# Verzeichnisdienste für Hochschulen auf Open Source Grundlage

Peter Gietz, DAASI International GmbH
Peter.gietz@daasi.de

Ausarbeitung eines Vortrags gehalten beim Workshop Informations- und Verzeichnisdienste in Hochschulen, Heinrich-Heine-Universität Düsseldorf, 11.10.2002

## Inhalt:

1. Einführung in Verzeichnisdienste und LDAP ..... 1
1.1. Definition des Begriffs "Verzeichnisdienst" ..... 1
1.2. Das Konzept von X.500/LDAP ..... 2
1.2.1. Das Informationsmodell ..... 2
1.2.2. Das Funktionsmodell ..... 4
1.2.3. Die Standardisierung von LDAP ..... 5
1.3. Die Open Source Implementierung OpenLDAP ..... 6
2. Anwendungsmöglichkeiten von LDAP ..... 6
2.1. Kontaktdateninformationsdienste und daran angefügte Dienste ..... 7
2.3. Metadirectory ..... 8
2.4. Zertifikatsserver für PKI ..... 8
2.5. Weitere Anwendungsmöglichkeiten ..... 9
3. LDAP in nationalem und internationalem Forschungsumfeld ..... 10
3.1. Internationale Kooperationen ..... 10
3.1.1. TERENA ..... 10
3.1.2. Das Middlewareprojekt im Rahmen von Internet2 ..... 10
3.2. LDAP im DFN Umfeld ..... 11
4. Referenzen ..... 12

# 1. Einführung in Verzeichnisdienste und LDAP

## *1.1. Definition des Begriffs "Verzeichnisdienst"*

Wenn man den Begriff Verzeichnisdienst (engl. *directory*) im Rahmen der EDV definieren möchte, so kann man zu einer allgemeinen und einer spezielleren Definition gelangen:

1.) Verzeichnisdienst, ist ein elektronischer Dienst, in welchem Daten in einer hierarchischen Struktur zur Verfügung gestellt werden.

Unter dieser allgemeinen Definition sind also sowohl Dateiverzeichnisse im Betriebssystem wie etwa MS/DOS oder Unix inbegriffen, die Informationen über Dateilänge, Erstellungsdatum der Datei, etc. liefern, wie auch der Domain Name Service (DNS), einer weltweit verteilten Datenbank, mittels derer man die Zugehörigkeit von Domain-Namen und IP-Adressen abbildet, oder auch der Whois-Dienst, mittels dessen Domain-Besitzer recherchiert werden können..

In diesem Beitrag soll aber die engere Definition gelten, nämlich:

2.) Verzeichnisdienst, ist ein spezialisierter elektronischer Dienst, der sich in seiner Technologie vom internationalen ITU/ISO-Standard X.500/9594 [X.500] ableitet und sich insbesondere durch ein bestimmtes objektorientiertes Datenmodell und ein Netzwerkprotokoll auszeichnet.

X.500[1] wurde von den nationalen Telefongesellschaften im Rahmen OSI-Modells (Open System Interconnection) definiert, insbesondere um ein international standardisiertes elektronisches Telefonbuch zu realisieren. Es wurde aber so allgemein und erweiterbar spezifiziert, dass sich mit diesem Standard beliebige Netzinformationsdienste realisieren lassen. X.500 wurde im Laufe der Zeit in bisher 4 Versionen weiterentwickelt (V1: 1988, V2:1993, V3: 1997 und V4: 2001). Im Rahmen der IETF[2] wurde der X.500-Standard in einer "light"-Version namens LDAP (Lightweight Directory Acces Protocol) standardisiert , wobei es v.a. darum ging, die komplexen Vorbedingungen der OSI-Welt, wie das Sieben-Schichten-Modell zu streichen, sowie eine ganze X.500-Protokoll-Suite auf ein einziges Protokoll zu begrenzen[3]. Neben den genuinen X.500 Implementierungen wie z.B. von Siemens (DirX) oder Isode (M-Vault) und den genuinen LDAP-Implementierungen, wie z.B. das Open-Source-Projekt OpenLDAP[4] oder die Implementierungen von Netscape (Netscape Directory Server) lehnen sich weitere Produkte mehr oder weniger strikt an den X.500-Standard an, wie z.B. Novell Directory Service (NDS) und - mit einigen Einschränkungen - Microsoft Active Directory (AD), welches seit Windows 2000 elementarer Bestandteil der Microsoft Betriebssysteme ist und die Benutzer- und Ressourcenverwaltung realisiert.

## *1.2. Das Konzept von X.500/LDAP*

Im Folgenden werden die Grundzüge des Daten- und Operations-Modells von LDAP beschrieben, welche größtenteils mit X.500 identisch sind.

### 1.2.1. Das Informationsmodell

X.500/LDAP ist also eine Datenbank mit einer hierarchischen Datenstruktur. Diese Datenbank ist für schnelles lesen optimiert und hat nur einfache Updatemechanismen, Einzeloperationen übergreifende Mechanismen wie Transaktionen werden im Standard bisher nicht spezifiziert, allerdings augenblicklich im Rahmen von LDAP angedacht [Transactions] und teilweise auch schon implementiert[5]. Die Daten können ähnlich wie das WWW beliebig im Netz auf verschiedene Server, sog. *Directory Service Agents* (DSA) verteilt werden. Zusätzlich ist eine redundante Spiegelung der Daten auf verschiedenen Servern möglich (*Replication* oder *Shadowing*). Auf die Daten wird über ein wohldefiniertes Netzwerkprotokoll zugegriffen werden. Beliebige Daten können gespeichert werden, also neben alphanumerische Daten, Namen, Adressen, Beschreibungen, Zahlen, etc. auch Zeiger auf andere Daten (sowohl Zeiger, die auf Bereiche innerhalb des eigenen hierarchischen Datenbaums verweisen, als auch Zeiger auf externe Daten, wie URIs oder Dateinamen). Darüber hinaus können aber auch Zertifikate im Rahmen einer PKI, so wie andere Binärdaten wie Grafiken, Photos, Diagramme, Audio-Information etc. gespeichert werden. Beim Datenmodell handelt es sich um ein offenes, also erweiterbares Modell für beliebige Daten.

Die hierarchische Baumstruktur zum Speichern der Daten wird Directory Information Tree (DIT) genannt. Daten werden in Knoten dieses Baumes den sogenannten Einträgen

[1] Immer noch eine der besten Einführungen in X.500 gibt [Chadwick].
[2] Vgl. http://www.ietf.org.
[3] Weiteres zu LDAP siehe weiter unten.
[4] Vgl. http://www.openldap.org.
[5] Hier sei insbesondere auf das neue transaktionsorientierte OpenLDAP-Datenbackend auf Basis von Berkeley DB verwiesen.

gespeichert. Jeder Knoten hat 0 bis n Kinderknoten aber nur genau 1 Elternknoten (mit Ausnahme des Wurzelknotens, welcher keinen übergeordneten Knoten hat).

Jeder Eintrag hat einen eindeutigen Namen. Innerhalb der eigenen Hierarchieebene wird der Name eines Eintrags Relative Distinguished Name (RDN) genannt. Keine zwei Geschwistereinträge (also mit gemeinsamen Elternknoten) dürfen den gleichen RDN haben. Alle RDNs auf dem Pfad von der Wurzel zum Eintrag bilden zusammen den Distinguished Name (DN) durch welchen jeder Eintrag einen im gesamten Baum eindeutigen Namen hat (vgl. Abb.1).

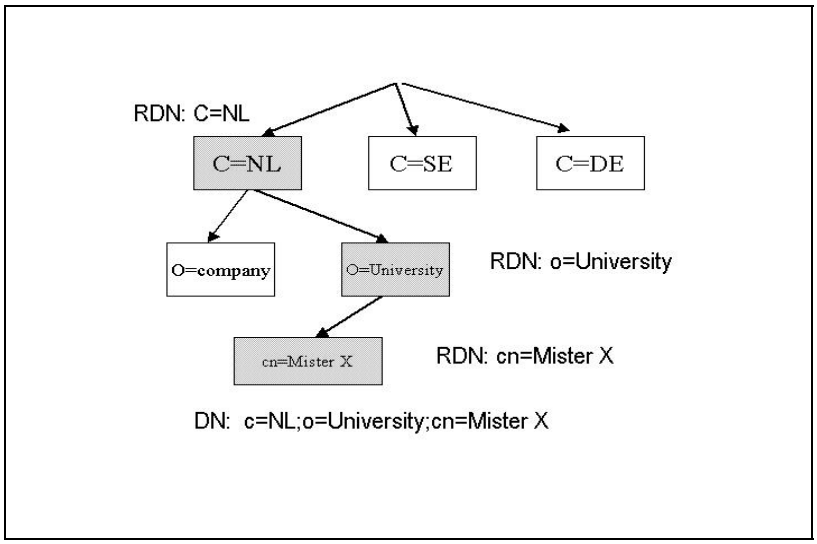


Abb. 1: Der *Directory Information Tree* und sein Namensraum

Ein Eintrag besteht aus sogenannten Attributen, welche die eigentlichen Informationsspeicher sind. Ein Attribut besteht aus einem Attributtyp und einem bzw. mehreren Attributwerten, abhängig davon, ob der Attributtyp als *single valued* oder *multi valued* definiert wurde. Durch den Attributtyp wird darüber hinaus die zugehörige Attributsyntax spezifiziert, welcher die Attributwerte zu folgen haben. Zusätzlich können im Attributtyp verschiedene Vergleichsregeln, sog. *Matching Rules* spezifiziert werden für einen String-Vergleich (*Equality Matching Rule)*, für einen Teilstring-Vergleich (*Substring Matching Rule*) sowie für die Sortierungsreihenfolge (*Ordering Matching Rule*). Darüberhinaus lassen sich selbstdefinierte *Matching Rules* erstellen (*Extensible Matching Rule*)

Alle Informationen werden, wie erwähnt in Attributwerten der Attributtypen abgelegt. Neben allgemeinen Attributtypen gibt es hierbei einige spezielle. Mit einem oder mehreren Attribut-Typ-Wert-Paare wird der RDN gebildet, diese Attribute werden *Naming Attribute* oder *Distinguished Attribute* genannt. Darüber hinaus muss jeder Eintrag mindestens ein sog. Objektklassen-Attribut haben, welches den gesamten Eintrag charakterisiert und einen Satz zu verwendender Attributtypen spezifiziert, wobei zwischen optionalen Attributen (*May*) und Pflichtattributen (*must*) unterschieden wird. Objektklassen können Attributtypen von übergeordneten Objektklassen erben. Als Wurzel dieser Vererbungshierarchie dient eine sog. abstrakte Objektklasse, meistens die Objektklasse *top.*

Eine Ansammlung von Objektklassen, Attributen, Attributsyntaxen und Vergleichsregeln, die für einen bestimmten Zweck definiert wurden, werden *Schema* genannt. Der Standard definiert Schema für Personen, Organisationen, Applikationen etc. Darüber hinaus können eigene Schemata definiert werden. Lokal kann man selbstdefiniertes Schema einfach

verwenden, wenn das Schema global genutzt werden soll muss man es standardisieren (als IETF Request For Comments, RFC) oder wenigstens registrieren.

Der Datenaustausch wird über ein Format namens LDIF (*LDAP Data Interchange Format*) [Good] realisiert. Es handelt sich hierbei um ein einfaches ASCII-Format, welches im Wesentlichen aus Attributtyp-Attributwert-Paaren besteht. Einzelne Datensätze werden durch eine Leerzeile voneinander getrennt. Das Zeichen # dient zum Einleiten von Kommentaren. Im folgenden Beispiel wird ein Landeseintrag, ein Organisations- und ein Personendatensatz beschrieben:

```
# Definition des Länderknotens für Deutschland
dn: c=DE
objectclass: top
objectclass: country
# Die Objektklasse country hat als Must-Attribut countryName, abgekürzt c
# Dieses Attribut wird gleichzeitig als namensgebendes Attribut verwendet (RDN)
c: DE

# Definition eines Organisationsknotens
dn: o=Universität Düsseldorf, c=DE
objectclass: organization
# Das einzige Must-Attribut organizationName, abgekürzt o, ist multi valued.
# Es dient hier einmal als namensgebendes SAttribut. Zum anderen wird
# eine weitere Formen des Organisationsnamens angegeben
o: Universität Düsseldorf
o: Heinrich Heine Universität Düsseldorf

# Definition eines Personeneintrags
dn: cn=Manfred Mustermann, o=Universität Düsseldorf, c=DE
objectclass: top
objectclass: person
# Eine zweite von person abgeleitete Objektklasse organizationalPerson fügt
# weitere may-Attributtypen zum Eintrag hinzu
objectclass: organizationalPerson
# Das namensgebende Attribut ist hier commonName, abgekürzt cn.
cn: Manfred Mustermann
cn: Manni Mustermann
mail: manni@dot.com
mail: mannfred.mustermann@uni-duesseldorf.de
telephoneNumber: 1234567
```

Abb. 2: Beispiel für eine LDIF-Datei

### 1.2.2. Das Funktionsmodell

Das Netzwerkprotokoll spezifiziert verschiedene Operationen, nämlich:

- Authentifizierungs-Operationen:
  - *bind* zum Erstellen einer Client-Server-Verbindung)
  - unbind oder *abandon* zum Beenden bzw. Abbruch einer solchen Verbindung
- Abfrage-Operationen:
  - *search* zum Suchen von Daten
  - *compare* zum Vergleich von Daten
- Update-Operationen:
  - *add* zum Hinzufügen von Einträgen
  - *delete* zum Löschen von Einträgen
  - *modify* zum Ändern von Attributwerten eines Eintrags
  - *modifyDN* zum Umbennenen eines Eintrags, also zur Änderung des Ortes im hierarchischen Baum.

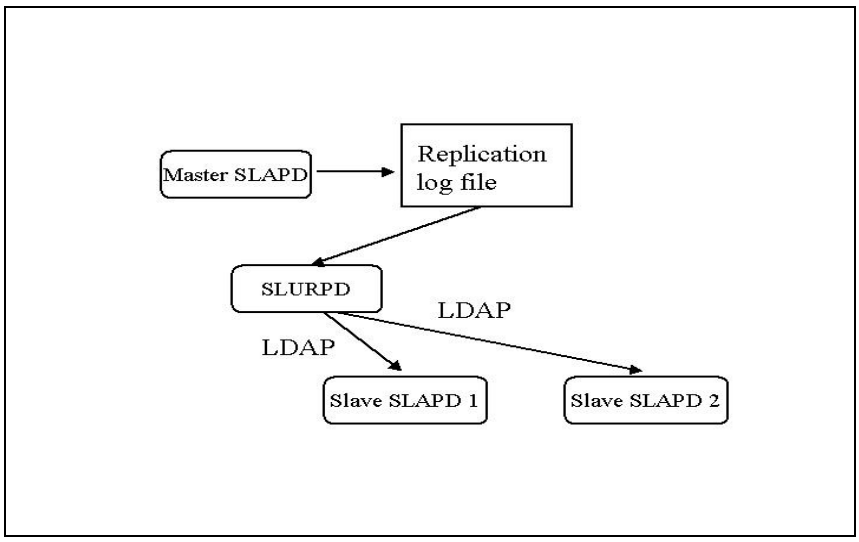
Zur Authentifizierung dienen verschiedene Mechanismen. Beim sog. *simple bind* wird entweder eine anonyme Verbindung oder eine authentifizierte Verbindung erstellt, indem ein Eintrag durch den DN spezifiziert wird und die Zugehörigkeit zu diesem Eintrag mittels eines Passworts, das dort in einem hierfür verwendeten Attributs gespeichert ist, bewiesen wird. Das Passwort geht hierbei jedoch ungeschützt über das Netz. Dies kann verhindert werden, indem zuvor eine Verschlüsselung initiiert wird mittels TLS (*Transport Layer Security*) [Dierks] welches die IETF-Version von SSL (*Secure Socket Layer*) ist. Darüberhinaus gibt es weitere Authentifizierungsmechanismen, die unter dem Begriff SASL (*Simple Authentication and Security Layer*) zusammengefasst sind [Myers].

### 1.2.3. Die Standardisierung von LDAP

Die aktuelle Version des IETF-Standards LDAP ist LDAPv3, welches in [Hodges] und in den dort referenzierten Texten spezifiziert wird. Obwohl LDAP ursprünglich – wie noch aus dem Namen *Lightweight Directory Access Protocol* ersichtlich – ein reines Zugriffsprotokoll zwischen Clients und X.500-Servern war, handelt es sich in der Version 3 um ein vollständiges Client-Server-System, mit genuinen LDAP-Servern. Neben den oben kurz zusammengefassten Informationsmodell, Namensraum, Netzwerkprotokoll, und sicheren Authentifizierungs- und Verschlüsselungsmechanismern, definiert der Standard, ein [6]Referierungsmodell (*Referral*), Erweiterungsmechanismen für Operationen und weitere Authentifizierungsverfahren, und eine LDAP URL.

Darüber hinaus wurde das Datenaustauschformat (LDIF) standardisiert, sowie – als Draft zwar nur de facto, jedoch in allen LDAP-Entwicklungsumgebungen implementiert - APIs (*Application Programming Interface*) für die Programmiersprachen C und Java.

Augenblicklich in Arbeit ist ein LDAP Client Update Protocol [Megginson], welches auch für Server-Server-Replikation verwendet werden kann. Bemühungen, einen darüber hinausgehenden Multi-Master-Replikationsstandard zu spezifizieren müssen vorerst als gescheitert betrachtet werden. Allerdings gibt es neben dem erwähnten LCUP auch noch mehrere Implementierungen des sog. SlurpD, einem Daemon, der auf Grundlage einer Replikations-Log-Datei, die Replikation der dort beschriebenen Daten auf mit Standard-LDAP-Mitteln durchführen kann (vgl. Abbildung 3). Schließlich wird versucht, LDAP-Zugriffskontrolle zu standardisieren [Stokes].




[6] Vgl. http://www.ietf.org/html.charters/ldup-charter.html.

Abb. 3: Beispielkonfiguration für Slurpd-basierende Replikation

Diese Standardisierungsbemühungen haben dazu geführt, dass eine große Anzahl von Programmen, eine LDAP-Schnittstelle haben. So sprechen alle heutigen Verzeichnisdienst-Implementierungen, also Alle X.500(93) Implementierungen, Novell Directory Service (NDS), Microsoft Active Directory (AD) LDAP, sowie viele Clientanwendungen, wie Mailagenten (für Emailadressenrecherche), Browser (über die LDAP-URL), Verschlüsselungsprogramme, etc. LDAP wird in vielen Standardimplementierungen als Authentifizierungsschnittstelle berücksichtigt, wie z.B. bei IMAP, SMTP Auth, etc. Auch der Apache Webserver verfügt über eine LDAP-Schnittstelle.

### *1.3. Die Open Source Implementierung OpenLDAP*

Mit dem von IBM finanzierten Open Source Projekt OpenLDAP steht mittlerweile ein vollständig LDAPv3 kompatible LDAP-Implementierung im Source Code zur Verfügung. Es handelt sich hierbei um Implementierungen eines LDAP Servers (slapd), des bereits erwähnten Replikations-Servers (slurpd), sowie eine reihe von Client-Anwendungen und einer Bibliothek zum Entwickeln eigener Clients. Der von einem internationalem Entwicklerteam unterstützte Hauptentwickler Kurt Zeilenga ist sehr stark an den IETF-Standardisierungsbemühungen beteiligt, sodass neue Standards relativ zeitnah in OpenLDAP implementieret werden. Diese Software wird in vielen Projekten im Produktionsbetrieb eingesetzt, sowohl im Forschungsbereich, als auch in kommerziellen Unternehmen.

OpenLDAP zeichnet sich durch Stabilität und relativ hoher Performanz aus. Es besitzt gute Zugriffskontrollmechanismen, die vom Authentifizierungsgrad, aber auch z.B. von bestimmten IP-Adressen abhängig gemacht werden können. Mit diesen Mechanismen lässt sich eine hohe Datensicherheit erzielen. Mit dem stabilen Replikationsmechanismus lassen sich die Daten auf beliebig vielen Rechnern spiegeln, wodurch hohe Zugriffsgeschwindigkeiten und Ausfallssicherheit gewährleistet werden können. Die Verteilung der Daten auf mehrere Server ist ebenfalls möglich, wodurch eine hohe Skalierbarkeit in Bezug auf Datenmengen erreicht werden kann.

Die objektorientierte Datenmodellierung ermöglicht beliebige Erweiterungsmöglichkeiten. Die Kompatibilität zum offenen Standard bewirkt eine Unabhängigkeit von Herstellern. Die unter OpenLDAP gespeicherten Daten lassen sich einfach auf andere LDAP-Implementierungen portieren. Die Daten sind über TCP/IP basiertes Netzwerkprotokoll, also sowohl in einem darauf basierenden Intranet, als auch wenn gewünscht im Internet zugänglich.

## 2. Anwendungsmöglichkeiten von LDAP

Ein wesentlicher Vorteil von LDAP ist, dass die selben Daten von verschiedenen Anwendungen verwendet werden können. So kann also ein LDAP-Server für verschiedene Client-Anwendungen Daten vorhalten, aber auch von allen gemeinsam genutzte Daten. Mit dem hierarchischen Datenmodell lassen sich viele in Organisationen und Unternehmen relevante Wirklichkeiten abbilden, wie z.B., Organisationsstruktur, Zugehörigkeiten zu verschiedenen Gruppen, Ressourcen, Lokalitäten, etc. Im Folgenden werden einige dieser Anwendungsmöglichkeiten beschrieben.

## 2.1. Kontaktdateninformationsdienste und daran angefügte Dienste

Ein Verzeichnis von Kontaktdaten, wie z.B. ein Telefonbuch ist die klassische Anwendung, die schon die Erfinder des X.500 (ITU) bei der Spezifikation im Auge hatten. Deswegen ist entsprechendes Schema bereits im Standard definiert, sodass sich ohne eigene Datenmodellierung Dienste für Personendaten (White Pages) und Organisationsdaten (Yellow Pages), oder eine Kombination dieser beiden realisieren lassen. Die Organisationsstruktur lässt sich beliebig tief abbilden, zum Beispiel in folgenden Ebenen:

organization= Universität
organizationalUnit= Mathematisch-Naturwissenschaftliche Fakultät
organizationalUnit= Wissenschaftliche Einrichtung Physik
organizationalUnit= Institut für Angewandte Physik
organizationalUnit= Arbeitsgebiet Materialforschung

Unter jedem dieser Knoten ließen sich Einträge für Personen, für Räumlichkeiten, oder für Geräte, wie Computer, Messgeräte, Drucker etc. einfügen.

Mit einer solchen Struktur lassen sich ein elektronisches Telefonbuch, ein elektronisches Emailverzeichnis realisiere, aber auch z.B. ein elektronisches Vorlesungsverzeichnis.

Auch eine Computer-Benutzerverwaltung lässt sich hinzufügen, wobei man für Unix-Benutzer auf bereits standardisierte LDAP Objektklassen zur Abbildung vom NIS (*Network Information System*) zurückgreifen kann [Howard], mittels dessen man User (/etc/passwd und shadow file), Groups (/etc/groups), IP services (/etc/services), IP protocols (/etc/protocols), RPCs (/etc/rpc), IP hosts and networks, NIS network groups and maps, MAC addresses und Boot information abbilden kann. Ähnliche Schemata wurden auch für Windows-Active-Directory definiert.

## 2.2. Authentifizierungsdienste

In größeren Organisationen, wie Universitäten und Fachhochschulen wird die Benutzerverwaltung und die Rechte-Vergabe für Computerzugriff immer unübersichtlicher. Benutzer haben Zugriff auf viele Rechner, und auf jedem dieser Rechner muss für jeden Benutzer eine eigene LoginID und ein Passwort verwaltet werden. Der Benutzer muss sich viele Passwörter merken, der Administrator hat einen sehr hohen Verwaltungsaufwand.

Als Lösung zu diesem Problem bietet sich ein zentraler verzeichnisdienstbasierter Authentifizierungsdienst an, wodurch jeder Benutzer nur noch ein einziges Passwort für alle Rechner haben kann. Es steht ihm allerdings weiterhin frei, auf bestimmten Rechnern durch die lokalen Authentifizierungsmechanismen zuzugreifen.

Unix-Clients können in einem solchen zentralen Authentifizierungsdienst mittels NSS / PAM-LDAP direkt auf LDAP-Server zugreifen, wie in z.B. in [Klasen] beschrieben wird. Auch die Anbindung an MS Active Directory (AD) ist möglich wodurch auch Windows-Clients in einen solchen Dienst integriert werden können. Hierbei dient entweder das Active Directory als der zentrale Verzeichnisdienst, oder aber die Windows-Clients werden über SAMBA an einen LDAP-Server angebunden.

Mit einem solchen Authentifizierungsdienst lässt sich nicht nur das Login realisieren, sondern er kann auch in verschiedene Netzanwendungen integriert werden, z.B. IMAP, POP, SMTP auth, FTP, HTTP auth, RSH, SSH, etc. etc. Dies ließe sich mit Kerberos realisieren, sodass nach einem einmal ausgestellten Kerberos-Ticket, alle auf alle diese Dienste ohne weitere

Passworteingabe, aber dennoch authentifiziert zugegriffen werden kann. Eine solche Lösung wird deshalb auch *Single Sign On* (SSO) genannt.

Allerdings sei hier nicht unerwähnt, dass ein solcher zentraler Dienst auch Nachteile in sich birgt: Ein Passwort für alle Rechner bedeutet auch einen *Single point of failure,* fällt also der Authentifizierungsdienst aus, funktioniert gar nichts mehr. Auch entsteht ein wesentlich größerer Schaden, wenn das eine Passwort kompromittiert wurde. Allerdings lassen sich sensible Dienste ja weiterhin mit dezentraler Authentifizierung nutzen. Ist eine solche als Rückfallmöglichkeit auf allen Rechnern und Diensten immer noch vorhanden, würde sich der Ausfall des zentralen Dienstes nur durch eine längere Wartezeit auf einen *Timeout* bemerkbar machen.

## 2.3. Metadirectory

Ein weiteres Problem, welches sich in vielen größeren Organisationen stellt, ist die Tatsache, dass gleiche Daten in verschiedenen Datenbanken, die wiederum auf verschiedenen Technologien beruhen, gepflegt werden. So wird eine Universität in der Regel eigene Datenbanken haben für die Emailbenutzerdaten, die Personaldaten und Telefondaten. In all diesen Datenbanken, werden meist von verschiedenen Administratoren die gleichen Daten verwaltet, wie Name, Abteilungszugehörigkeit, Telefonnummer, Raumnummer, etc. Aus organisatorischen Gründen, oder aus Gründen des Datenschutzes lassen sich aber diese Datenbanken nicht zu einer einzigen verschmelzen.

Hier bietet sich eine sog. Metadirectory-Lösung an, bei der durch einen Verzeichnisdienst die verschiedenen Datenbanken synchronisiert werden. Die gleichen Daten müssen nur einmal in eine der Datenbanken eingegeben, bzw. gepflegt werden. In den durch das Metadirectory verknüpften Datenbanken werden sie automatisch angelegt bzw. geändert. Außerdem ermöglicht das Metadirectory eine übergreifende Sicht auf alle Daten. Dennoch bleiben die einzelnen Prozesse flexibel an Organisationsabläufe und Datenschutzvorkehrungen anpassbar.

Voraussetzung für ein Metadirectory sind sogenannte Konnektoren, die die Synchronisierung gewährleisten. Zwar gibt es bereits kommerzielle Standardlösungen, die Konnektoren für gängige Datenbanken zur Verfügung stellen. In vielen Fällen wird es jedoch notwendig sein, individuell an die eigene Datenbanklandschaft angepasste Konnektoren zu entwickeln. Hier wird eine Open-Source-basierte Directory-Lösung wesentlich flexibler sein, da sie besser anpassbar ist.

## 2.4. Zertifikatsserver für PKI

PKI (*Public Key Infrastructure)* ist ein auf asymmetrischer Verschlüsselungstechnik basierender Dienst, der sichere Authentifizierung und digitale Signatur ermöglicht. Wesentlich hierbei ist, dass im Gegensatz zur symmetrischen Verschlüsselung, wo nur ein Schlüssel benötigt wird, ein Schlüsselpaar erzeugt werden muss, das sich in einen geheimen, privaten Schlüssel und einen öffentlichen Schlüssel aufteilt, die in einer mathematischen Relation stehen, die bewirkt, dass man vom privaten Schlüssel auf den öffentlichen schließen kann, jedoch nicht umgekehrt vom öffentlichen auf den privaten. Mit dem öffentlichen Schlüssel kann man also etwas verschlüsseln, das nur mit dem Privatem Schlüssel entschlüsselt werden kann. Wird also ein solcher öffentlicher Schlüssel veröffentlicht, kann jeder ohne vorherige Kommunizierung von Schlüsseln für jemanden ein Dokument verschlüsseln, das nur mit dessen privaten Schlüssel entschlüsselt werden. Eine mit dem privaten Schlüssel generierte digitale Signatur kann von jedermann mittels des öffentlichen Schlüssels überprüft werden. Im sog. Zertifikat wird die zu einem öffentlichen Schlüssel

zugehörige Identität von einer vertrauenswürdigen Stelle (Certification Authority, CA) durch digitale Signatur bestätigt.

Verzeichnisdiensttechnologie hat sich als ideal erwiesen für die Veröffentlichung von Zertifikaten. So können auch Standardanwendungen (S/MIME und PGP) auf die Zertifikate zugreifen. Zusätzlich können im Verzeichnisdienst können zurückgerufene Zertifikate in sog. Certificate Revocation Lists (CRL) dokumentiert werden. Darüber hinaus kann ein Zertifikatsserver als Grundlage für einen Online Certificate Status Protocol (OCSP) Dienst dienen, der mittels eines sehr einfachen Protokolls den Status eines Zertifikats mitteilen kann.

Entweder betreibt eine CA den Verzeichnisdienst selber, oder liefert Zertifikate auf einem gesicherten Weg an den Betreiber. Neben einem bereits definiertem LDAP-Schema zur Ablage von Zertifikaten, gibt es augenblicklich eine IETF-Aktivität, ein Modell zu definieren, mittels dessen man auch viele Zertifikate einer Person so ablegen kann, dass Clienten mit einfachen Mitteln auf das eine gesuchte Zertifikat zugreifen kann [Gietz].

## 2.5. Weitere Anwendungsmöglichkeiten

Zusätzlich zu den oben erwähnten Anwendungsmöglichkeiten von LDAP-Technologien gibt es viele andere Bereiche, in denen diese Technologie sich sinnvoll einsetzen lässt. Um den Rahmen dieses Aufsatzes nicht zu sprengen, seien sie hier nur erwähnt:

- Verzeichnisdienste im Bereich Digital Libraries
  Auch Bücher und andere Text- und Multimedia-Ressourcen lassen sich mit LDAP verwalten. Hierbei können nicht nur die solche Ressourcen beschreibenden Metadaten verwaltet werden, sondern auch das kontrollierte Vokabular zur Verschlagwortung. Die hierarchische Struktur ist hierbei ideal, um Klassifikationssysteme abzubilden. Aber auch neuere Erschließungstechnologien, sog. Ontologien, also multidimensionale Klassifikationssysteme, in denen Begriffe und beliebig viele Relationen zwischen diesen Begriffen abgebildet werden, lassen sich mit LDAP-Technologien realisieren. Solche Technologien werden auch benötigt, um das WWW im Rahmen eines sog. *Semantic Web* besser zu erschließen,
- LDAP und Grid Computing
  Das Problem, welches sich in vielen Naturwissenschaften, z.B. in der Kernphysik und in der Metereologie stellt, ist es Petabytes (also Tausende von Tausenden von Gigabites) von Daten mit komplexen Algorithmen analysieren zu müssen. Hierzu werden gleichzeitig tausende CPUs und Festplatten benötigt. Grid Computing[7] will dieses Problem mit einer Infrastruktur lösen, deren Komplexität vor dem Benutzer versteckt wird in Analogie zum komplexen Stromnetz (Grid) welches sich dem Benutzer in einfachen Steckdosen darbietet. Die wichtigste Implementierung eines solchen Dienstes namens Globus[8] (www.globus.org) basiert teilweise auf OpenLDAP, namentlich das *Grid Information System* und das *Replika Management System*.
- LDAP und Netzwerkverwaltung
  Im Rahmen von dem sog. *Directory Enabled Networking*, werden Regeln nach denen die Netzkomponenten die Datenpakete weiterleiten (*Router Policy),* in einem Verzeichnisdienst abgelegt. Hier sei z.B. auf die Arbeiten der IPSec-Policy

[7] Vgl. http://www.gridforum.org.
[8] Vgl. http://www.globus.org.

Arbeitsgruppe der IETF verwiesen[9]. Eine Testimplementierung mit OpenLDAP wird von [Vollrath] beschrieben.

# 3. LDAP in nationalem und internationalem Forschungsumfeld

LDAP gewinnt zunehmend an Bedeutung im Universitären Umfeld Europas. Dieser Abschnitt versucht diese Entwicklung skizzenhaft zu dokumentieren.

## *3.1. Internationale Kooperationen*

### 3.1.1. TERENA

TERENA (*Trans-European-Research and Education Networking Association*)[10] ist eine europäische Vereinigung der Nationalen Forschungsnetze (DFN, SurfNet, etc.), welche sich zum Ziel gemacht hat, Forschung und Pilotierung von Netztechnologien und –Anwendungen zu fördern. Hierzu werden Konferenzen veranstaltet, Einzelprojekte gefördert, sowie Arbeitsgruppen, sog. *Task Forces* errichtet.

Auf dem Gebiet Verzeichnisdienste wurde die TF-LSD (Task Force LDAP Service Deployment) ins Leben gerufen[11], die Aktivitäten zu einem LDAP-Index-basierten Europäischen White-Pages-Verzeichnisdienst sowie zu Zertifikatsservern koordiniert.

Desweiteren unterstützt TERENA zwei Projekte der Firma DAASI International auf dem Gebiet Verzeichnisdienste.

Im Projekt DEEP (*Development of an European EduPerson)* geht es darum, den Bedarf an europaweitem Standardschema zur Abbildung von Personen und Organisationseinheiten im universitären Umfeld. In der ersten Phase wurde hierzu eine Bedarfsanalyse mittels eines Web basierten Fragebogens erstellt[12].

Im Projekt Schema Registry[13] wird das Problem angegangen, bereits definiertes standardisiertes und gut dokumentiertes LDAP-Schema besser zugänglich zu machen. Es geht also darum eine Datenbank aufzubauen, in der definierte Objektklassen, Attributtypen, Syntaxen, etc. über Metadaten suchbar zu machen. Ein solches Informationssystem zum Auffinden bzw. Registrieren von definiertem Schema wird im Rahmen dieses Projektes mit Verzeichnisdiensttechnologie implementiert.

### 3.1.2. Das Middlewareprojekt im Rahmen von Internet2

In den USA gibt es ein sehr großes Projekt zum Aufbau einer neuen Generation des Forschungsnetzes namens Internet2[14]. Im Rahmen dieses Projektes werden zusätzlich zu den reinen Netzwerkarbeiten auch neue Anwendungen definiert und implementiert. Ein wesentlicher Teil dieser Arbeiten werden im Rahmen des Teilprojektes MACEDir[15] entwickelt, in dem es um den Aufbau sogenannter Middleware geht, also einer

[9] Cgl. http://www.ietf.org/html.charters/ipsp-charter.html.
[10] Vgl. http://www.terena.nl.
[11] Vgl. http://www.terena.nl/task-forces/tf-lsd.
[12] Vgl. http://www.daasi.de/surveys/DEEP.
[13] Vgl. http:// www.daasi.de/projects/Schemaregistry.
[14] Vgl. http://www.internet2.org/.
[15] Vgl. http://middleware.internet2.edu/.

Softwareschicht, die zwischen dem Netzwerk und den eigentlichen Anwendungen steht und die letzteren grundsätzliche Dienste zur Verfügung stellen. Es geht im Wesentlichen um Informations- und Authentifizierungsdienste, die größtenteils auf Verzeichnisdiensttechnologie beruhen.

Im Rahmen dieser Aktivitäten wurden u.A. ein Personenschema für amerikanische Universitätsangehörige namens EduPerson entwickelt[16], sehr nützliche Dokumentation zu Implemeni´tierung von Verzeichnisdiensten erstellt [Gettes], sowie ein Domänen-übergreifendes Authentifizierungssystem namens Shiboleth[17] entwickelt. Weitere interessante Arbeiten dieser Gruppe wurden über Metadirectory [Bellina] und Abbildung von Gruppen [Barton] in Verzeichnisdiensten im Rahmen eines NFS-Förderprogramms gemacht.

## *3.2. LDAP im DFN Umfeld*

Seit 1994 gibt es DFN-Projekte an der Universität Tübingen zum Thema Verzeichnisdienste, die mit Mitteln des Bundesministerium für Bildung und Forschung gefördert wurden.

Die ersten beiden Projekte beschäftigten sich mit dem Aufbau und Betrieb eines Emailverzeichnis für die Forschung in Deutschland als zentraler Verzeichnisdienst für Organisationen, die nicht selbst Verzeichnisdienste betreiben. Dieser Dienst wird AMBIX genannt, Aufnahme von Mail-Benutzern in das X.500[18]. Er erlaubt Zugriff via Webfrontend auf gegenwärtig ca. 60.000 Datensätze. Die beteiligten Organisationen liefern ihre Benutzerdaten über einfache ASCII-basierte Datenlieferungsformate und wurden vom Projekt bei eventuellen Konvertierungsarbeiten unterstützt.

Von Vorneherein wurde hierbei der Datenschutz berücksichtigt. Ein von Rechtsexperten abgesegnetes Verfahren einer Wiederspruchslösung mit der Aufnahme nur eines Minimalsets von Datenfeldern und Datenexportbeschränkung an Länder mit unzureichender Datenschutzgesetzgebung implementiert. Um Datensammelprogrammen, sog. *Crawler* von potentiellen Spammern, also Organisationen, die ungebetene Werbe-E-Mails verschicken, abzuwehren wurden Algorithmen zum Erkennen von solchen Crawlern entwickelt. Zusätzlich wurden in die AMBIX-Datenbank spezielle Scheineinträge eingefügt deren Emailadresse sonst nirgends veröffentlicht sind, um den Erfolg solcher Abwehrmechanismen zu überprüfen. Bisher hat das Projekt wir kein Spam auf diesen Adressen erhalten.

2002 wurde ein letztes Neudesign der Software und Weboberfläche durchgeführt. Nun läuft der Dienst, der ursprünglich auf einer Implementierung des X.500(88)-Standards beruhte, unter LDAPv3 auf OpenLDAP-Basis. Ein neues Schema: DFNOrgPerson ermöglicht u.A. die Integration einer neuer Sichtbarkeitsoption, mit der der Betroiffene selbst entscheiden kann, wie weit sein Eintrag sichtbar ist, nur in eigener Domain, nur in Deutschland, nur in Datenschutztreibenden Ländern, oder aber weltweit.

Zusätzlich zu AMBIX wurde in den Projekten ein Kompetenzzentrum zur Beratung von Forschungsinstituten in Deutschland namens DFN Directory Services[19] aufgebaut, ein deutschlandweiter Index aller LDAP-Server im Forschungsbereich implementiert und integriert mit AMBIX betrieben, sowie den Betrieb des deutschen X.500-Countrylevel Servers im Rahmen des Europäischen Projekts NameFLOW[20] gewährleistet.

[16] Vgl. http://www.educause.edu/eduperson/.
[17] Vgl. http://middleware.internet2.edu/shibboleth/.
[18] Vgl. http://ambix2002.directory.dfn.de/.
[19] Vgl. http://www.directory.dfn.de/.
[20] Vgl. http://www.dante.net/nameflow/index.html.

In den letzten Projektphasen konzentrierte sich das Projektteam v.a. auf Konzeption und Implementierung von und Beratung zu Problemlösungen auf den Gebieten zentrales Authentifizierungssystem und Zertifikatsverzeichnis für PGP und X.509. Das Projekt läuft im Januar 2003 aus und der Weiterbetrieb all dieser Dienste muss nun von einer Nachfolgeorganisation übernommen werden. Hierzu wurde noch im Projektzeitraum eine kommerzielle GmbH, die DAASI International, gegründet, die jedoch nach Projektende keine weitere Förderung vom DFN erhalten wird. Es werden gegenwärtig Finanzierungsmodelle für den Weiterbetrieb diskutiert. Es wird sich nicht vermeiden lassen, die Dienste für beteiligte Organisationen kostenpflichtig zu machen.

Die DAASI International GmbH bietet Consulting, Design, Implementierung, Schulung, aber auch Serverhosting, und Datenmanagement an, auf den Gebieten Verzeichnisdiensttechnologien, PKI und Informationsmanagement (XML). Das Team wird sich wo möglich weiter an Forschung und Standardisierung beteiligen und ist offen für Kooperationen im Rahmen Forschungs- und Entwicklungsprojekten in Europa, im Bund und in den Ländern.

Um sich und ihren Kunden Produkt-Unabhängigkeit zu bewahren, konzentriert sie sich weiterhin auf Offene Standards und Open Source Software. Allerdings stellt sie auch gerne ihre Expertise in anderen Verzeichnisdiensttechnologien, wie X.500 Implementierungen, andere LDAP Implementierungen (Sun, Netscape, IBM), sowie Active Directory und Novell Directory Services zur Verfügung. Durch ihre Kontakte und Erfahrungen sind deutsche Forschungseinrichtungen weiterhin Hauptzielgruppe

Gegenwärtig arbeitet DAASI International an einem Projekt für die Universität Münster, in welchem auf Open Source Basis ein Veröffentlichungs-Verzeichnisdienst aufgebaut wird, der in folgenden Schritten zu einem Administrations- und Zertifikatsserver ausgebaut wird. Die Universität Münster ist hierbei offen für Abstimmungsprozesse mit anderen Hochschulen in NRW.

# 4. Referenzen


[Barton] Barton, Thomas: Practices in Directory Groups, NSF Middleware Initiative: Released for Public Review, January 2002, http://middleware.internet2.edu/dir/groups/rpr-nmi-edit-mace_dir-groups_best_practices-1.0.html.

[Bellina] Bellina, Brendan: Metadirectory Practices for Enterprise Directories in Higher Education, NSF Middleware Initiative, October 2002, http://middleware.internet2.edu/dir/metadirectories/internet2-mace-dir-metadirectories-practices-200210.htm.

[Chadwick] Chadwick, David: Understanding X.500 – The Directory, London ... 1994

[Dierks] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.txt.

[Gettes] Gettes, Michael: A Recipe for Configuring and Operating LDAP Directories, Version 2.1 (2002/10/10), May, 2000 (original version), http://www.georgetown.edu/giia/internet2/ldap-recipe/.

[Gietz] Gietz, Peter; Klasen, Norbert: An LDAPv3 Schema for X.509 Certificates, November 2002, (work in Progress), http://www.ietf.org/internet-drafts/draft-klasen-ldap-x509certificate-schema-01.txt.

[Good] Good, G.: The LDAP Data Interchange Format (LDIF) - Technical Specification, RFC 2849, June 2000, http://www.ietf.org/rfc/rfc2849.txt.

[Hodges] Hodges, J.; Morgan, R.: Lightweight Directory Access Protocol (v3): Technical Specification, RFC 3377, September 2002, http://www.ietf.org/rfc/rfc3377.txt.

[Howard] Howard, L.: An Approach for Using LDAP as a Network Information Service, RFC 2307, March 1998, http://www.ietf.org/rfc/rfc2307.txt.

[Klasen] Klasen, Norbert: Directory Services for Linux, in comparison with Novell NDS and Microsoft Active Directory. A thesis for submission to the Department of Computer Science in partial fulfillment of the requirements for the degree of "Diplom-Informatiker" at the Rheinisch-Westfälische Technische Hochschule, Aachen, August 2001, http://www.daasi.de/staff/norbert/thesis.pdf.

[Megginson] Megginson, R., et.al.: LDAP Client Update Protocol, June 2002, (Work in progress), http://www.ietf.org/internet-drafts/draft-ietf-ldup-lcup-03.txt.

[Myers] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, http://www.ietf.org/rfc/rfc2222.txt.

[Stokes] Stokes, Ellen, et.al.: Access Control Requirements for LDAP, RFC 2820, May 2000, http://www.ietf.org/rfc/rfc2820.txt.

[Vollrath] Vollrath, Christian: Entwicklung einer LDAP-basierten Verwaltung von IPsec-Policy-Informationen unter Verwendung des Common Information Model, Diplomarbeit zur Diplomprüfung im Fach Informatik dem Prüfungsausschuss für den Diplomstudiengang Informatik der Eberhard-Karls-Universität zu Tübingen vorgelegt, Tübingen, 30. April 2002, www.vollrath.org/studium/diplomarbeit/.

[X.500] ITU-T. Information technology – Open Systems Interconnection – The Directory, International Telecommunications Union, Geneva, 1993:
Overview of concepts, models and service. Recommendation X.500
Models. Recommendation X.501
Authentication framework. Recommendation X.509
Abstract service definition. Recommendation X.511,
Procedures for distributed operation. Recommendation X.518
Protocol specifications. Recommendation X.519
Selected attribute types. Recommendation X.520
Selected object classes. Recommendation X.521
Replication. Recommendation X.525
Use of systems management for administration of the Directory. Recommendation X.530

[Zeilenga] Zeilenga, Kurt: LDAP Transactions, November 2002, (work in Progress), http://www.ietf.org/internet-drafts/draft-zeilenga-ldap-txn-05.txt.