Directories und LDAP

Michael Behrendt
mb@michael-behrendt.net
http://www.michael-behrendt.net

 Kongress 23. November 2003

Zusammenfassung

Directories und LDAP am Beispiel eines Adressbuchs, das mittels OpenLDAP, Perl und Java zwischen Handy, PDA, WWW und Notebook synchronisiert wird.

Überblick

- Verzeichnis Dienste bzw. Directory Services - Was ist das ?
- LDAP Schnittstelle im Detail - Ideen, Konzepte, Probleme
- Anwendung aus der Praxis - Lösungsansatz für ein Adressbuch

Verzeichnis Dienste / Directory Services

Verzeichnis Dienste / Directory Services

- Was sind die typische Merkmale ?
- Was kann man damit machen ?
- Was ist es nicht ?

Verzeichnis Dienste - Was sind die typische Merkmale für Anwender?

- Vergleich mit der realen Welt: Telefonbücher, Fernsehprogramm, Versandhaus Kataloge
- Verzeichnisse sind dynamisch
  - Inhalt ist stehts aktuell
  - Keine Fehlerquellen beim Verteilen von Informationen
  - Pflege ist sehr einfach möglich - Motivation für gute Pflege
- Verzeichnisse sind anpassbar
  - Nahe zu beliebige Informationstypen speicherbar
  - Beliebige Wege zum Ablegen und zur Suche der Information
- Verzeichnisse können sicher sein
  - Zugriffsrechte
  - Authentifizierung der Anwender

Verzeichnis Dienste - Was sind die typische Merkmale für Techniker?

- Spezialisierte Datenbank
- Weit mehr Lese als Schreibzugriffe
- Einfache Erweiterbarkeit des Datenmodels (Objekte)
- Daten werden weiter verbreitet (Zugriffe von unterschiedlichsten Clienttypen)
- Replikation (Arbeit mit Kopien des Verzeichnisses)
- Performance (Optimierung auf Lesezugriffe)
- Standards (haben wegen weiterer Verbeitung der Daten mehr Bedeutung)
- Transaktionen und Joins (Gibt es typischer Weise nicht)

Verzeichnis Dienste - Was kann man damit machen?

- Zentrale Verwaltung von “Dingen, Daten, Informationen”
- Abfragemöglichkeiten
  - Searching - Suche TelNr für Hans Müller, Müllerstr. 11, 90489 Nürnberg
  - Browsing - Zeige alle Personen (mit TelNr), die in der Müllerstr. 11, 90489 Nürnberg wohnen

Verzeichnis Dienste - Was ist es nicht?

- Datenbank
  - read/write Verhältnis
  - fehlende Transaktionen
  - fehlende SELECT * FROM Table1 INNER JOIN Table2 WITH A=B
- Dateisystem
  - Dateien können organisiert abgelegt werden (vgl. Verzeichnisbäume des Dateisystems)
  - Dateien / Binärdaten können in Directories gehalten werden (z.B. JPG des Passfotos)
  - Dateien sind jedoch häufig sehr groß - ungeeignet
  - Kein wahlfreier (random access) Zugriff möglich - ungeeignet
- Web/FTP Server
  - Ein Web/FTP Server ist zunächst mal auch ein Dateisystem - siehe dort
  - Bei einem Web Server sind Anwendungen über CGIs und Application Server integriert
- DNS Server
  - Das DNS ist ein klassischer Verzeichnisdienst
  - Warum sollte man es Ändern ? Es funktioniert doch ...

LDAP Schnittstelle im Detail

LDAP Schnittstelle im Detail

- Geschichte
- wichtige RFCs
- die vier wichtigen Modelle
- Tips zum Suchen

LDAP - Geschichte, Teil 1/3

- Am Anfang war
  - X.500 mit DAP (X.500s directory client access protocol)
  - DAP - umfangreich, komplex, schwierig zu Implementieren, X.500/OSI Protokollstack
- Entwicklungsstufe 1 - zwei konkurrierende leichtgewichtige Ansätze
  - DAS (RFC1202 - Directory Assistance Service)
  - DIXIE (RFC1249 - Directory Interface to X.500 Implemented Efficiently)
  - beide mit Tcp/Ip und Gatewayrechner zwischen Tcp/Ip und X.500
  - Gatewayrechner ist Protocol Translator von z.B. DIXIE nach DAP
  - Idee überzeugte, aber nur schlechte Implementierung verfügbar

LDAP - Geschichte, Teil 2/3

- Entwicklungsstufe 2 - LDAP Juli 1993
  - erste breit eingesetzte Version LDAPv2 (RFC1777)
  - bietet alle wichtigen Funktionen von DAP (redundantes und exotisches gestrichen)
  - Attribute sind vorwiegend Text-Strings, keine abstrakten Objekte (Performance, Einfachheit)
  - Nachrichten werden nach X.500 Regeln codiert (vereinfacht Arbeit des Gateways)
  - setzt auf Tcp/Ip anstelle des OSI Stacks auf
- Entwicklungsstufe 3 - LDAPv3 1997
  - Internationalisierung - UTF-8
  - Referrals - Verweise auf andere Directories (vgl. eine WWW-URL)
  - Sicherheit - Stichwörter: SASL, TLS, SSL
  - Erweiterbarkeit - neue Befehle können eingefügt werden
  - Discovery - Datenschemas, Protokollerweiterungen, usw. können über spezielles Directory abgefragt werden

LDAP - Geschichte, Teil 3/3

- Entwicklungsstufe 4 - vom Protokoll zum eigenständigen Produkt
  - Früher: Protokoll zum Zugriff auf ein X.500 Directory
  - Heute: Produkt, X.500 Directory wurde durch eigene Datenbank ersetzt, siehe zum Beispiel:
    - Netscape Directory Server
    - Sun ONE Directory Server
    - Microsoft Active Directory
    - Apple NetInfo Directory
    - IBM Directory Server
    - Novell eDirectory
    - Oracle Internet Directory
    - OpenLDAP slapd (stand-alone LDAP server, siehe http://www.openldap.org)

LDAP - wichtige IETF RFCs

- RFC2251: Lightweight Directory Access Protocol (v3)
- RFC2252: Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
- RFC2253: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
- RFC2254: The String Representation of LDAP Search Filters
- RFC2255: The LDAP URL Format
- RFC2256: A Summary of the X.500(96) User Schema for Use with LDAPv3
- RFC2829: Authentication Methods for LDAP
- RFC2830: Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
- RFC3377: Lightweight Directory Access Protocol (v3): Technical Specification

LDAP - Modell

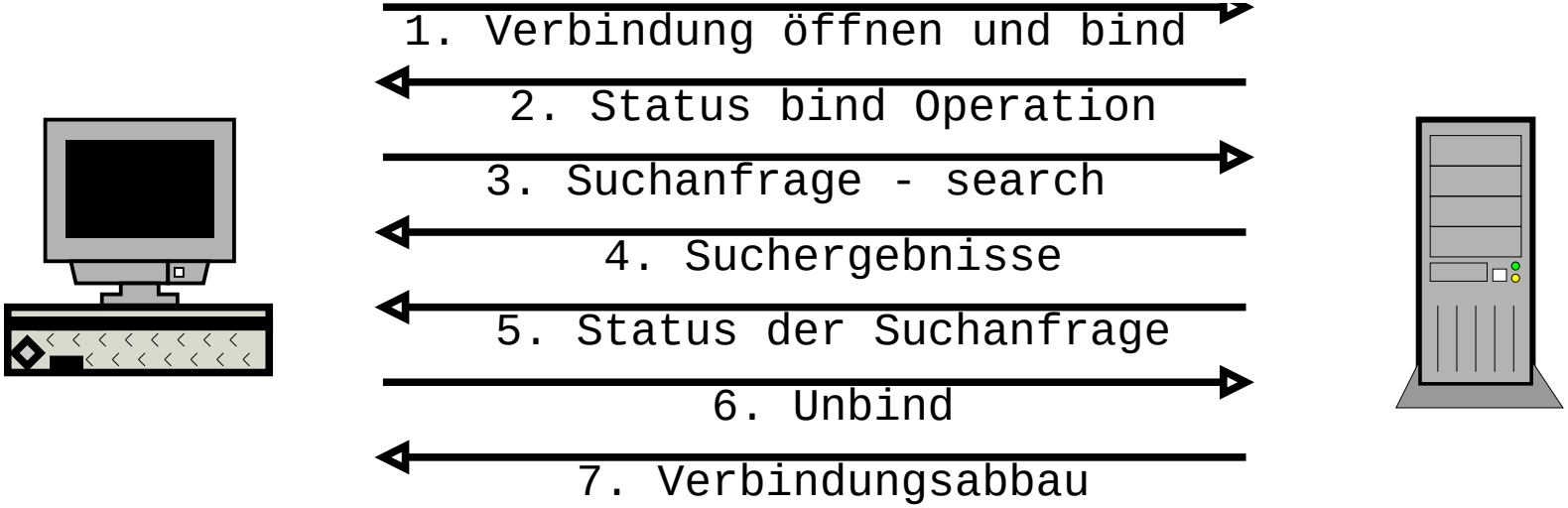
Das LDAP Protokoll besteht aus vier Modellen:

- Funktionsmodell - beschreibt was man alles mit Daten machen kann
- Informationsmodell - beschreibt was in ein Verzeichnis abgelegt werden kann
- Namensmodell - beschreibt wie Daten angeordnet und angesprochen werden
- Sicherheitsmodell - beschreibt wie Verzeichnis geschützt werden kann

LDAP - Funktionsmodell

- Abfrage Operationen
  - search
  - read (Implementiert durch search mit Mächtigkeit der Ergebnissmenge = 1)
  - compare
- Update Operationen
  - add
  - delete
  - modifiy
  - modifiy DN (rename)
- Authentifizierungs & Kontrolloperationen
  - bind - Authentifizieren
  - unbind - Abmelden
  - abandom - Anfrage abbrechen
- LDAPv3 Erweiterungen (exop - extended operations)
  - Beispiel für exop: StartTLS
  - LDAP control - kann Verhalten bestehender Operationen abändern
  - SASL für Authentifizierung

LDAP - Funktionsmodell - typische Session



LDAP - Funktionsmodell - LDAP auf dem Kabel

- LDAP setzt auf LBER auf, nicht auf Text wie http !
- LBER (Lightweight BER) - abgespeckte Variante von BER
- BER (Basic Encoding Rules) - Vorschrift zur Codierung von Datentypen wie Integer, String, usw.
- BER ist systemunabhängig und sehr kompakt
- Spezielles Plug-In für Protokollanalysatoren wie Ethereal nötig

LDAP - Informationsmodell

- Entry
  - Sammlung von Informationen über ein Objekt, häufig identisch mit der realen Welt
  - ein Entry besteht aus:
    - DN (distinguished name) - eindeutige Identifizierung
    - Bündel von Attributen
- Attribute
  - jedes Attribut beschreibt eine besondere Eigenschaft des Objektes
  - ein Attribut besteht aus:
    - ein oder mehrere Values
    - Syntax Definition - wie müssen Values "ausschauen"
    - Regeln für Vergleich von Values (z.B. ignoriere Groß-/Kleinschreibung)
    - Regeln für Sortierungen (z.B. lexigraphisch, nummerisch)
  - Unterscheidung in zwei Klassen
    - user attributes - "normale" Daten durch Nutzer gepflegt
    - operational attributes - werden automatisch gepflegt z.B. modfiyTimeStamp
- Value(s)
  - die eigentlich zu speichernden Daten

LDAP - Informationsmodell - getrennte Datenhaltung

Das Informationsmodell wird in zwei Bereiche geteilt

- Anwendungsdaten
  - die eigentlich zu speichernden Daten mit eindeutiger Identifizierung
  - diese werden zwischen Server und Client ausgetauscht
  - Online: über LDAP
  - Offline: über LDIF ASCII-Text-Schnittstelle mit Import-/Export Tools
- Schemadaten
  - die formelle Beschreibung der möglichen Objekte mit ihren Attributen
  - sind am Server hinterlegt (OpenLDAP: Textfiles geparst beim Serverstart)
  - eingehende Anwendungsdaten werden gegen Schema geprüft
  - Client kann sich bei Server über hinterlegte Schema informieren (teilweise Zukunft)

LDAP - Informationsmodell - Beispiel

Daten im LDIF Text Format

```
dn: cn=admin,dc=example,dc=com
objectClass: top
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: LDAP administrator
userPassword: {SSHA}WzHezyTBKY2d41234567890
```

Zugehöriger Schema-Auszug aus RFC2256

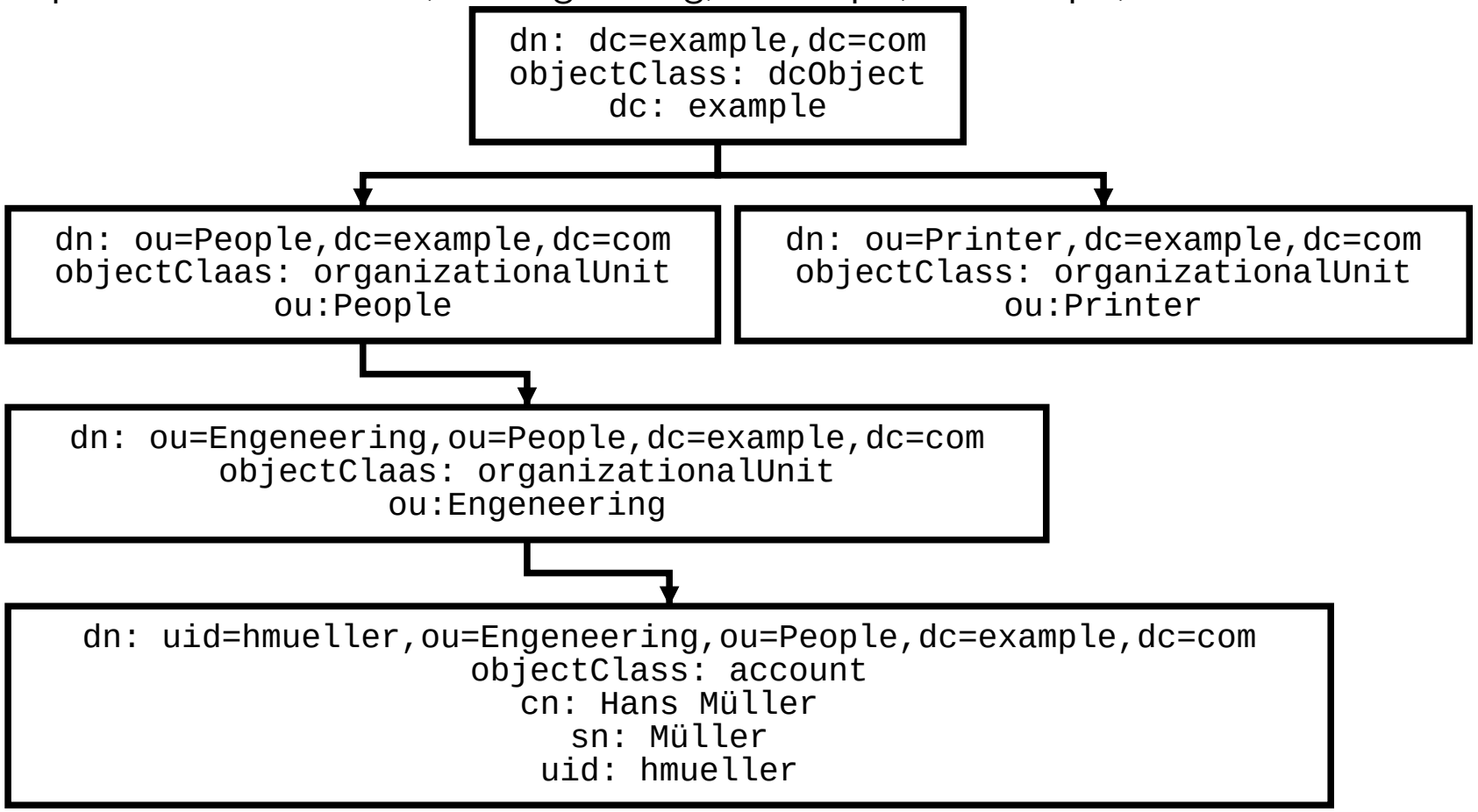
```
( 2.5.4.41 NAME 'name' EQUALITY caseIgnoreMatch
           SUBSTR caseIgnoreSubstringsMatch
           SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
( 2.5.4.3  NAME 'cn' SUP name )
( 2.5.6.8  NAME 'organizationalRole' SUP top STRUCTURAL MUST cn
           MAY ( x121Address $ registeredAddress $
                   ... einiges ausgelassen ...
                 $ l $ description ) )
```

LDAP - Namensmodell, Teil 1/3

- jedes Objekt ist über seinen DN (distinguished name) eindeutig identifiziert
- durch Zerlegung des DN kann es in einen Baum eingeordnet werden
- vergleichbar mit einem Dateisystem, aber drei wesentliche Unterschiede
  - es gibt keinen richtigen Root Eintrag (vgl. / im Dateisystem)
  - jeder Knoten kann auch Container für Daten sein (in einem Verzeichnis-Eintrag können keine Daten gespeichert werden)
  - Leserichtung bei Namensgebung (Dateisystem Wurzel $\longrightarrow$ Blatt)

LDAP - Namensmodell, Teil 2/3, Beispiel

Beispiel: dn: uid=hmueller,ou=Engineering,ou=People,dc=example,dc=com



LDAP - Namensmodell, Teil 3/3

- zunächst gibt es keinen Bedarf für eine hierarchische Gliederung, da z.B. eine Benutzerliste auch über den Suchfilter (objectClass=account) erzeugt werden kann
- Für Fragen wie
  - Welche Abteilungen gibt es ?
  - Welche Benutzer sind in einer Abteilung ?

  ist eine hierarchische Ordnung notwendig.
- es gibt kein allgemeingültiges Muster für einen Baum, individuelle Ausarbeitung erforderlich
- ein einmal eingeführter Baum ist in der Zukunft sehr schwierig zu ändern, da die Pfade häufig in Clients “hartcodiert” sind (vgl. Verzeichnisnamen in Skripten)

LDAP - Sicherheitsmodell

- eine genaue Zugriffssteuerung ist über ACLs (access control lists) am Server bis hin zu Attributebene möglich
- Sicherheitskonzept muss individuell ausgearbeitet werden

Tip:

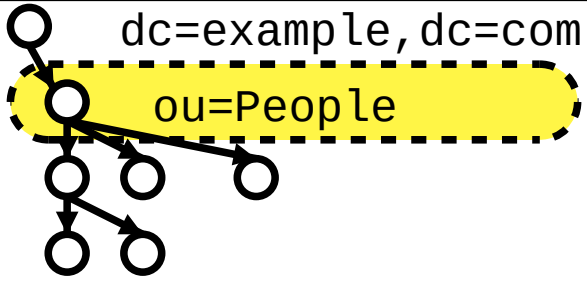
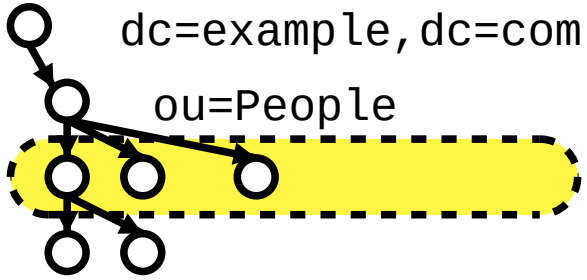
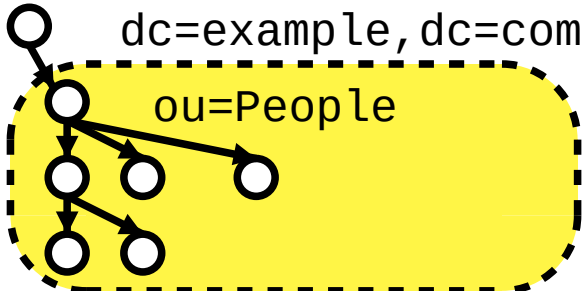
- bei einem LDAPv2 Bind wird Paßwort im Klartext übermittelt → Sicherheitsrisiko !
- Lösung: LDAPv2 Bind sperren → Problem: manche Software ist noch nicht an LDAPv3 angepasst
- erst ab LDAPv3 sind Hash-Funktionen wie MD5 möglich
- noch besser ab LDAPv3: Verschlüsselung mit TLS

LDAP - Suchen

Parameter für die LDAP Suchanfrage

- Basis Objekt (z.B. DN des ersten Entries)
- Suchtiefe (base, onelevel, sub)
- Optionen für Alias-Auflösung
- Maximale Größe der Ergebnissmenge (0 bedeutet alles bis Server-Maxmimum)
- Zeitlimit in Sekunden
- Nur Attribute (boolean)
- Suchfilter
- Liste gewünschter Attribute

LDAP - Suchen - Suchtiefe

Basis Objekt, Suchtiefe	Suchbereich
ou=People,dc=example,dc=com, base	
ou=People,dc=example,dc=com, onelevel	
ou=People,dc=example,dc=com, sub	

LDAP - Suchen - Suchfilter

Suchfilter	Filtertyp
(sn=Müller)	Gleicheitsfilter
(sn=Mü*)	Substring Filter
(sn~=Müller)	Ähnlichkeitsfilter
(telephoneNumer=*)	Existenzfilter
(age<=21)	Bereichsfilter >= und <=
(!(age<=21))	Ersatz für (age>21)
(&(sn=Müller)(L=Nürnberg))	Und Verknüpfung
(|(sn=Müller)(sn=Huber))	Oder Verknüpfung

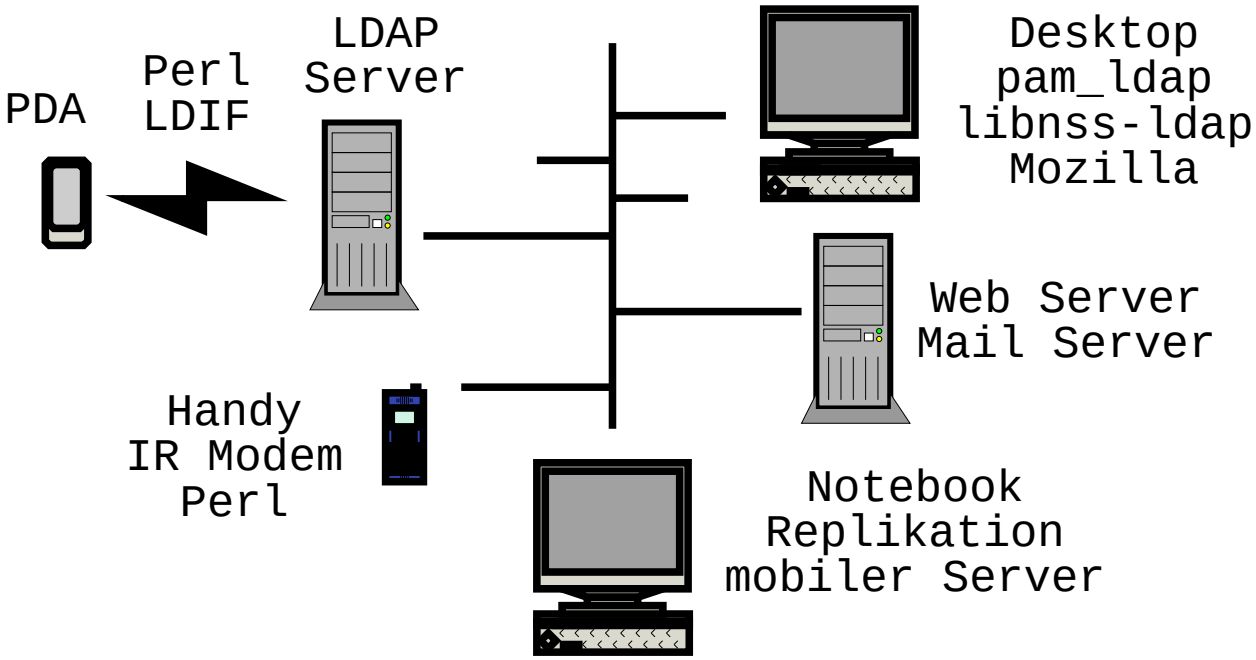
Zeichen	Escape Sequenz
*	\2A
(	\28
)	\29
\	\5C
NUL	\00

LDAP - Suchen - Attributliste

Attribut Liste	Rückgabewerte
cn, sn, givenName	nur cn, sn und givenName
*	alle Anwenderattribute (ohne operational attributes)
1.1	keine Attribute
modifiersName	nur modfifiersName (operational attribute)
+	alle operational attributes (Achtung: war nur mal Draft, nie RFC!)

Anwendung aus der Praxis

Adressbuchproblem



Adressbuchproblem - PDA zu LDIF

- Idee: vcf2ldap Perl Skript
- VCF Export des PDAs wird mittels Perl Skript in LDIF gewandelt
- LDIF wird mittels `ldapadd` importiert
- Trick
  - ISO8859-1 nach UTF-8 Konvertierung ab Perl 5.8
  - `binmode(STDOUT, ":utf8");  # man perluniintro`

Adressbuchproblem - OpenLDAP der LDAP Server

- slapd - `http://www.openldap.org`
- Kann mit unterschiedlichen Backends umgehen (bdb,ldbm)
- Nicht immer sind alle Backends bei vorkompilierten Distributionspacketen aktiv
- Empfehlenswerte Optionen in slapd.conf
  - `schemacheck on`
  - `loglevel 99` (zum Testen, sonst sehr still)
  - `allow bind_v2` (für Mozilla, Problem...)
  - `rootdn "cn=admin,dc=example,dc=com"`
- `slapd -t -d 255` zum Syntax Check der Konfigdateien
- Kommandozeilen-Tools

Operation	Online	Offline (über LDIF)
Update	ldapadd, ldapmodify	slapadd
Abfrage, Dump	ldapsearch	slapcat

Adressbuchproblem - OpenLDAP erster Login

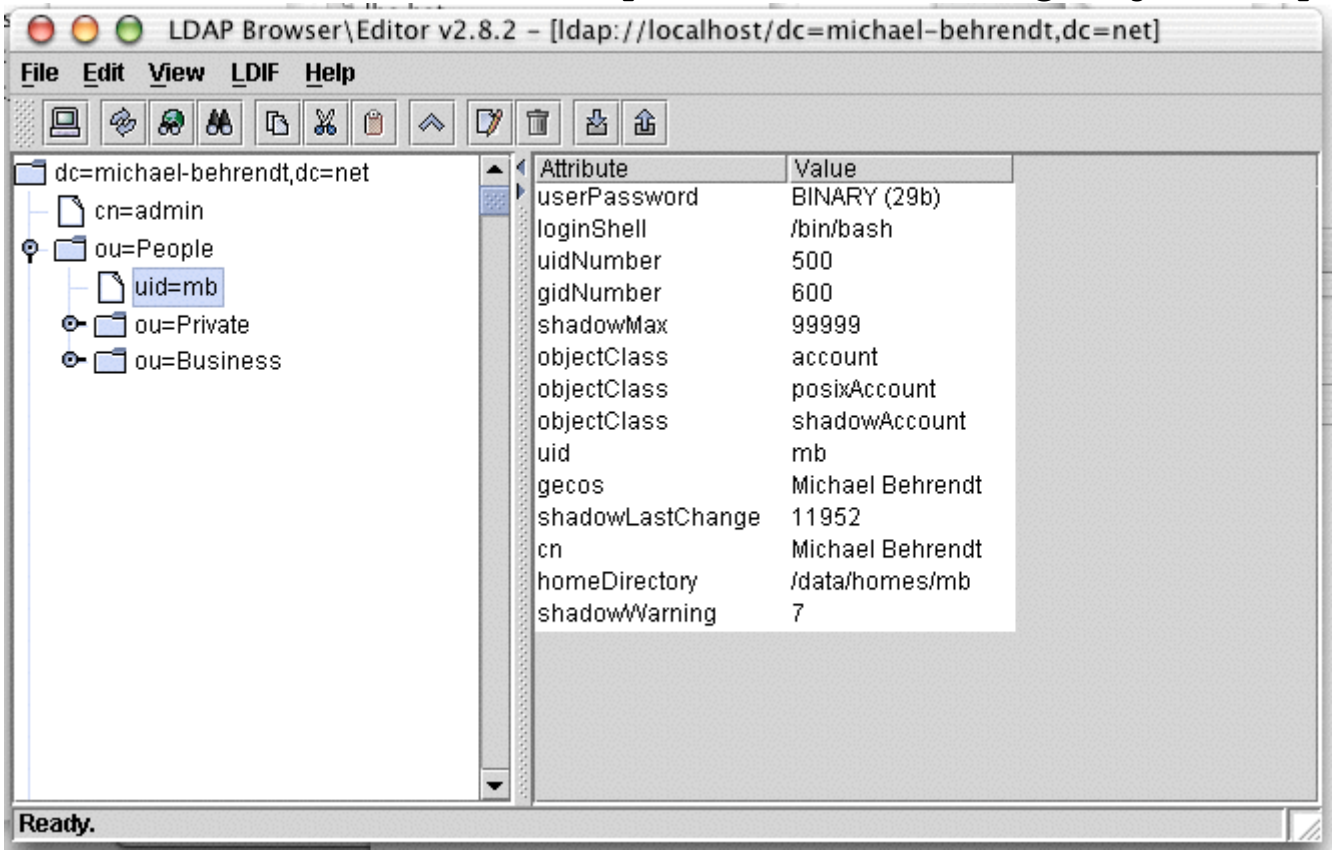
- Am Anfrang Henne Ei Problem
  - Admin Login braucht Datenbasis
  - Datenbasis wird grade durch Admin angelegt
- Lösung: Offline Modus Datenbasis anlegen (Passwort mittels `slappasswd` hashen)
- `slapadd -v << EOF`

```
dn: dc=example,dc=com
objectClass: dcObject
dc: example
description: LDAP Verzeichniss von example.com

dn: cn=admin,dc=example,dc=com
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: LDAP administrator
userPassword: {SSHA}WzHezyTBKY2d4kFLC123412341234
EOF
```

Adressbuchproblem - OpenLDAP erster Login

- ab jetzt einloggen als “cn=admin,dc=example,dc=com” möglich
- nützliches Tool ldapbrowser `http://www.iit.edu/~gawojar/ldap`



Adressbuchproblem - Handy Telefonbuch

- Idee: Programmierung des Handy Telefonbuches mit Daten aus LDAP
- Trick: Bei (Nokia-)Handys mit eingebautem Modem kann über AT Kommandos auf Telefonbuch zugegriffen werden
  - AT+CPBF Find Phone Book Entries
  - AT+CPBR Read Phone Book Entries
  - AT+CPBS Select Phone Book Memory Storage
  - AT+CPBW Write Phone Book Entries

Adressbuchproblem - Zugriff aus Java für Webschnittstelle

- JNDI - Java Naming and Directory Interface
- ab Java 1.3 im Standard
- `ftp://ftp.javasoft.com/docs/j2se1.3/jndi.pdf`

Fragen und Antworten

In eigener Sache ..

Danke für Ihre Aufmerksamkeit !