Verteilte Systeme

Verzeichnisdienste

Verzeichnisdienste

Problem:

Wie finde ich in einem großen verteilten System (z.B. dem Internet) eine(n) Dienst/Station/Person/Datum?

Idee:

Ein zentraler Dienst, den ich statisch adressieren kann, der mir diese Daten bietet.

Verzeichnisdienste

Es gibt verschiedene Dimensionen von Verzeichnisdiensten

- Lokal vs .Global
- Zentral vs. Verteilt
- Flach vs. Hirarchisch

Verteilte Systeme

Domain Name System (DNS)

Domain Name System (DNS)

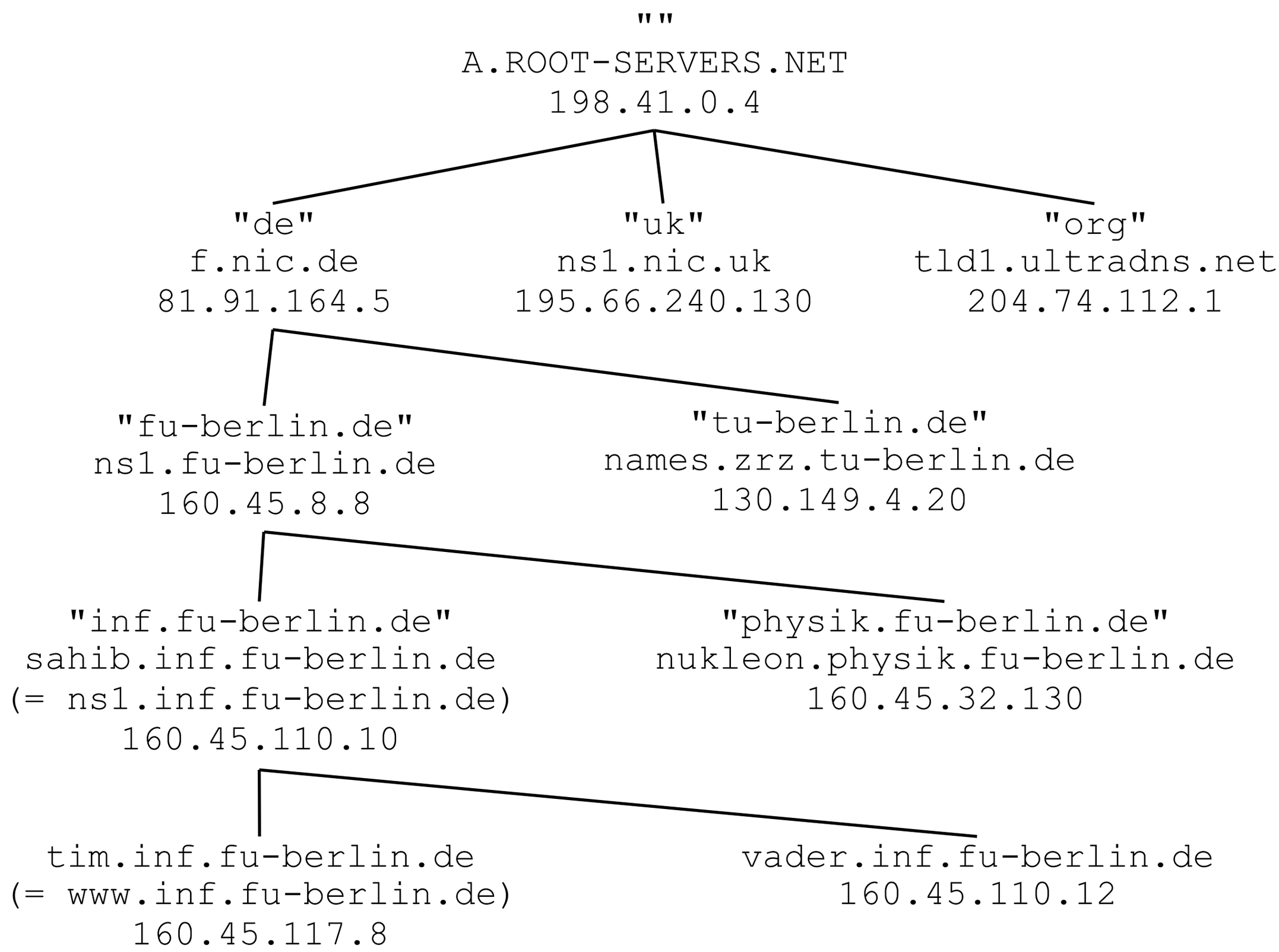
Auflösung von menschenlesbaren Rechner-Namen zu Adressen

```
www.inf.fu-berlin.de -> 160.45.117.200
```

Ursprünglich zentrale Registrierung per E-Mail beim
Stanford Research Institute (SRI) in Datei HOSTS.TXT

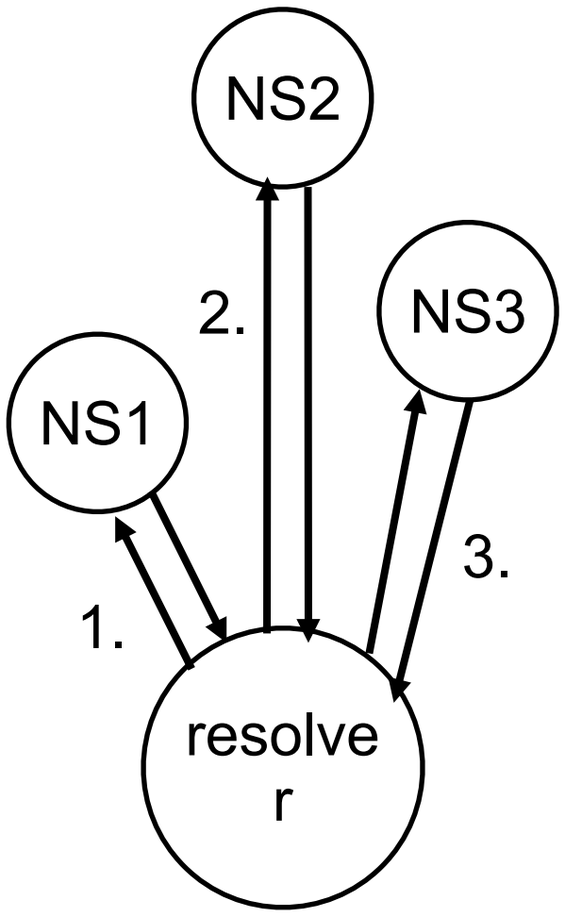
Beschaffung wöchentlich mittels FTP, Basis für /etc/hosts

Seit 1983 DNS (Mockapetris et al.)
Dezentrale Verwaltung durch Netz von domain name servers
für bestimmte Bereiche der Domänen-Hierarchie
mit Caching für Lastreduzierung
und Replikation für Ausfallsicherheit

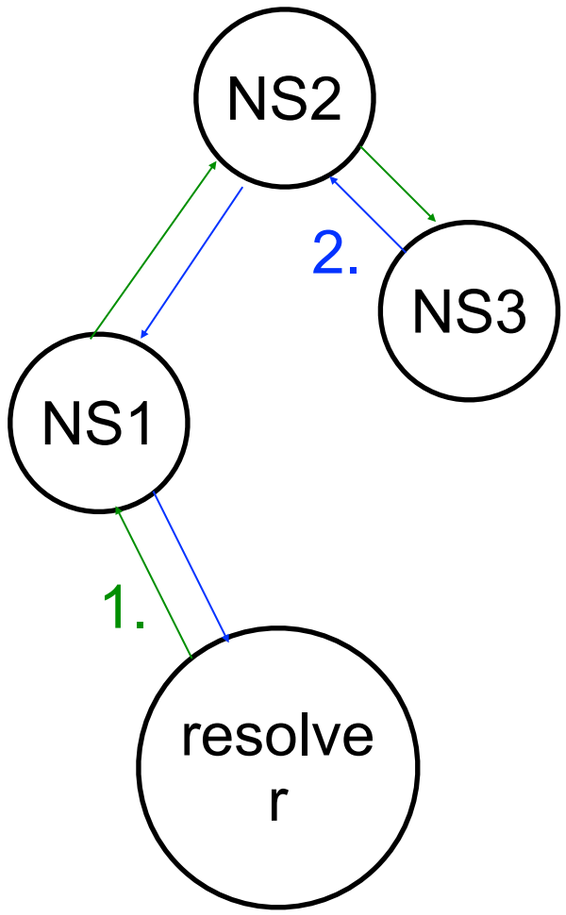


Normalerweise gibt es mehrere DNS-Server pro Domäne (evtl. rotierend)

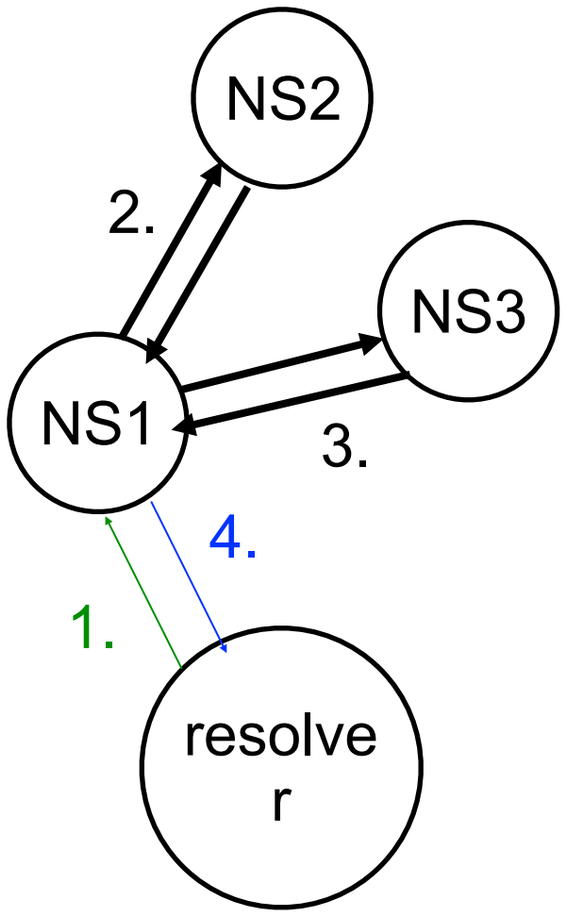
DNS-Abfragen auf mehrere Arten möglich:



iterativ



rekursiv



gemischt

Rekursive Weiterleitung durch DNS-Server ist optional.

Caching der Antworten reduziert Kommunikationsvolumen.

Programme wenden sich nicht direkt an DNS-Server,
sondern verwenden lokalen resolver (Bibliothek, Prozess)

Typische Konfiguration eines UNIX-resolvers:

Datei /etc/resolv.conf

```
search imp.fu-berlin.de mi.fu-berlin.de
nameserver 160.45.113.3          (cox.imp.fu-berlin.de)
nameserver 160.45.113.4          (quitte.imp.fu-berlin.de)
nameserver 160.45.41.8           (impdc1.imp.fu-berlin.de)
```

Einfache Rechner-Namen ohne Domänen-Teil werden der Reihe nach um die bei search angegebenen Domänen erweitert.

Sofern der erste bei nameserver angegebene DNS-Server nicht antwortet, werden die nachfolgenden der Reihe nach kontaktiert.

Die Datenbank eine DNS-Servers besteht aus mehreren Zonen-Dateien, normalerweise zwei pro Domäne:

- Abbildung von Namen auf Adressen
- Abbildung von Adressen auf Namen (optional)

Zonen-Dateien werden vom Master-Server auf Slave-Server repliziert.

Eine Zone enthält mehrere resource records (RR) der Form

```
<name> [<TTL>] [<class>] <type> <rdata>
```

`name`	ist ein Rechner-Name oder die spezielle kodierte Adresse davon
`TTL`	steuert die Verweildauer im Cache eines DNS-Servers/Resolvers
`class`	gibt die behandelte Protokollfamilie an (IN = Internet)
`type`	bestimmt den Typ des Eintrags und gültige Werte für `rdata`
`rdata`	beschreibt Eigenschaften von name, z.B. zugehörige Adresse

Pro Zone steuert ein **SOA** (start of authority) RR die Replikation auf Slave-Server:

```
inf.fu-berlin.de. IN SOA cox.mi.fu-berlin.de. \
        hostmaster.inf.fu-berlin.de. (
                917     ; serial number
                10801   ; refresh time, 3 hours 1 sec
                3600    ; retry time, 1 hour
                1209600 ; expiry time, 14 days
                86400   ; default TTL for RR caching, 1 day
        )
```

Replikat muss aktualisert werden, falls höhere `serial`-Nummer

Slave-Server kontaktiert den Master alle `refresh` Sekunden zur Prüfung

Wenn Master nicht erreichbar, neuer Versuch alle `retry` Sekunden

Falls nach `expiry` Sekunden keine Antwort, Zone ungültig

Ausserdem: Alle RR-Einträge der Zone dürfen maximal TTL Sekunden in DNS Caches verbleiben, sofern nicht individuell angegeben.

`A` RR bilden dann Namen auf IPv4-Adressen ab.

`AAAA` RR bilden dann Namen auf IPv6-Adressen ab.

Mit einem CNAME RR kann man alternative Namen (alias) vergeben.

```
piglet.imp.fu-berlin.de. IN  A      160.45.117.170
cst.mi.fu-berlin.de.     IN  CNAME  piglet.imp.fu-berlin.de.
```

Weitere Einträge geben gültige DNS-Server für die Zone an:

```
inf.fu-berlin.de.          IN  NS  ns1.fu-berlin.de.
ns1.fu-berlin.de.          IN  A   160.45.8.8
```

sowie Delegation an Server für untergeordnete Domänen:

```
spline.inf.fu-berlin.de.      IN NS    ns1.spline.inf.fu-berlin.de.
ns1.spline.inf.fu-berlin.de. IN A     130.133.110.70
ns1.spline.inf.fu-berlin.de. IN AAAA  2001:6f8:1c3c:babe::70:1
```

Dabei muss auch die Adresse des zuständigen Servers als glue record angegeben werden, um Anfragen korrekt weiterzuleiten.

Viele weitere RR-Typen:

`MX`	mail exchanger = zuständiger E-Mail-Server
`SRV`	Verweise auf Dienste
`TXT`	Freitext!
...	

Zu jeder Zone sollte es auch eine reverse-Zone gegeben für korrespondierende Rückwärtsauflösung von Adresse zu Name.

Realisiert mit spezieller Pseudo-Domäne `in-addr.arpa` :

```
117.45.160.in-addr.arpa   IN NS  sahib.inf.fu-berlin.de.
8.117.45.160.in-addr.arpa IN PTR tim.inf.fu-berlin.de.
```

umgekehrte IP-Adresse als "Domäne"

IDNA - International Domain Names in Applications

Problem: Für Domänen-Namen ist nur eine sehr kleine Untermenge von ASCII-Zeichen erlaubt: A-Z, 0-9 und - Darstellung von anderen Zeichen (Umlaute, Akzente, ...) oder anderen Schrift-Systemen (Griechisch, Japanisch, ...) nicht möglich.

Abhilfe: Anwendungen kodieren Namen zunächst in kompatible Form "Punycode" (analog zu UNICODE als UTF-8-Kodierung), z.B.

bücher.de -> xn--bcher-kva.de

xn--	reservierter IDNA-Prefix als Markierung
bcher	Zeichen des Original-Namens ohne Sonderzeichen
-	Trennzeichen
kva	Kombination aus Sonderzeichen (UNICODE) und Position

DNS – Sicherheit

Problem: Viele Dienste stützen ihr Sicherheitsmodell auf die Korrektheit von Domänen-Namen, aber DNS sieht keine Sicherheitsmechanismen vor (Cache-Poisoning, rekursive DNS-Server die Ergebnisse verfälschen)

Abhilfe: Signieren der RRs durch neue Record-Typen:

RRSIG <Typ> <Alg> … - Signatur eines eines RR-Sets

DNSKEY - Public-Key der Zone

Zusätzlich zur Delegation durch NS-Records enthalten die übergeordneten Zonen auch RRSIG-Records mit den Hashes der Public-Keys der untergeordneten Zonen -> Chain of Trust.

DNS – Sicherheit

Problem: Wie signiere ich das nicht-existieren von Einträgen?

Abhilfe: Zu jedem RRSIG-Eintrag existiert ein zusätzlicher Eintrag der Form

- RRSIG NSEC <Alg> … - Signatur eines eines RR-Sets

der auf den nächsten existierenden Eintrag verweist, bei Anfrage eines nicht existenten wird der vorhergehende NSEC-Record geliefert.

Neue Probleme:

- Zone walking
  - -> Hashes statt echter Einträge
- Dynamische Zonen (z.B. für IPv6 Rückwärtsauflösung)
  - -> Online-Signatur, dass der eine Eintrag nicht exisitiert

Bonjour / Rendezvous – DNS as discovery service

Idee: DNS-System zum Auffinden von **Geräten** und **Diensten** verwenden

Dazu Verwendung von **PTR, SRV** (service) und **TXT** RRs:

```
_printer._tcp.local.    IN  PTR  LaserWriter 8500._printer._tcp.local.
LaserWriter 8500._printer._tcp.local.  IN SRV  0 0 515 myhost.local.
                        IN  TXT  #pdl=application/postscript#color=T
```

`LaserWriter 8500`	Name des antwortenden Dienstes
`_printer`	Protokoll für Dienst (hier: LPR)
`_tcp`	zugrundeliegendes Transport-Protokoll
`.local.`	spezielle mDNS-Domäne für lokales Netz (opt)
`0 0`	Priorität/Gewicht des Dienstes, steuert Auswahl
`515 myhost.local.`	Port und Name des anbietenden Rechners

Auflisten durch PTR-Anfrage an DNS-Server, oder Multicast für `.local`

SRV-Anfrage liefert Host/Port, TXT-Anfrage liefert Dienst-Details

Verteilte Systeme

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP)

Zugriff auf hierarchische Datenbank innerhalb von Organisationen zum Auffinden von Ressourcen wie z.B.

- Benutzern
- Gruppen
- Rechnern
- Adressbüchern
- u.v.m. …

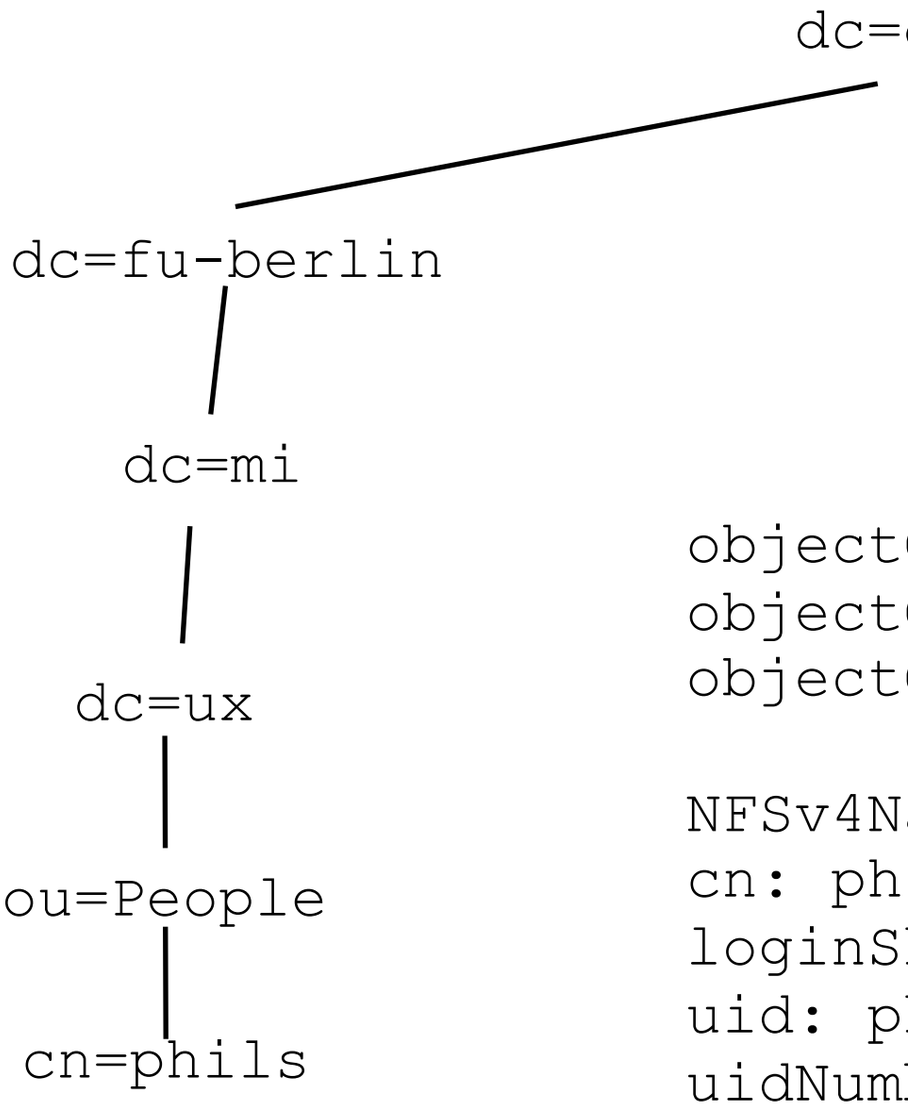
An der Universität von Michigan entwickelte Vereinfachung des X.500 DAP Protokolls, dass nie vollständig Implementiert wurde (ISO Netzwerkstack, extrem komplex)

Einträge im Verzeichnen haben einen hierarchischen Namen, den **Distinguished Name** (DN) und eine Reihe von Klassen (Object Classes), für die es eine Reihe von verpflichtenden und optionalen Feldern gibt (festgelegt im Schema des Verzeichnises).

Abfragen werden in zwei Phasen gemach

- **Binden** an den LDAP-Server (incl. autentifizierung)
- Durchsuchen des Baumes abwärts vom **Base-DN**

ggf. sind Verweise auf andere LDAP-Server für Unterbäume möglich




```
objectClass: top
objectClass: posixAccount
objectClass: NFSv4RemotePerson

NFSv4Name: phils@mi.fu-berlin.de
cn: phils
loginShell: /usr/bin/zsh
uid: phils
uidNumber: 13446
homeDirectory: /home/fenn/phils
gidNumber: 11163
GSSAuthName: phils@FU-BERLIN.DE
gecos: Philipp Schmidt
```

Vorteile:

- Schneller Zugriff
- Hierarchische Struktur inkl. Delegation
- Flexible Struktur, schnell individuell zu erweitern

Nachteile:

- Keine Normalformen, Multi-Value Felder
- Sehr individuelle Namensräume (Grund zur Verwirrung)
- Komplexe Berechtigungsstruktur

Implementierungen (Auswahl)

- OpenLDAP (Server)
- Microsoft Active Directory (Integration in Server, Exchangen, …)
- IBM (Integration in Lotus Notes, Server, …)
- u.v.m. …

Übung 6 zum 28. Juni 2011

Wir werden über die nächsten Übungszettel eine kleine verteilte Peer-to-Peer Wirtschaftssimulation schreiben.

Jeder Knoten soll nun entweder einen Planeten, auf dem verschiedene Handelswaren angeboten werden,

oder ein Handelsraumschiff darstellen.

Implementieren Sie zwei Kommandos

goods
: listet die im Universum gehandelten Waren auf

course <good>
: Listet an und Verkaufskurse der Handelswaren im Universum auf.