP Clients - Red Hat

- system-config-authentication



www.LinuxCampus.net

LDAP Authentifikation

- libnss_ldap.so* und pam_ldap.so*
  - sind die Bibliotheken welche vom Linux System verwendet werden um die LDAP Authentication am lokalen Linux System durchzuführen
- SUSE
  - die Konfiguration wird zentral über /etc/ldap.conf geregelt
- DEBIAN
  - /etc/libnss-ldap.conf und /etc/pam_ldap.conf
    - http://wiki.debian.org/LDAP/PAM

Software Installation

- nss_ldap
  - Erlaubt ein LDAP Directory für Quellen wie User, Passwörter, Services ,Alias,...

- pam_ldap
  - ermöglicht das Ändern von Passwörtern im DIT

Name Service Caching Daemon

- Name Service Caching Daemon
  - Der nscd dient als Cache für die in /etc/nsswitch.conf definierten Services
  - Bei Problemen /etc/init.d/nscd neustarten
  - Welche Informationen im Cache wielange gehalten werden sollen wird in der Datei /etc/nscd.conf bestimmt

```
enable-cache            passwd          yes
positive-time-to-live   passwd          600
negative-time-to-live   passwd          20
suggested-size          passwd          211
check-files             passwd          yes
persistent              passwd          yes
shared                  passwd          yes
max-db-size             passwd          33554432
auto-propagate          passwd          yes
```

Benutzer anzeigen

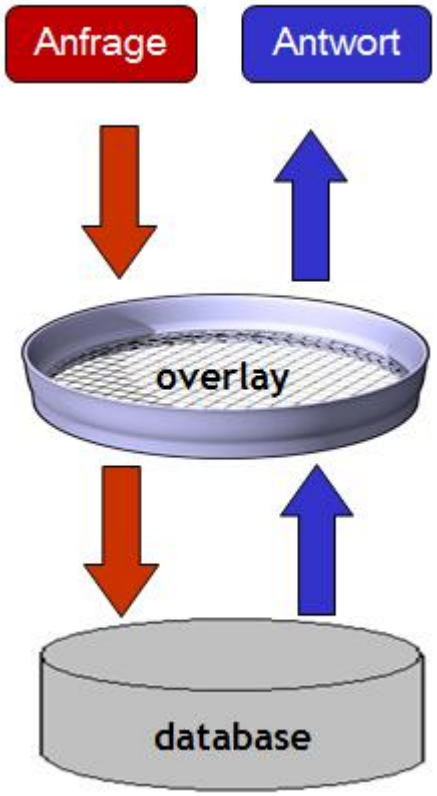
- Welche Benutzer kennt das System
  - getent passwd ...passwd, ldap, ...

```
# Alle dem System bekannten User anzeigen
peter@tux~> getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
```

Overlays in OpenLDAP

Überblick Overlays

- Aufgabe von Overlays
  - Overlays bieten die Möglichkeit das Verhalten der eingesetzten Datenbank zu modifizieren ohne die Datenbank bzw. das Datenbank Backend selbst zu verändern
  - Die Overlays werden dem eigentlichen Datenbank Backend modular vorgeschalten.
  - Anfragen werden von Overlays entgegen genommen, modifizieren sie und leiten sie an das Backend weiter. Auf umgekehrten Weg passiert das Gleiche



Auszug aus den verfügbaren Overlays

Overlay	Aufgabe
accesslog	Auditing für DB-Zugriffe, z.b für delta-syncrepl
auditlog	einfache Variante von accesslog, loggt in Klartextdatei
chain	automatische, serverseitige Verfolgung von Referals
collect	trägt kollektive Attribute (RFC3671) zusammen
constraint	Attributwert mit regulären Ausdrücken eingrenzen
dds	zeitlich limitierte Objekte erstellen
dynlist	dynamische Gruppen über Filterkriterien erstellen
memberof	Reverse Group Membership
pcache	Proxy Caching für ldapsearch Requests
refinit	Sicherstellung der referentiellen Integrität im DIT
syncprov	stellt die LDAP Sync Replikation zur Verfügung
unique	Sicherstellung der Einzigartigkeit von Attributen
valsort	zurückgegebene Werte sortieren

Overlay dynlist

Gruppen

```
# struktur.ldif
dn: cn=superuser,dc=local,dc=site
objectClass: top
objectClass: groupOfNames
cn: superuser
member: uid=ckent,ou=verkauf,dc=local,dc=site
member: uid=skiu,ou=forschung,dc=local,dc=site
member: uid=hcallahan,ou=verkauf,dc=local,dc=site
```

- member
  - Die Gruppenzugehörigkeit muss manuell durch den Admin verwaltet werden

Overlay dynlist

- Overlay dynlist (slapo-dynlist(5))
  - Die Gruppenzugehörigkeit muss normalerweise manuell durch den Admin verwaltet werden
  - dynlist ermöglicht es uns basierend auf gewissen Filterkriterien die Mitglieder dynamisch zu einer Gruppe hinzuzufügen
    - admingroup: Alle Mitglieder die Admin im Titel haben
  - Die Filterkriterien können während der Laufzeit modifiziert werden

Aufgabe

- Erweitern der Konfigurationsdatei
  - mit dem Schema dyngroup
  - mit den Zeilen overlay und dynlist-attrset am Ende der Datei
  - danach Dienst starten

```
# /etc/openldap/slapd.conf
...
include      /etc/openldap/schema/nis.schema
include      /etc/openldap/schema/dyngroup.schema
...
index        objectClass    eq
overlay  dynlist
dynlist-attrset   groupOfURLs  memberURL
```

Erklärung

- include /etc/openldap/schema/dyngroup.schema
  - Dieses Schema liefert uns neue Objektklasse groupOfURLs und Attribute wie memberURL
- overlay dynlist
  - aktiviert die Verwendung von dynlist
- dynlist-attrset groupOfURLs memberURL
  - groupOfURLs bestimmt die Objektklasse unter der die gefunden Objekte als Gruppe zusammen gefasst werden
  - memberURL ist das Kriterium welches verwendet wird um die Objekte dynamisch zu kollektieren
    - ldap:///baseDN[??[searchScope][?searchFilter]]

Erstellen der neuen Gruppe: dyngroup

```
# dyngroup.ldif
dn: cn=dyngroup,dc=local,dc=site
objectClass: top
objectClass: groupOfURLs
cn: dyngroup
memberURL: ldap:///dc=local,dc=site?sn?sub?(title=admin)
```

ldapadd -xWD cn=ldapadmin,dc=local,dc=site -f dyngroup.ldif

- Erklärung
  - Die memberURL sorgt dafür, dass alle bestehenden oder neuen Mitarbeiter im gesamten DIT die im Titel admin stehen haben mit ihrem Attribut sn in der Gruppe cn=dyngroup zusammengefasst werden

Tipp

- memberURL
  - hier können alle Filterverknüpfungen gemäß RFC 2255 verwendet werden

ldap:///dc=local,dc=site?sn?sub?(title=admin)

ldap:///dc=local,dc=site?sn?sub?(&(title=admin)(mail=*verkauf*)))

Title admin bei einem User setzen

```
# titleadmin.ldif
dn: uid=hcallahan,ou=verkauf,dc=local,dc=site
changetype: modify
add: title
title: admin
```

- ldapadd -x -W -D cn=ldapadmin,dc=local,dc=site -f titleadmin.ldif
- ldapsearch -x -LLL cn=dyngroup

```
dn: cn=dyngroup,dc=local,dc=site
objectClass: groupOfURLs
objectClass: top
cn: dyngroup
memberURL: ldap:///dc=local,dc=site?sn?sub?(title=admin)
sn: Callahan
sn: Kent
```

Aufgabe

- Erweitern der Konfigurationsdatei
  - den Eintrag member ergänzen und danach den Dienst neu starten

```
# /etc/openldap/slapd.conf
...
index       objectClass    eq
overlay dynlist
dynlist-attrset  groupOfURLs  memberURL  member
```

```
sles11sp1:~/ldif # ldapsearch -x -LLL  cn=dyngroup
dn: cn=dyngroup,dc=local,dc=site
objectClass: groupOfURLs
objectClass: top
cn: dyngroup
memberURL: ldap:///dc=local,dc=site?sn?sub?(title=admin)
member: uid=hcallahan,ou=verkauf,dc=local,dc=site
member: uid=ckent,ou=verkauf,dc=local,dc=site
```

Erklärung

- member
  - wenn als dritter Parameter member gesetzt wird werden die gefundenen Parameter nicht aufgrund ihres Suchattributes (cn) aufgelistet sondern mit ihrem kompletten DN
    - Ein korrekter DN ist Voraussetzung für ACLs!
- Achtung
  - Auch wenn unser member URL die richtigen Ergebnisse liefert muss er angepasst werden damit ACLs funktionieren

falsch: ldap:///dc=local,dc=site?sn?sub?(&title=admin)

richtig: ldap:///dc=local,dc=site??sub?(&title=admin)

Overlay accesslog

Überblick

- Aufgabe von Overlay accesslog
  - Dieses Overlay bietet uns die Möglichkeit jegliche Zugriffe (read, write, session) auf unser DIT in einer eigenen Datenbank zu loggen
  - Im Gegensatz zu den normalen Debuglogs kann auf diese Weise genau eingestellt werden was protokolliert wird

Aufgabe

```
# /etc/openldap/slapd.conf
#### Beginn BDB database definitions ####
database    hdb
suffix      cn=logs
rootdn      cn=logs
rootpw      {SSHA}4PvZLcpQ7s1CyQG+yworyl5DcrFTn78q
directory   /var/lib/ldaplogs
index       reqStart, reqEnd, reqMod, reqResult eq
... database hdb definition...
index       objectClass    eq
overlay     accesslog
logdb       cn=logs
logops      writes session
logpurge    08:00 00:30
```

Erklärung

- overlay accesslog
  - aktiviert die Overlay Direktive
- logops writes session
  - Mit writes werden sämtliche add/modify/delete Aktionen abgedeckt
  - Mit session sämtliche Anmelde und Abmeldevorgänge
- logpurge 08:00 00:30
  - die Kriterien wann alte Logeinträge wieder gelöscht werden
  - alle 30 Minuten <interval> auf Log Einträge prüfen die älter als 8 Stunden <age> sind und diese löschen

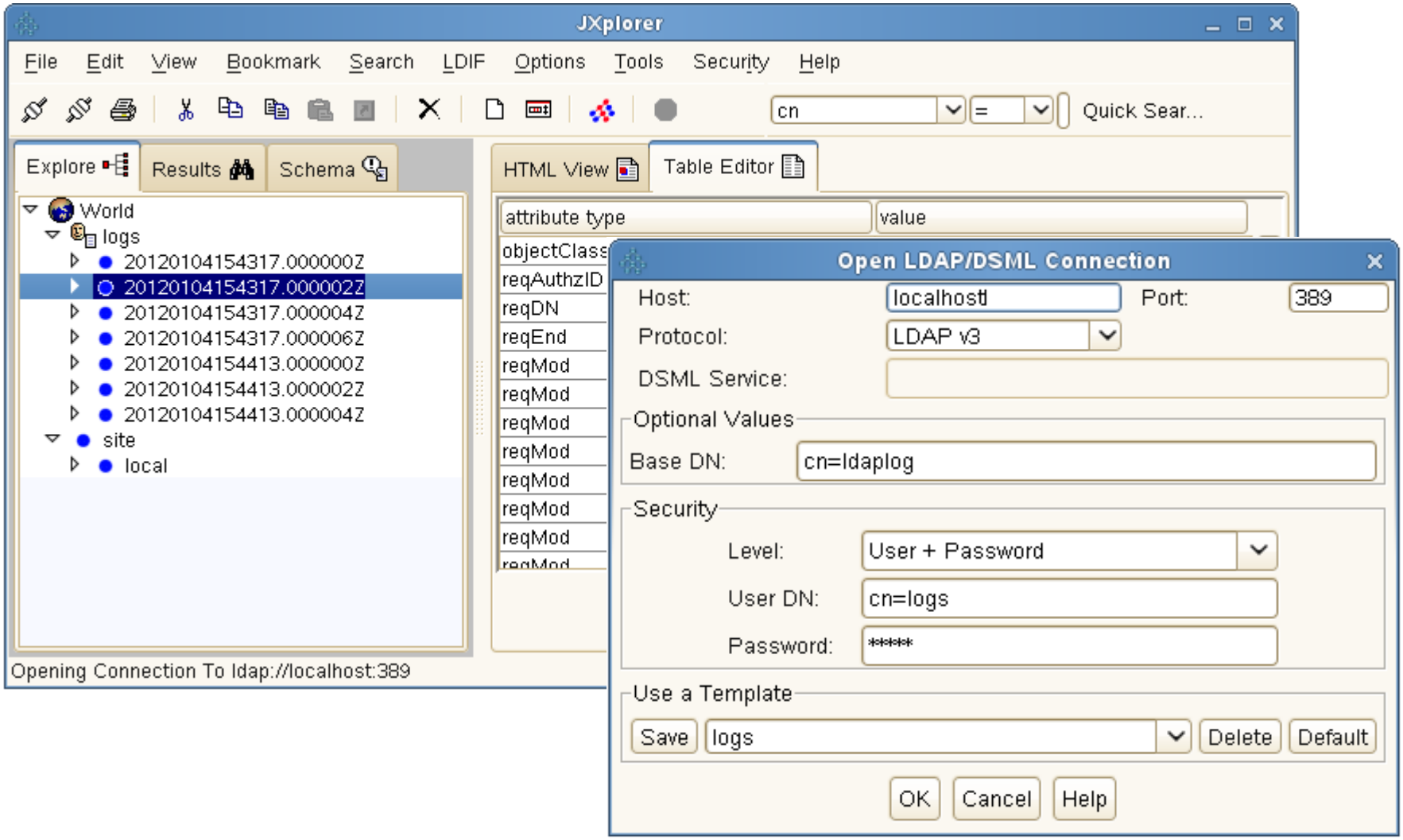
Aufgabe 2

- Erstellen des definierten Datenbank Verzeichnisses
  - mkdir /var/lib/ldaplogs
  - chown ldap.ldap /var/lib/ldaplogs/
- Starten des Daemon
  - /etc/init.d/ldap start

Testen der Log Funktion

- Produzieren von Logeinträgen
  - ldapadd -x -W -D cn=ldapadmin,dc=local,dc=site -f traffic.ldif
- Überprüfen der access Datenbank
  - ldapsearch -x -b cn=logs

Logs im JXplorer




LinuxCampus.net
trainings from the experts