Verzeichnisdienste

DNS

NIS, NetWare Bindary

X.500, LDAP

eDirectory, Active Directory, OpenLDAP

Was sind Verzeichnisdienste?

- Dienste zum Bereitstellen und Verwalten von Informationen
- Stellen z.B. eine zentrale Sammlung von Netzwerkdaten zur Verfügung (Benutzer, Computer, Zugangsdaten, Richtlinien,….)
- Die Speicherung der Informationen erfolgt in einer hierarchisch angelegten (verteilten) Datenbank

- schon seit Mitte der 80er bekannt !

Einteilung der Verzeichnisdienste

- Applikationsspezifisch
  - DNS
- Plattformabhängig
  - NIS
  - NetWare Bindery
- Allgemeingültig
  - X.500, LDAP
  - eDirectory
  - Active Directory
  - Open LDAP

Applikationsspezifische Verzeichnisdienste

- Für abgegrenzte Aufgabengebiete konzipiert
- Optimum zwischen Funktionalität und Ressourcenbedarf
- Domain Name Service (DNS) ist der heute am häufigsten genutzte Dienst.
- Weiter Beispiele sind ERP-Systeme wie SAP, ABAS, …

Plattformabhängige Verzeichnisdienste

- Network Information Service (NIS)
  - Im Unix-Bereich etablierte Verzeichnisdienst
  - Von Firma Sun plattformoffen konzipiert
  - Nur für kleinere Netzwerke mit bis zu 10.000 Nutzern, veraltet.
- NetWare Bindery
  - mit der NetWare Version 2 eingeführt
  - kein Domain- und kein Replikationskonzept
  - Abgelöst von Novell Directory Services für NetWare 4

Allgemeine Verzeichnisdienste

- **X.500**
  - Vater aller allgemeinen Verzeichnisdienste
  - große Komplexität, hohe Systemanforderungen
- **Novell Directory Service (NDS) und eDirectory**
  - baut auf viele Konzepte von X.500 auf
  - schon seit der Netware Version 4 (Einführung 1994/95) im Einsatz
  - eDirectory für verschiedene Betriebssysteme verfügbar
- **Active Directory**
  - Mit Windows 2000 von MS eingeführt (2001)
  - Der Aufbau ist vergleichbar mit dem Konzept des DNS, Domänenstruktur.
  - Auch für Linux unter SAMBA verfügbar
- **LDAP**
  - leichtgewichtige Alternative zum X.500-Zugriffsprotokoll
  - Weniger komplex und geringere Systemanforderungen als X.500

X.500 Allgemeines

- Bei X.500 handelt es sich um eine Empfehlung für einen Verzeichnisdienst von der International Telecommunication Union (ITU (http://www.itu.int)) im Rahmen der X-Serie (Data Networks and Open System Communications). Die Empfehlung erschien erstmals im Jahr 1988. Eine der Hauptaufgaben der ITU ist es, internationale Standards zur weltweiten Kommunikation vorzuschlagen.
- Die Empfehlung von X.500 besteht aus zehn Dokumenten. Alle sind auch unter ISO 9594-1...10 als Standard von der International Organization for Standardization (ISO (http://www.iso.org)) aufgenommen worden.

Aufbau eines X.500-Verzeichnisses

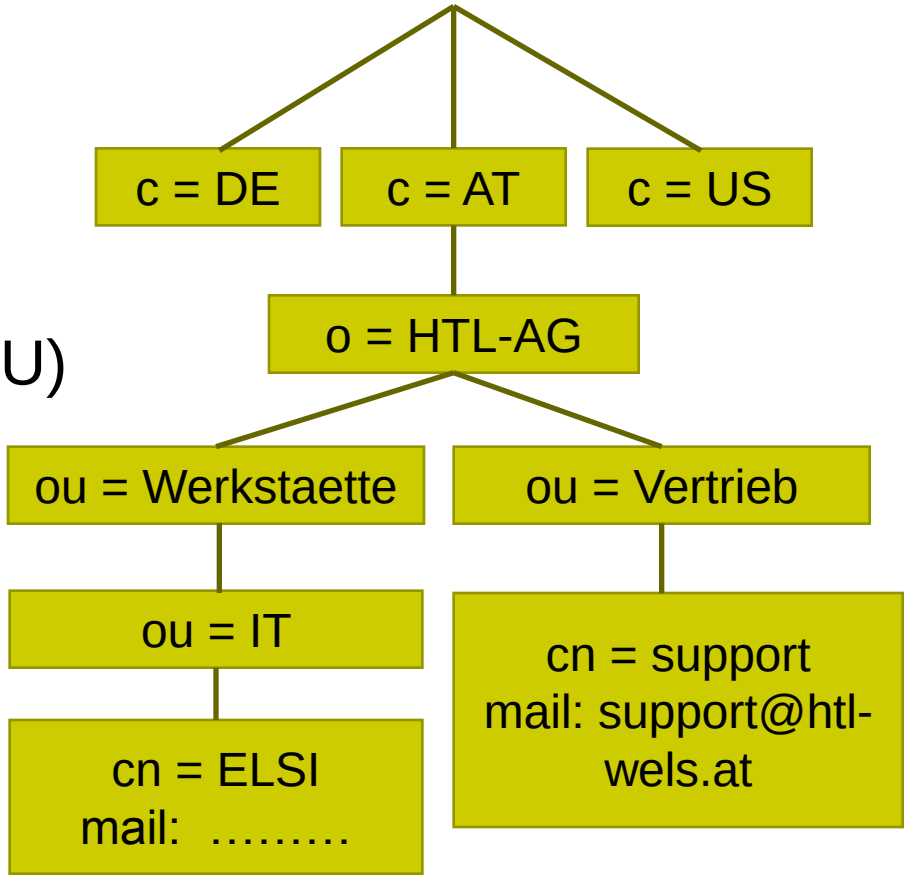
- Der Grundgedanke von X.500 ist ein globales und verteiltes Verzeichnis, auf das man von überall zugreifen kann. Es ist baumartig strukturiert mit einem namenlosen Wurzelobjekt, der Root.
- Die durch das Verzeichnis bereitgestellten Daten bezeichnet man als Directory Information Base (DIB), den Baum selbst als Directory Information Tree (DIT).
- Für die Einträge sind Objektklassen definiert, wobei jeder Eintrag mindestens einer dieser Klassen angehört. Diese Objektklassen werden auch als Schema bezeichnet.
- Innerhalb jeder Objektklasse gibt es wiederum Attributtypen, von denen mindestens einer vorhanden sein muss.
- jeder Eintrag in einem X.500-Verzeichnis ist also die Instanz einer oder mehrerer Objektklassen und enthält einen oder mehrere Werte für die einzelnen Attributtypen.

Aufbau eines X.500-Verzeichnisses

- Jedes Attribut hat einen Typ
- Die Typenbezeichnung eines Attributs sind meist einfach zu merkende Kürzel wie z.B. *cn* (für *common name*), *ou* für (*organizational unit*), *s* (für *state*), *c* (für *country*), *mail* (für *e-mail address*), dc (für domain component), …..
- Der Wert eines Attributs wird in Form einer Zeichenkette spezifiziert
- Die erlaubten Werte eines Attributs sind vom Typ abhängig
- Die Objekte werden in einer hierarchischen Struktur gespeichert, die politische, geographische oder organisatorische Grenzen widerspiegelt
- Die größten Einheiten werden an die Spitze des Verzeichnisbaumes gestellt, der sich nach unten immer weiter auffächert.

Beispiel: Baum und Attribute

- Country (C)
- Common Name (CN)
- Organization Name (O)
- Organization Unit Name (OU)



Namensraummodell

- Directory Information Tree (DIT)
  - Jeder Knoten hat 0 bis *n* Kinderknoten
  - Jeder Knoten hat genau 1 Elternknoten
- Objekte sind Baumknoten
- Jeder Eintrag hat in seiner Hierarchieebene einen eindeutigen Namen: Relative Distinguished Name (RDN)
- Alle RDNs von dem Objekt bis zur Wurzel ergeben den
- Distinguished Name (DN) eines Objektes, der dieses eindeutig identifiziert

Beispiele für DN:
CN=elsi,OU=lehrer,OU=benutzer,DC=schule,DC=local
CN=elsi,OU=lehrer,O=htl-wels,C=AT

X.500: Benutzerzugriff

Der Zugriff auf das Verzeichnis erfolgt über einen so genannten Directory User Agent (DUA) mit folgenden Operationen:

Read	Auslesen eines Attributes
Compare	Vergleichen eines Wertes mit einem Attribut
List	liefert eine Liste der RDNs, ab momentaner Baum-Position in die Unterverzweigungen hinein
Search	Durchsuchen eines Baumbereichs
Abandom	Abbrechen einer Operation
Add	Hinzufügen eines Eintrags mit Attributen
Remove	Entfernen eines Eintrags
Modify	Änderung eines Eintrags, ändern oder hinzufügen von Attributen
Modify Distinguished Name	Ändern und verschieben eines RDN

X.500: Vernetzung und Replikation

- X.500 definiert ein verteiltes Verzeichnis. Das bedeutet, dass die Daten nicht zentral gespeichert sein müssen. Es existiert ein Netz von Servern, die jeweils nur einen Teilbaum verwalten. Bei Bedarf kommunizieren diese untereinander, beispielsweise um Anfragen weiterzuleiten, die nicht den eigenen Datenbestand betreffen. Die einzelnen Server bezeichnet man auch als Directory System Agents (DSA).

- Um Lesezugriffe auf das X.500-Verzeichnis zu beschleunigen, kann es sinnvoll sein, für einen Teilbaum mehrere Server zur Verfügung zu stellen. Zudem erhöht sich dadurch auch die Ausfallsicherheit für den entsprechenden Teilbaum.

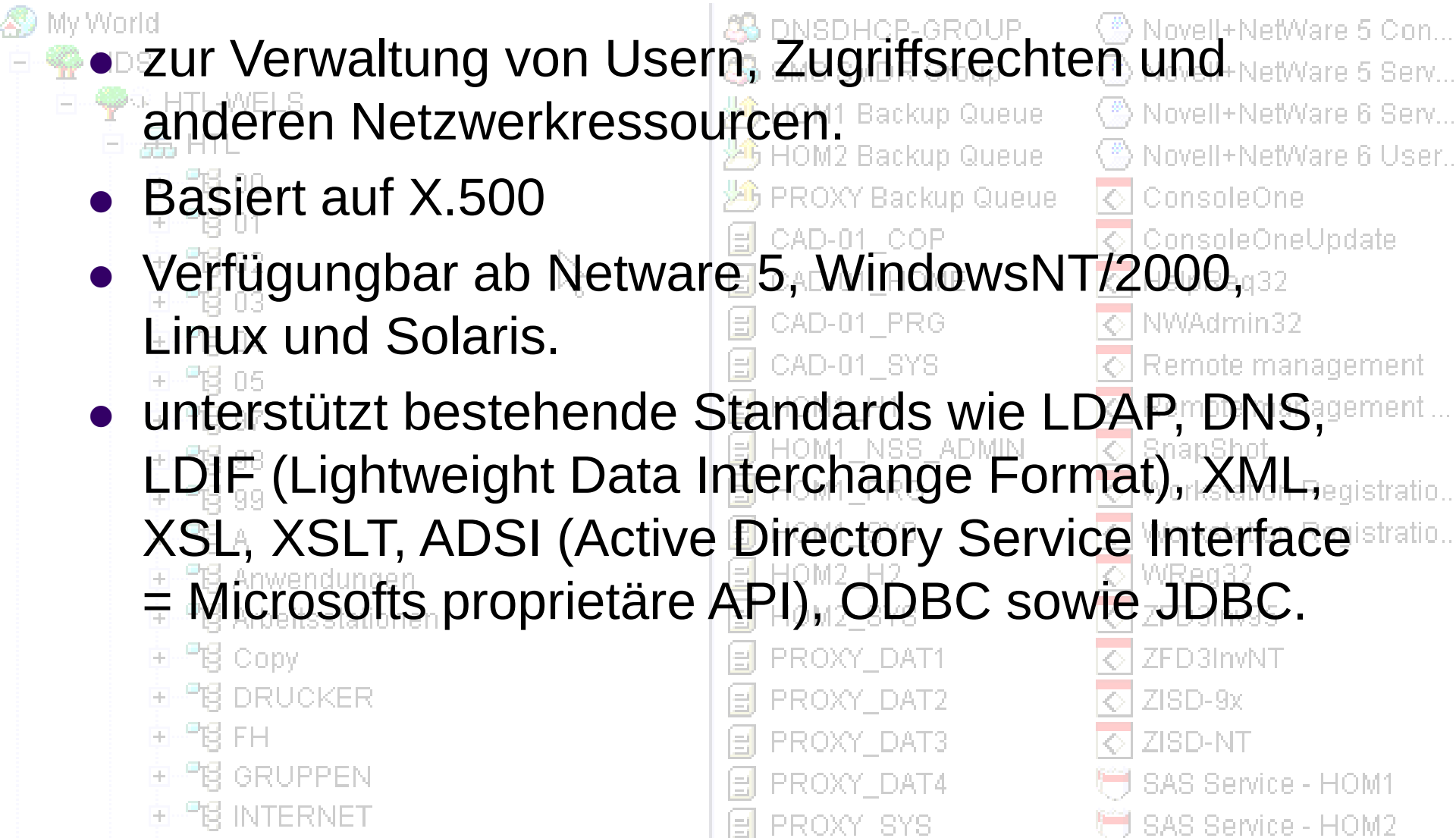
- Beim Einsatz mehrerer Server in einem Teilbaum werden die Daten repliziert. Es gibt einen Master-Server, auf dem der Datenbestand erstellt und gepflegt wird, und einen oder mehrere Slave-Server, die eine Kopie des Datenbestands speichern. Die Slave-Server holen sich in regelmäßigen Abständen vom Master-Server den aktuellen Datenbestand.

X.500: Protokolle

Die Spezifikation von X.500 definiert verschiedene Protokolle für die Bereitstellung des Verzeichnisdienstes. Diese basieren auf den sieben Schichten des OSI-Schichtenmodells:

Protokoll	Beschreibung
Directory Access Protocol (DAP)	Definiert die Kommunikation zwischen Directory User Agent (DUA) und Directory System Agent (DSA).
Directory System Protocol (DSP)	Zuständig für die Kommunikation zwischen Directory System Agents (DSA).
Directory Information Shadowing Protocol (DISP)	Definiert den Austausch von Informationen zur Replikation des Datenbestands zwischen einem Master- und einem Slave-Server.
Directory Operational Binding Management Protocol (DOP)	Definiert den Austausch von administrativen Informationen zwischen zwei Directory System Agents (DSA).

Novell eDirectory
Novells Directory Service (NDS)



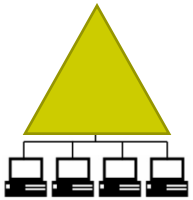
- zur Verwaltung von Usern, Zugriffsrechten und anderen Netzwerkressourcen.
- Basiert auf X.500
- Verfügungbar ab Netware 5, WindowsNT/2000, Linux und Solaris.
- unterstützt bestehende Standards wie LDAP, DNS, LDIF (Lightweight Data Interchange Format), XML, XSL, XSLT, ADSI (Active Directory Service Interface = Microsofts proprietäre API), ODBC sowie JDBC.

Microsoft Active Directory

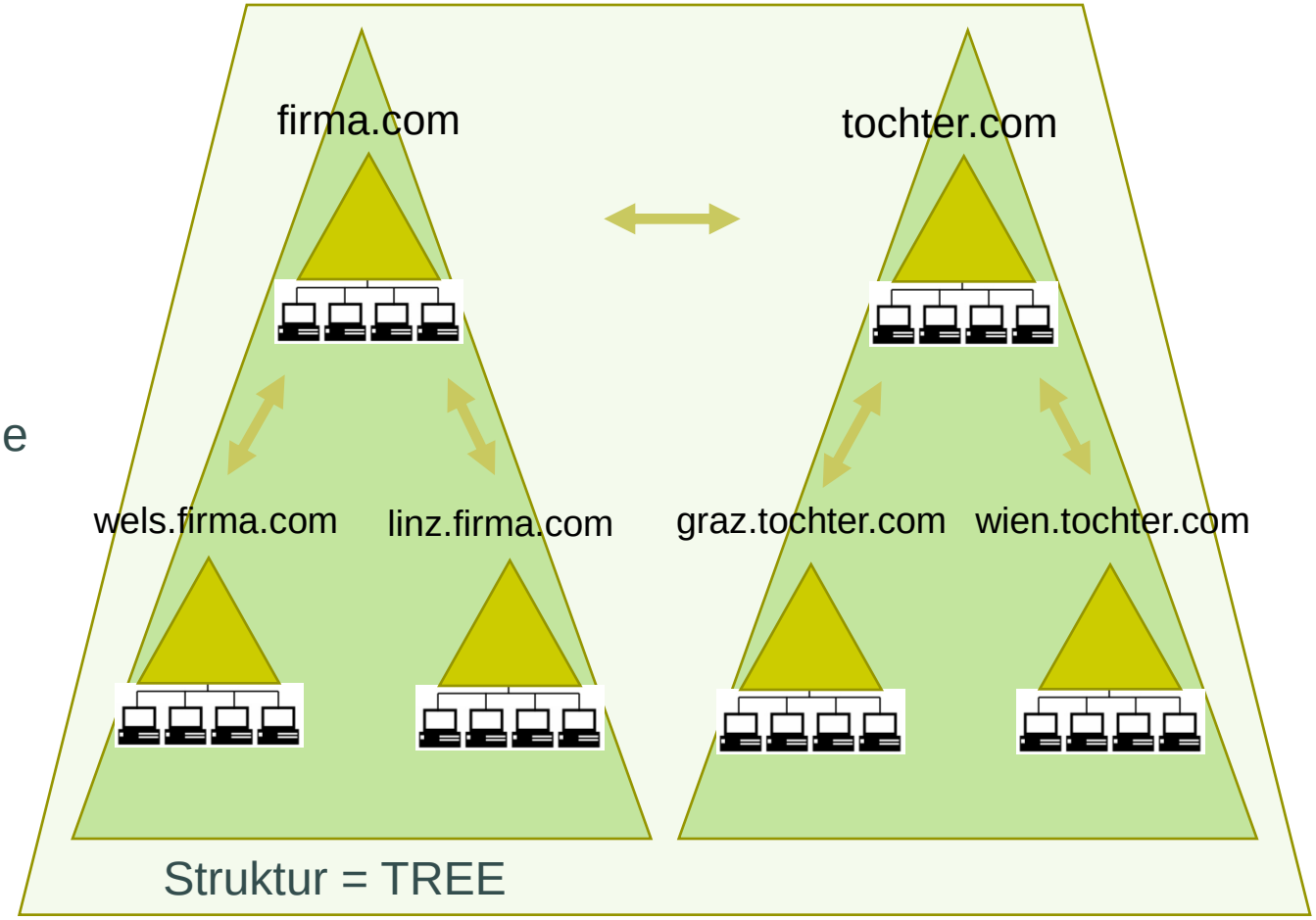
- Ab Windows 2000 / 2008
- X.500 konform
- Besteht heute aus mehreren Komponenten
  - **Active Directory Domain Services**
  - Active Directory Lightweight Directory Services
  - Active Directory Federation Services
  - Active Directory Rights Management Services
  - Active Directory Certificate Services
- Domänenstruktur sehr ähnlich wie DNS
- Benötigt unbedingt einen DNS Server

- https://de.wikipedia.org/wiki/Active_Directory

Domänenstrukturen



Einzelne Domäne



Gesamtstruktur = FOREST

Aufbau und Objekte des Active Directory

- Der Aufbau der Domänenstruktur des Active Directory ist vergleichbar (verknüpft) mit dem Konzept des DNS.
- Benutzernamen ähnlich Internet

  z.B.: elsi@schule.local
- **Logische Elemente von Active Directory**
  - **Objekte:** User, Computer, Drucker, Richtlinien,
  - **Objektattribute:** siehe Übung http://www.selfadsi.de/user-attributes.htm
  - https://docs.microsoft.com/de-de/windows/win32/adschema/active-directory-schema?redirectedfrom=MSDN
- **Strukturelle Komponenten von Active Directory**
  - **Domäne:** stellt eine Sicherheitsgrenze in einem einzelnen Computernetzwerk dar. Vertrauensstellungen zwischen den Domänen.
  - **Organisationseinheiten:** OU zur weitere Unterteilung in der Domäne, sind Container Objekte die wieder andere Objekte enthalten
  - **Zugriffsrechte:** Mit Vererbung kann wie im Dateisystem auch unterbrochen werden.

Jeder Computer in der Domaine ist auch irgendwie ein Benutzer.

LDAP
(Lightweight Directory Access Protocol)

- 1993 erstmals in RFC 1487 definiert
- Vereinfachte Version des X.500 DAP
- Aktuelle Version 3 in RFC 2251
- Es enthält die Beschreibung eines kompletten Protokolls, um mit TCP/IP-basierten Clients über einen vermittelnden LDAP-Server auf ein X.500-Verzeichnis zuzugreifen.
- Das Lightweight Directory Access Protocol ist zwar durch internationale RFCs definiert, jedoch noch kein offizieller Standard. Dennoch kann man bei LDAP von einem De-facto-Standard sprechen.

LDAP:
Protokoll oder Verzeichnisdienst?

- Grundsätzlich definiert LDAP ein Kommunikationsprotokoll. Es werden der Transport und das Format von Nachrichten definiert, die von einem Client für den Zugriff auf einen X.500-konformen Verzeichnisdienst verwendet werden.

- Anstatt eines LDAP-Gateways und der Umweg über einen X.500-Server kann der LDAP-Server auch direkt auf das Verzeichnis zugreifen. Man bezeichnet diese Konstellation auch als

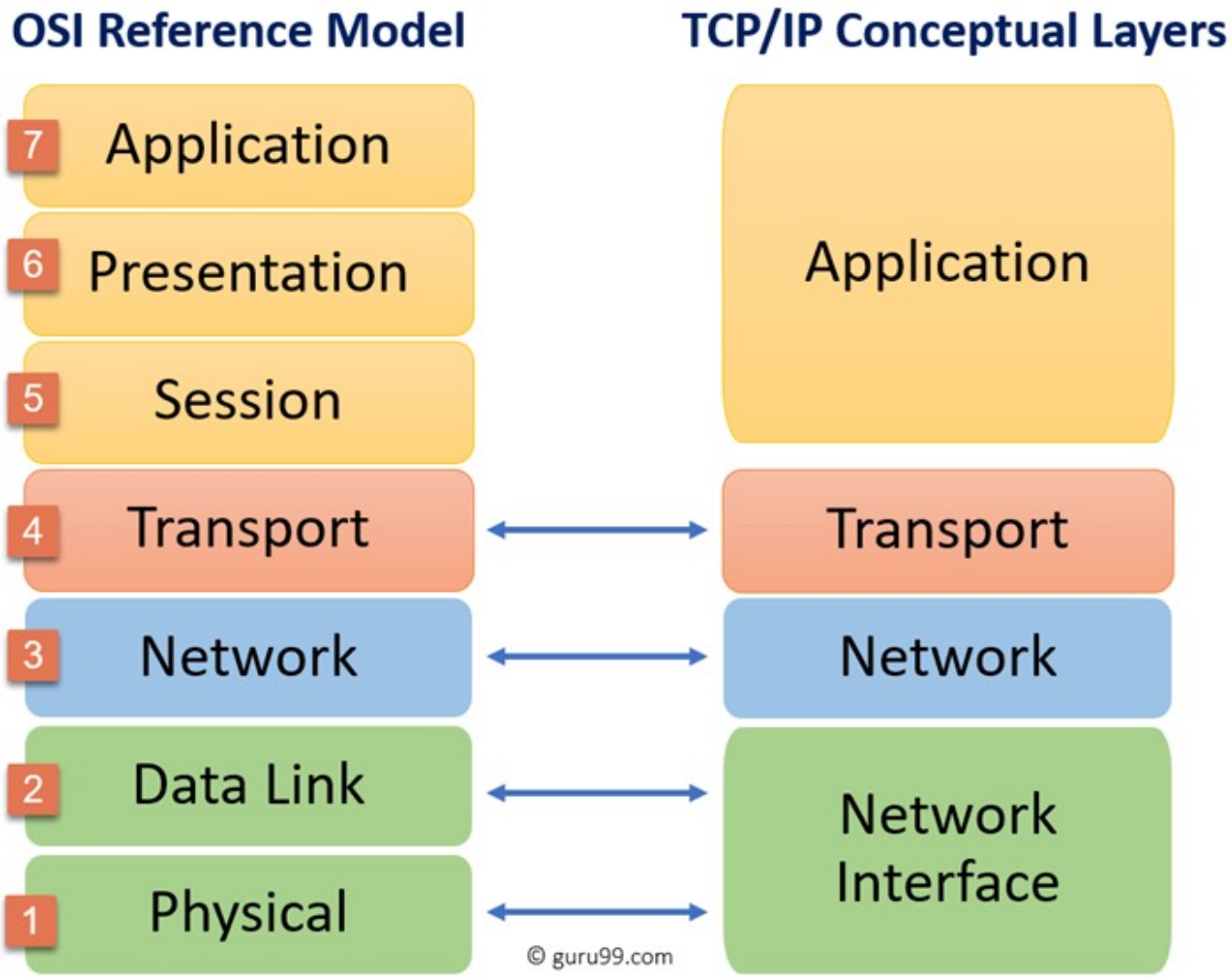
  Stand-alone-LDAP-Server

- Für den LDAP-Client spielt es keine Rolle, ob der LDAP-Server direkt auf das Verzeichnis zugreift oder als LDAP-Gateway fungiert.
- Greift der LDAP-Server direkt auf das Verzeichnis zu, kann man von einem LDAP-Verzeichnisdienst sprechen. Die Architektur ist ein Client-/Server-Modell das mit seiner Einfachheit gegenüber X.500 eine einfachere Realisierung ermöglicht.

LDAP Zusammenfassung

- Verbreiteter Verzeichnisdienst für beliebige Daten
- Für Unix, Linux und auch für Microsoft Windows verfügbar, Zugriff auch auf NetWare eDirectory
- Vereinfachte Version von X.500
  - X.500 baut auf OSI-Stack auf
  - LDAP baut hingegen auf dem TCP/IP-Stack auf
  - Geringerer Funktionsumfang
  - Leichter zu implementieren
  - Schneller
- Implementierungen
  - LDAPv1: 1991, Universität Michigan
  - LDAPv2: 1993
  - LDAPv3: 1997 – 2000
  - OpenLDAP: Open Source (www.openldap.org)

Der Standardport ist:
389 für ungesicherte sowie nur mit STARTTLS gesicherte Verbindungen
636 für mit TLS gesicherte Verbindungen (LDAPS)


OSI Reference Model
TCP/IP Conceptual Layers
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
Application
Transport
Network
Network Interface
© guru99.com

LDIF (*LDAP Data Interchange Format*)

- Format zum Datenaustausch mit LDAP
- Einfache Textdatei im ASCII-Format
- Besteht aus Attributtyp-Attributwert-Paaren
- Einzelne Datensätze werden durch Leerzeilen getrennt
- Das Zeichen # dient zum Einleiten von Kommentaren.

Beispiel LDIF-Datei

Benutzer im AD (nicht vollständig)
dn: CN=user1 syt5, OU=GruppeA, OU=SYT5, DC=sytdom,DC=local
displayName: user1 syt5
givenName: user1
sAMAccountType: 805306368
primaryGroupID: 513
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
badPasswordTime: 0
cn: user1 syt5
userAccountControl: 512
userPrincipalName: syt5user1@sytdom.local
pwdLastSet: 131870772356045235
sn: syt5
name: user1 syt5

Beispiel LDIF-Datei

```
# erstellen eines Linux Benutzers
version: 1
dn: cn=myuser1,ou=myou,o=htl
changetype: modify
add: objectClass
objectClass: posixAccount
-
add: loginShell
loginShell: /bin/bash
-
add: homeDirectory
homeDirectory: myuser1
-
add: gidNumber
gidNumber: 100
-
add: uidNumber
uidNumber: 1101
```

Rechte und Vererbung im ADS