OpenLDAP
Lightweight Directory Access Protocol
V1.2 – 12.7.2003

Thomas Birnthaler
Hermann Gottschalk

www.ostc.de

Firmenprofil OSTC GmbH

- OSTC = Open Source Training and Consulting
- 2 Mitarbeiter mit langjähriger IT-Erfahrung
  - **Thomas Birnthaler**, Dipl.-Informatiker, eMail: tb@ostc.de
  - **Hermann Gottschalk**, Dipl.-Physiker, eMail: hg@ostc.de
  - SCLT (SuSE Certified Linux Trainer)
- Portfolio
  - IT-Training (UNIX, Linux, Netzwerk, SW-Entwicklung)
  - IT-Consulting (Security und Open Source)
  - IT-Sicherheits-Lösungen auf UNIX/Linux-Basis
- Partnerschaften
  - SuSE Business Partner
  - GeNUA Vertriebspartner

Inhaltsverzeichnis

Inhaltsverzeichnis

Was ist LDAP?

- LDAP = **L**ightweight **D**irectory **A**ccess **P**rotocol
  - Unterstützt einen "**Verzeichnisdienst**" (Directory Service)
  - In den frühen 90ern an der **Universität von Michigan** entwickelt und implementiert
- Es gibt verschiedene **kommerzielle** LDAP-Server
  - iPlanet (SUN Solaris)
  - NDS (Novell Directory Services)
  - eDirectory (Novell)
  - Active Directory (Microsoft)
- **OpenLDAP** ist eine "Open Source"-Implementierung von LDAP
  - **Standard** unter Linux und BSD-UNIX
  - Auch auf allen anderen UNIX-Systemen nutzbar

Was ist LDAP?

- **Vorteile von "Open Source"**
  - Hält sich an Standards (RFC = Request for Comments)
  - Bei allgemeinem Interesse entwicklen viele mit
  - Wird von vielen "begutachtet" → **Sicherheit**
  - **Portabel** → Auf vielen Systemen einsetzbar
  - Fehler können selbst behoben werden
  - Fehlende Features können selber ergänzt werden
  - Niedrige Kosten
  - Keine Auslieferung an einen Hersteller
- **Nachteile von "Open Source"**
  - Kein "Hersteller" der direkt Support leistet
  - Keine Garantie
  - Produkt wird evtl. nie fertiggestellt

Was ist LDAP?

- Die **GNU General Public Licence** (**GPL**) besagt:
  - Zeitlich unbegrenzt nutzbar
  - Software darf beliebig **kopiert** und weitergegeben werden (als **Quellcode**, *das **Copyright** muß unverändert bleiben*).
  - Quellcode darf beliebig **verändert** und verändert weitergegeben werden (*Änderungen sind zu kennzeichnen*).
  - **Keine Lizenzgebühr** erlaubt (*aber Handlingkosten, Gebühren für Support, Garantie, ...*).
  - **Keinerlei Garantie** für die Software (*wo gibt‘s die schon...?*).
- ***Mittels GPL-Software erstellte Software** „erbt“ diese Lizenz automatisch (außer LGPL = Library GNU Public Licence).*
- Weitere Open Source "**Lizenzen**"
  - BSD, LPGL, X Consortium, Artistic, MPL, QPL, ...

Was ist LDAP?

- **Probleme** in heutigen IT-Landschaften
  - "Gewachsene" Strukturen
  - Zugriff auf viele verschiedene **Ressourcen** notwendig
  - Viele verschiedene Verzeichnisse mit
    - Informationen in **Spezial-Formaten**
    - **Redundanten** Informationen
    - Verschiedenartigen **Schnittstellen**
  - Viele Import / Export-Programme

- **Wunsch** in heutigen IT-Landschaften
  - Zentrale, einheitliche Datenhaltung
  - Einheitliche Schnittstelle
  - Zugriffskontrolle per Authorisierung

Was ist LDAP?

- Die Lösung: **Verzeichnisdienste / Directory Services**
  - Bestandteil des täglichen Lebens
    - Telefonbuch ("White Pages")
    - Branchenfernsprechbuch ("Yellow Pages")
  - Besserer Begriff: "**Datenbank**"
  - Beispiele aus dem IT-Bereich
    - DNS, NIS, Whois, LDAP, X.500, NDS, Finger, ...

- **LDAP**
  - "Heisser Tip" für Systemadministratoren
  - Speicherung und Abruf von Informationen über
    - **Objekte**: Personen und sonstige Einheiten einer Organisation
    - **Eigenschaften**: Telefon, Lage des Büros, Benutzer-Account, Passwort, Passfoto, ...

Was ist LDAP?

- Was ist ein **Verzeichnisdienst / Directory Service**?
  - Sammlung von **Objekten**
  - In einer gewissen **Ordnung**
  - Mit **Detailinformationen** zu jedem Objekt

- Beispiel **Telefonbuch**
  - Objekte = Personen (und Firmen)
  - Ordnung = Alphabetisch
  - Detailinformationen = Telefonnummer(n) [+ Adresse]

- Elektronische Verzeichnisse in Computern sind natürlich **viel flexibler** als solche auf Papier

Was ist LDAP?

- **Unterschied zu normalen Datenbanken**
  - Auf Finden und Auslesen von Informationen **spezialisiert**
    - Hauptoperation: **Suchen** → fortgeschrittene Suchbefehle
  - Grosse Anzahl von Anfragen gleichzeitig möglich
  - **Schreibzugriff** beschränkt auf Administratoren
  - Eignen sich **nicht für häufig veränderte** Informationen
    - Okay: Info über Netzwerkdrucker
    - Schlecht: Aufträge in Druckerwarteschlangen
  - Meist "**verteilt**" auf mehrere Standorte
    - Lokale Informationen
    - Zugriff auf entfernte Informationen
      - Per Verweis ("referral")
      - Transparent

Was ist LDAP?

- **Unterschied zu normalen Datenbanken**
  - Verteilte Speicherung
  - Datenreplikation
  - Hierarchische Struktur

  } ® Skalierbar
- Ablage der Information in
  - **Entries** (Einträge, Objekte) mit
  - **Attributen** (Merkmale)
- **Datenbank-Analogie**
  - Entry = Record / Tabellenzeile
  - Attribut = Feld / Tabellenspalte
- LDAP-Daten werden immer in irgendeiner Art von (spezialisierter) **Datenbank** abgelegt

Was ist LDAP?

- "**Objekte**" können z.B. sein
  - Standorte
  - Abteilungen
  - Personen
  - Drucker
  - Räume
  - Rechner

- Die grundlegende Struktur (welche Objekte gibt es und welche Attribute besitzen sie) wird als "**Schema**" bezeichnet
  - Für bestimmte Anwendungsfälle vordefiniert verfügbar
  - An lokale Bedürfnisse anpassbar
  - Je stärker man sich an bereits vordefinierte Schemata hält, desto "**zukunftssicherer**" ist die Datenbank

Was ist LDAP?

- **Unterschied zu normalen Datenbanken**
  - "**Transaktionen**" werden **nicht** unterstützt
    - Kein **"Alles-oder-nichts"-Prinzip**, d.h. kurzeitig werden
    - Anomalien und Inkonsistenzen beim Update von Verzeichnissen in Kauf genommen
  - Nur kleine Menge an "**Datentypen**" möglich
    - Namen
    - Telefonnummern
    - IP-Adressen
  - Einfache und optimierte **Abfragesprache** statt SQL (Structured Query Language)
  - Hält **offene Standards** ein (RFC = Request for Comment)

Was ist LDAP?

- **Ein Verzeichnisdienst ist kein(e) ...**
  - ... **Allzweckdatenbank**
    - Transaktionen, Aktualisierung, Relational, Referentielle Integrität, Stored Procedures, ...
  - ... **Dateisystem**
    - *Verstauen von Large Objects (Verweise darauf schon*)
  - ... **Ersatz für lokale Dateiablage**
    - Für nicht-lokalen Einsatz gedacht
  - ... **Netzwerk-Management-Tool**
    - Tracking der schnell ändernden Zustände beobachteter Objekte

Was ist LDAP?

- **Welche Probleme löst LDAP?**
  - **Daten nicht normalisiert**
    - Viele Verzeichnisse
    - Informationen mehrfach abgelegt
  - **Aktualisierung** von Informationen an **mehreren Orten**
    - Mühsam
    - Zeitaufwändig
    - Fehlerträchtig
  - **Zugriffskontrolle** mehrfach unterschiedlich gelöst
    - Sichere Passwort-Politik
    - Sichere Transport-Protokolle
    - Sichere Authentifizierung
    - Ausreichende Verfügbarkeit

Was ist LDAP?

- **LDAP ermöglicht einfach und schnell**
  - Normalisierte Datenhaltung
  - Zentrale Verwaltung
  - Konsistenz der
    - Schnittstelle zum User
    - Richtlinien für das Netzwerk-Management
    - Security Policy

Was ist LDAP?

- **Folgende Eigenschaften von LDAP unterstützen seine Ziele**
  - Universaldesign – **Schemata**
  - Einfaches Protokoll – "**Lightweigt**"
  - Verteilte Architektur – "**Referrals**"
  - Integration von Sicherheitskonzepten
    - **TLS** (Transport Layer Security)
    - **SASL** (Simple Authentication and Security Layer)
    - **ACL** (Access Control Lists)
  - Offener Standard – IETF (Internet Engineering Task Force)
  - Server-Funktionen und Schematas von Clients abfragbar
    → Interoperabilität
  - **Internationalisierung** – UTF-8 (spezielle Unicode Codierung)
  - Erweiterbarkeit – Controls und Extensions

Was ist LDAP?

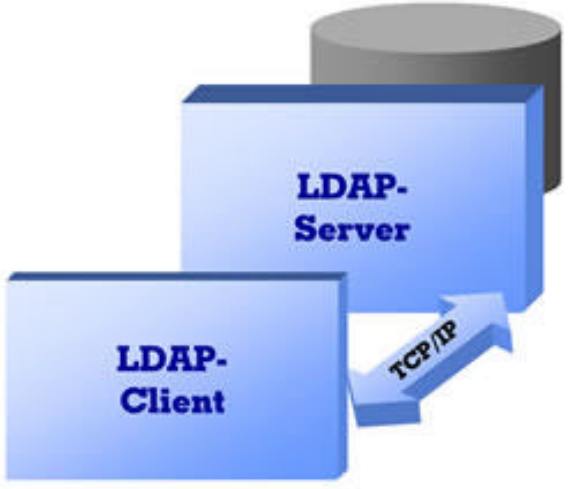
- **Was ist an LDAP nicht so toll?**
  - Wird nicht immer **vollständig unterstützt**
    - SUN Solaris: keine Verschlüsselung
    - Microsoft: AD erst vollständig kompatibel ab .NET
  - **Authentifizierung-Schnittstelle** wird teilweise weggelassen
  - **Entscheider** wissen nichts über LDAP
  - Kaum **Support** verfügbar
  - **Einsteigerdokumentation** kaum verfügbar (*die RFCs sind ein "hartes Brot"*)
  - Keine guten **LDAP-Bücher** verfügbar

Was ist LDAP?

- **Ist LDAP ein Protokoll oder ein Verzeichnis?**
  - Das "P" in LDAP sagt: ein **Kommunikations-Protokoll**
    - Definiert Transport und Format von Nachrichten zwischen
    - Einem Client und einem X.500-Verzeichnis
  - Ein X.500-Verzeichnis versteht gar keine LDAP-Nachrichten (OSI ↔ TCP/IP-Protokoll)
    - D.h. es ist ein Gateway-Prozess / Proxy / Front-End notwendig, der die LDAP-Anfragen "übersetzt".
    - Dieses ist der **LDAP-Server**, der wiederum ein Client des X.500-Servers ist.

Was ist LDAP?

- **Ist LDAP ein Protokoll oder ein Verzeichnis?**
  - X.500-Server sind kompliziert, ebenso das X.500-Protokoll.
  - Daher wurde versucht, die Informationen direkt im LDAP-Server abzulegen und den X.500-Server wegzulassen
    - Die LDAP-Server wurden komplizierter
    - Name: "**Standalone-LDAP-Server**"
  - Den Clients ist egal, wo die Informationen stehen
  - Nur noch TCP/IP-Protokoll



Inhaltsverzeichnis

Geschichte und Versionen von LDAP

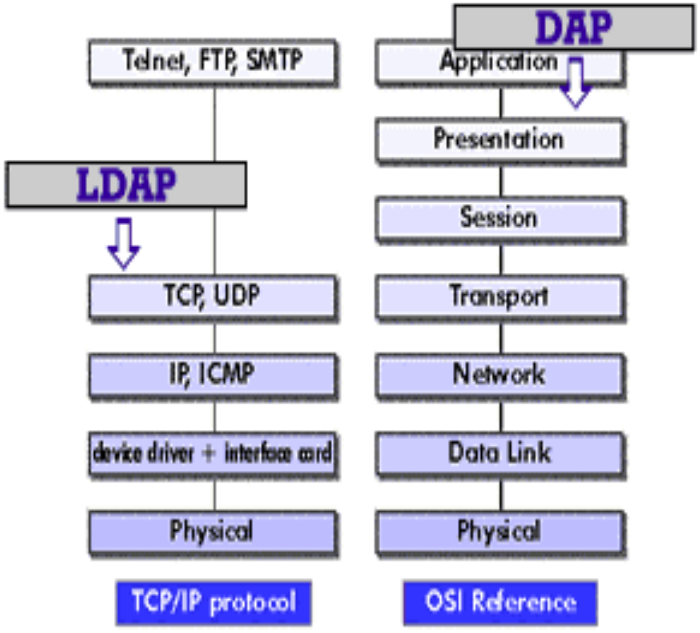
- Seit den 70er Jahren werden Standards entwickelt, die IT-Systemen verschiedener Hersteller ermöglichen, miteinander zu arbeiten
- **2 große Standardisierungsbewegungen**
  - Von der **Telekommunikations-Seite**
    - CCITT (Comitee Consultatif International Telephonique et Telegraphique) → ITU (International Telecommunications Union)
    - ISO (International Standards Organization)

      → Entwickeln gemeinsam das **OSI-Referenzmodell**
  - Rund um das entstehende **Internet**
    - IETF (Internet Engineering Task Force)
    - IAB (Internet Architecture Board)

      → Entwickeln gemeinsam **RFCs** (Request for Comments)

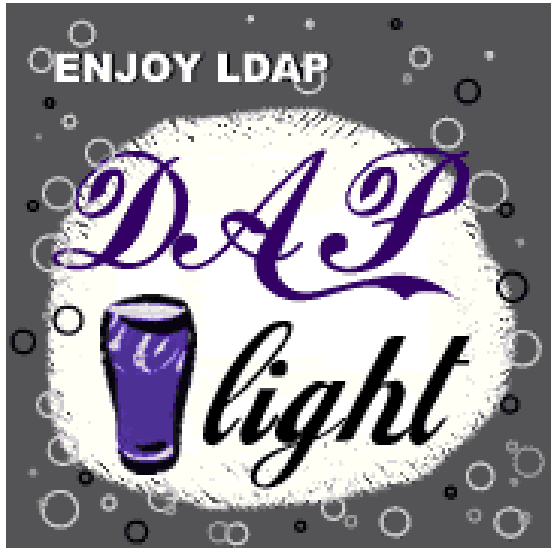
Geschichte und Versionen von LDAP

- Beide behandeln Problem aus **unterschiedlicher Perspektive**
  - **OSI**: Theoretisch, sehr komplex
  - **RFC**: Pragmatisch, möglichst einfach
- OSI behandelt wichtige Aspekte weit verteilter Systeme
  - 1988 – X.500 Directory Service
  - Hierarchischer Namensraum
  - Mächtige Suchfunktionen
  - **DAP** (Directory Access Protocol)
- OSI-Protokollstapel gibt es in vielen kleineren Umgebungen nicht
  - **LDAP** (Lightweight Directory Access Protocol)



Geschichte und Versionen von LDAP

- **LDAP macht folgendes anders**
  - Setzt auf TCP/IP auf
  - Verzichtet auf einige "esoterische" Funktionen
- Die aktuelle Version LDAP v3 fügt hinzu
  - **Internationalisierung** – Unicode
  - **Referrals** – Verweis auf andere Server – Verteilung der Information
  - **Sicherheit**: SASL und TSL
  - **Erweiterbarkeit**: Extensions + Controls
  - **Offenlegung** der Funktionen und Schemata: Clients können Server abfragen



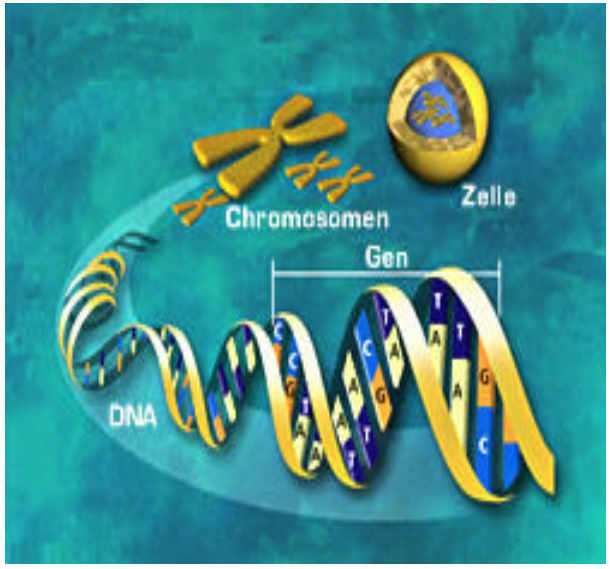
Inhaltsverzeichnis

Inhaltsverzeichnis

LDAP Konzepte und Architektur

Um ein LDAP-Directory zu benutzen oder zu konzipieren,
muss man die 4 Elemente seines **logischen Modell** kennen

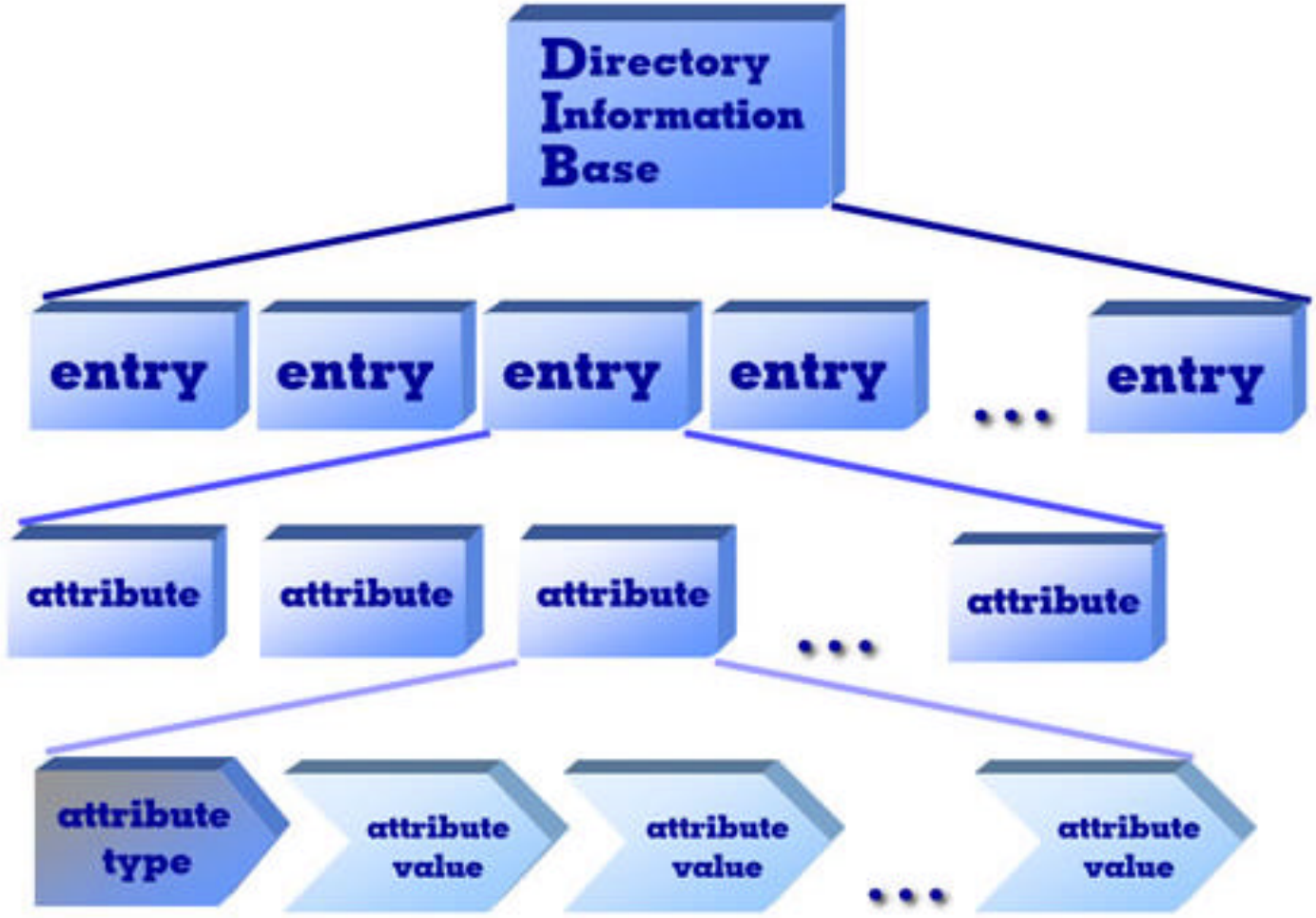
- **Information Model**
  Strukturierung der Informationen in einem LDAP-Verzeichnis
- **Naming Model**
  Identifikation und Adressierung dieser Informationen
- **Functional Model**
  Welche Operationen sind möglich?
- **Security Model**
  Wie werden die Informationen vor unauthorisiertem Zugriff geschützt?



Information Model

- **DIB** (**D**irectory **I**nformation **B**ase) bezeichnet die Gesamtheit aller Informationen im Verzeichnis (normalerweise unsichtbar)
- Besteht aus **Einträgen** (**Entries**)
  - Stehen zueinander in **hierarchischer Beziehung**
  - Enthalten Informationen über ein **Objekt der Realität**
- Ein Eintrag enthält eine Liste von **Attributen** (**Eigenschaften**)
  - Jedes Attribut hat einen **Attribut-Typ** (*festgelegt durch die **Syntax***)
  - Jedes Attribut hat **Attribut-Werte**

LDAP Konzepte und Architektur



Beispiele für Einträge

Verzeichnis	Einträge
Telefonbuch	Personen
Bibliothekskatalog	Bücher
Hochschulverzeichnis	Studenten, Dozenten, Angestellte, Fakultäten, Seminarräume, Vorlesungen, ...
Fahrplan	Linien, Haltestellen, Fahrzeiten, Umsteigemöglichkeiten, ...
IT-Infrastruktur	Rechner, Switches, Hubs, Server, Drucker, ...
Firma	Mitarbeiter, Räume, Abteilungen, Standorte, ...

LDAP Konzepte und Architektur

- **Ein Attribut kann**
  - keinen, einen oder viele Werte haben
- **Attribute können sein**
  - **Mandatorisch** (*müssen vorhanden sein*)
  - **Optional** (*dürfen fehlen*)
- **Die Attribut-Syntax definiert**
  - Erlaubten Wertebereich
  - Einschränkungen (Constraints)
    (*z.B. nur Ziffern, maximal Länge 8 Zeichen*)
  - Verhalten bei Suchoperationen
    (*z.B. Gross/Kleinschreibung, Leer-/Sonderzeichen ignorieren*)

Beispiele für Attribute

Objekt	Attribut
Person	PersNr, Name, Vorname, Größe, Augenfarbe, Passfoto, ...
Buch	ISBN, Autor, Titel, Verlag, Erscheinungsjahr, ...
Student	ImmatrikulationsNr, Fachrichtung, Name, Vorname, ...
Haltestelle	Adresse, Koordinaten, Linien, ...
Rechner	Name, IP-Adresse, Standort, Typ, Betriebssystem, SW, ...
Abteilung	AbtNr, Bezeichnung, Standort, Gebäude, Stockwerk, ...

Objektorientierter "Blick" auf LDAP

- Jeder Eintrag beschreibt ein "**Objekt**"
- Objekte sind Instanzen einer "**Objektklasse**"
- **Objektklasse**
  - Verallgemeinerte Beschreibung vieler gleichartiger Objekte
  - Jede Objektklasse besitzt eine Liste von **Attributen**
    - Zwingend vorgeschriebene (*mandatory*)
    - Nicht verbindlich vorgeschriebene (*optional*)

Objektklassen-Typen

- **Abstrakt**
  Nur benutzt, um daraus andere Objektklassen abzuleiten, d.h. ist eine Superklasse oder ein "Template" (z.B. `top`). Bequemer Weg, um "Attribut-Sammlungen" zu definieren, die eine Reihe von Klassen gemeinsam haben sollen
- **Strukturell**
  Einträge müssen immer zu einer derartigen Objektklasse gehören
- **Auxiliär** (Hilfsklasse)
  Sammlung von Attributen, die verschiedenen Einträgen zugeordnet werden sollen

Beispiel für Objektklassen

- Abstrakte Objektklasse
  `Weihnachtsplätzchen`
- Strukturelle Objektklassen
  `Spritzgebäck, Printen, Lebkuchen`
- Auxiliäre Objektklasse
  `Verzierung`

Schema

- Alle Objektklassen eines Directory-Servers werden in einer zusammenfassenden Beschreibung abgelegt, dem "Schema"
- Ein Schema beschreibt also:
  - Welche **Objektklassen** sind erlaubt
  - Welche **Attribute**
    - Müssen sie haben
    - Dürfen sie haben
  - Welche **Syntax** (Werte) sind für die Attribute erlaubt

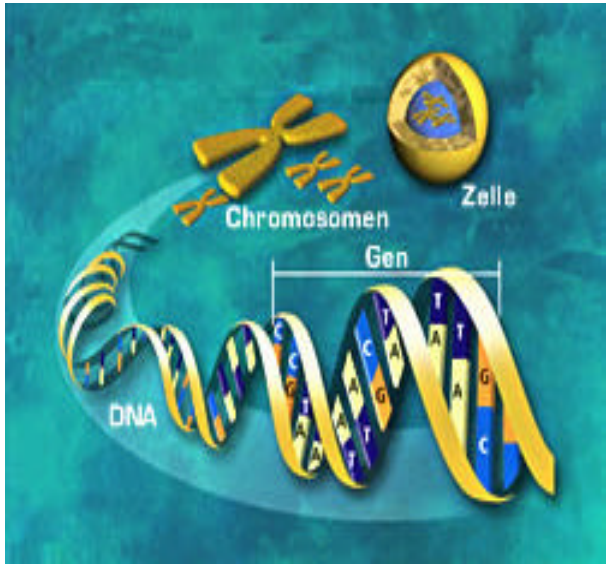
Beispiel

- Schema bestehend aus einer einzigen Objektklasse `Person`
- Zwingendes Attribut "Nachname" vom Typ String (Zeichenkette) aus Buchstaben `surname`
- Optionales Attribut "Telefonnummer" vom Typ String aus Ziffern, Leerzeichen und Bindestrich `telephoneNumber`

LDAP Konzepte und Architektur

Um ein LDAP-Directory zu benutzen oder zu konzipieren,
muss man die 4 Elemente seines **logischen Modell** kennen

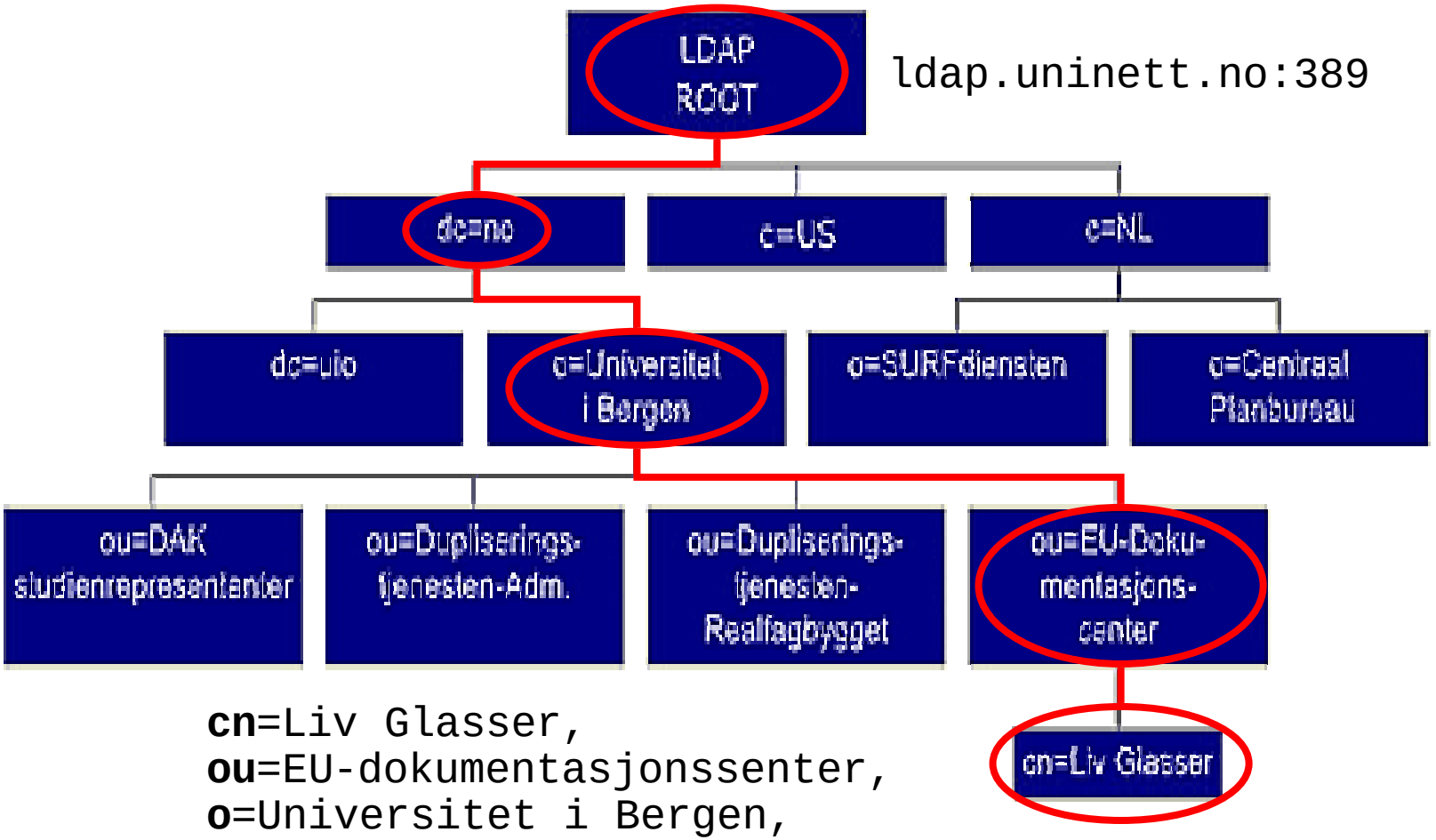
- ✓ **Information Model**
  Strukturierung der Informationen in einem LDAP-Verzeichnis
- **Naming Model**
  Identifikation und Adressierung dieser Informationen
- **Functional Model**
  Welche Operationen sind möglich?
- **Security Model**
  Wie werden die Informationen vor unauthorisiertem Zugriff geschützt?



Naming Model (**LDAP Namespaces**)

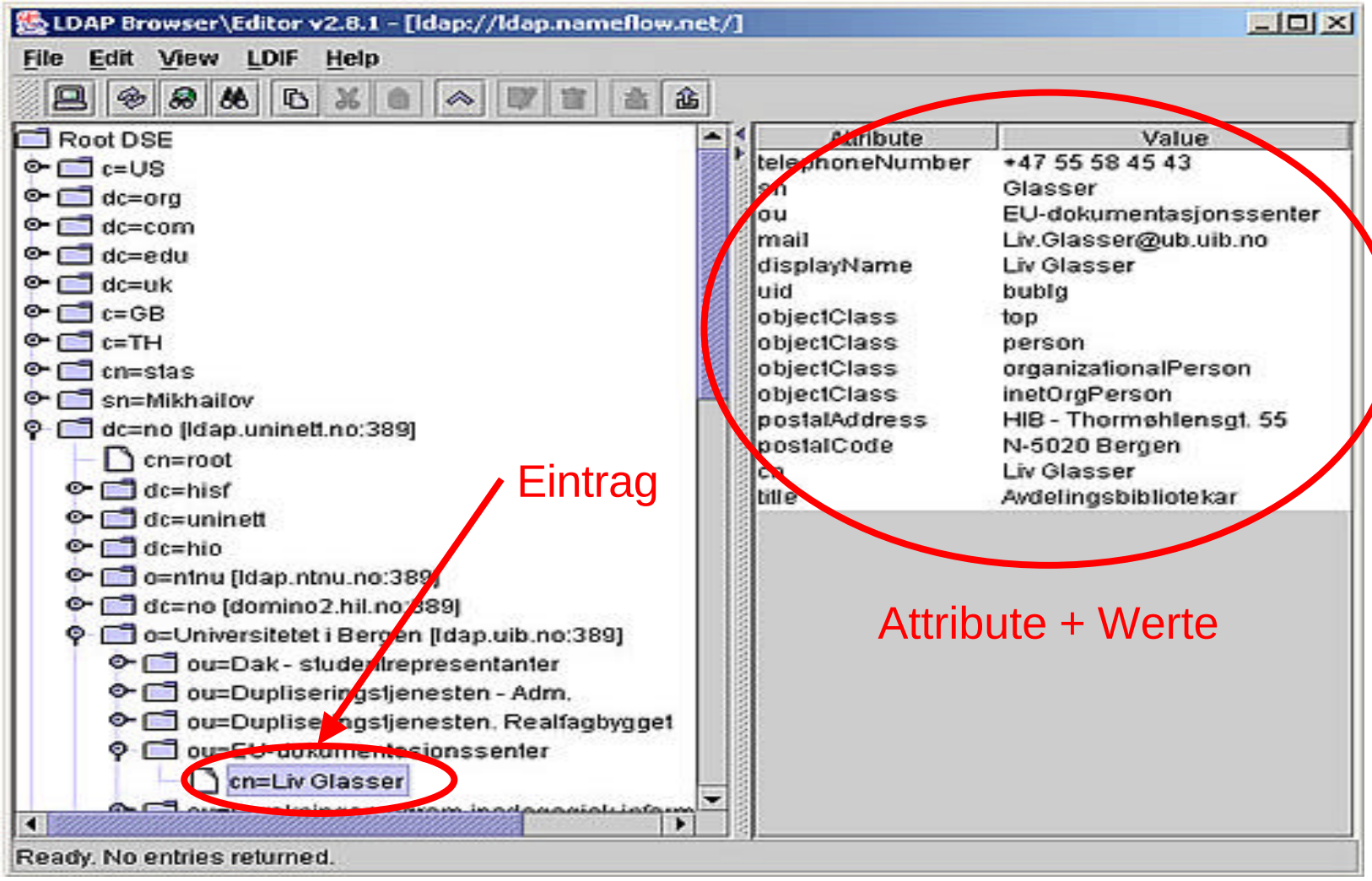
- Jeder Eintrag wird *eindeutig* über seinen **Distinguished Name** (**DN**) identifiziert.
  - Beschreibt, wo innerhalb der Verzeichnishierarchie sich der Eintrag befindet
  - "**Voll qualifizierter**" Name für den Eintrag (*analog FQHN*)
  - Einträge können Eltern-, Kinder- und Nachbar-Einträge haben.
- Ein **DN** besteht aus **Relative Distinguished Names** (**RDN**)
  - Ähnlich **Pfadangaben** in einem Verzeichnisbaum, die sich aus Verzeichnis- und Unterverzeichnisnamen zusammensetzen
  - Analog URLs **von rechts nach links** immer spezieller
- LDAP-Server-URL + DN sind **weltweit eindeutig**, z.B. `ldap://ldap.telekom.com/o=T-Online,c=de`

LDAP Konzepte und Architektur



cn=Liv Glasser,
ou=EU-dokumentasjonssenter,
o=Universitet i Bergen,
dc=NO

LDAP Konzepte und Architektur



Beispiel

- Eintrag von Typ `"Person"`: Abteilungsbibliothekarin im EU-Dokumentationszentrum der Universität Bergen in Norwegen
- Ihre DN ist (*eine Zeile!*)

```
cn=Liv Glasser,
ou=EU-dokumentasjonssenter,
o=Universitet i Bergen,
dc=NO
```

- Der LDAP-Server ist

```
ldap.uninett.no:389
```

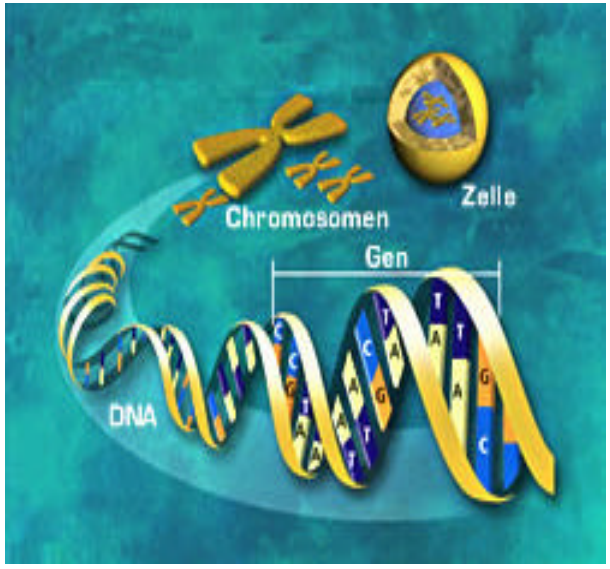

Portion oder **Partition**

- In LDAP-Namespaces kann es "**Unterabteilungen**" geben.
- Diese können auf verschiedene Server aufgeteilt werden.
- Es ist auch möglich, die gleiche Partition auf mehrere Server gleichzeitig zu verteilen (**Replikation**).

LDAP Konzepte und Architektur

Um ein LDAP-Directory zu benutzen oder zu konzipieren,
muss man die 4 Elemente seines **logischen Modell** kennen

- ✓ **Information Model**
  Strukturierung der Informationen in einem LDAP-Verzeichnis
- ✓ **Naming Model**
  Identifikation und Adressierung dieser Informationen
- **Functional Model**
  Welche Operationen sind möglich?
- **Security Model**
  Wie werden die Informationen vor unauthorisiertem Zugriff geschützt?



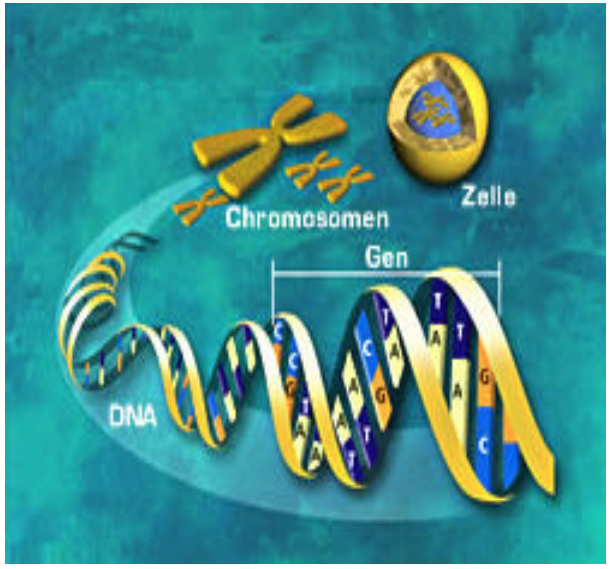
Functional Model

- Definiert Operationen, um Einträge zu finden, auszulesen und zu bearbeiten:
  - **Suche** nach Einträgen anhand bestimmter Kriterien
  - **Hinzufügen** von Einträgen
  - **Löschen** von Einträgen
  - **Ändern** von Einträgen
  - Ändern (**Umziehen**) des DN oder RDN von Einträgen
  - **Vergleich** von Einträgen

LDAP Konzepte und Architektur

Um ein LDAP-Directory zu benutzen oder zu konzipieren,
muss man die 4 Elemente seines **logischen Modell** kennen

- ✓ **Information Model**
  Strukturierung der Informationen in einem LDAP-Verzeichnis
- ✓ **Naming Model**
  Identifikation und Adressierung dieser Informationen
- ✓ **Functional Model**
  Welche Operationen sind möglich?
- ● **Security Model**
  Wie werden die Informationen vor unauthorisiertem Zugriff geschützt?



Security Model

- Informationen zwischen Client und LDAP-Server werden in Form von **Nachrichten** (**Messages**) ausgetauscht.
- Dazu muss eine **TCP/IP-Verbindung** aufgebaut werden
- Es finden folgende Schritte statt
  - **Binding**: Client meldet sich über Host-Name + Port-Nummer (389) am Server an
    - Authentifiziert mit Username + Passwort oder
    - Anonym → Standardberechtigungen
    - Evtl. wird die Sitzung sogar verschlüsselt (TLS)
  - **Operating**: Client setzt Aufträge an den Server ab, Der Server prüft ihre Zulässigkeit und Korrektheit, führt die Operation aus und liefert das Ergebnis an den Client zurück.
  - **Unbinding**: Client meldet sich wieder vom Server ab.

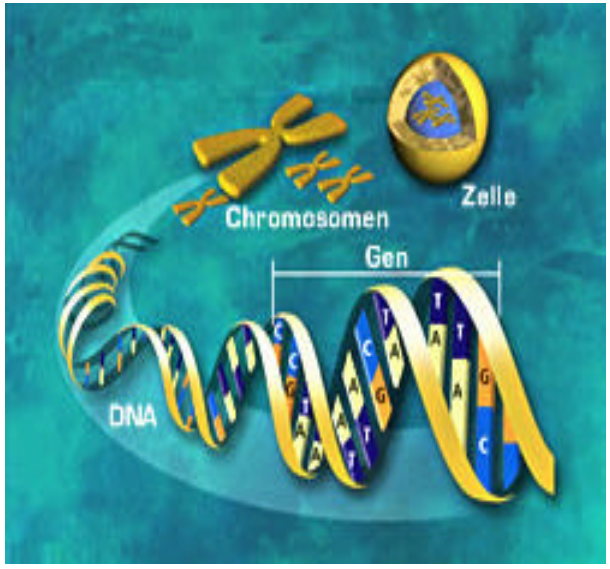
Typische Operation ist "Suchen"

- Einschränkung des zu durchsuchenden Teils des Verzeichnisbaums durch
  - DN und RDN
- Frei wählbare Suchkriterien anhand von
  - Attributen und ihren
  - Werten
- Festlegen, welche Informationen zurückgeliefert werden soll
  - Attributliste

LDAP Konzepte und Architektur

Um ein LDAP-Directory zu benutzen oder zu konzipieren,
muss man die 4 Elemente seines **logischen Modell** kennen

- ✓ **Information Model**
  Strukturierung der Informationen in einem LDAP-Verzeichnis
- ✓ **Naming Model**
  Identifikation und Adressierung dieser Informationen
- ✓ **Functional Model**
  Welche Operationen sind möglich?
- ✓ **Security Model**
  Wie werden die Informationen vor unauthorisiertem Zugriff geschützt?



Inhaltsverzeichnis

Objektklassen

- LDAP-Objekte sind **standardisiert**, um die Zusammenarbeit zwischen verschiedenen LDAP-Servern zu gewährleisten.
- Beispiele dafür sind:
  - `core`
  - `cosine`
  - `inetOrgPerson`
  - `nis`
  - `misc`
  - `samba`
- Diese Schema-Definitionen stehen meist in Dateien unter `/etc/openldap/schema`

Objektklassen

- Kennt man die Objektklasse(n) eines Eintrags, so weiß man schon eine ganze Menge über ihn, da seine Attribute und ihre möglichen Werte bekannt sind
- Beispiel `inetOrgPerson`
  - `cn` = Voller Name (**common name**)
  - `sn` = Nachname (**surname**)
  - `mail` = eMail-Adresse
  - `givenname` = Vorname
  - `initials` = Namenskürzel
  - ...
- Ein Eintrag kann **mehreren Objektklassen** zugeordnet werden, die Attribute "addieren" sich dann zu einer "Gesamtbeschreibung"

LDAP Data Interchange Format (LDIF)

- Dient zur **externen Repräsentation** von Verzeichnis-Daten.
  - Ladbar
  - Auslesbar
- Aufbau:
  - Pro Eintrag eine
  - Folge von Zeilen mit
  - Attribut + Werte-Paaren getrennt durch **Doppelpunkt**
  - Einträge werden durch **Leerzeilen** getrennt
- Die Daten im folgenden Beispiel "James Bond" sind im **LDIF-Format** dargestellt

Beispiel "James Bond"

Distinguished Name

```
dn: cn=James Bond, ou=MI6, dc=gov, dc=uk
```

Objektklassen

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

Beispiel "James Bond"

Attribute der Objektklasse person

```
sn:                Bond
cn:                James Bond
telephoneNumber:   020 7930 9007
telephoneNumber:   020 7930 9070
telephoneNumber:   020 7930 9700
userPassword:      {crypt}gerührt-nicht-geschüttelt
description:       Agent 007 of Her Majesty's ¿
                   Secret Service MI6
```

Mehrfacher Wert

Beispiel "James Bond"

Attribute der Objektklasse `organizationalPerson`

```
ou:                          MI6
title:                       Commander
street:                      The Enquiries Desk
postOfficeBox:               3255
st:                          CT
postalCode:                  SW1P 1AE
facsimileTelephoneNumber:    020 7930 9000
```


Beispiel "James Bond"

Attribute der Objektklasse `inetOrgPerson`

```
departmentNumber:   00
employeeType:       permanent
givenName:          James
initials:           JB
jpegPhoto:          james.jpg
audio:              james.wav
homePhone:          020 7930 9007
pager:              Opening the toy cabinet
preferredLanguage:  English
userCertificate:    certs/jb_cert.pem
```


Vielen Dank für Ihre Aufmerksamkeit!
Für Fragen stehen wir Ihnen zur Verfügung

Thomas Birnthaler
Hermann Gottschalk

www.ostc.de