Internetanwendungstechnik

Dienste und Anwendungen

Gero Mühl

Technische Universität Berlin
Fakultät IV – Elektrotechnik und Informatik
Kommunikations- und Betriebssysteme (KBS)
Einsteinufer 17, Sekr. EN6, 10587 Berlin

Dienste und Anwendungen

Domain Name Service (DNS)

Anforderungen

- Sprechende, benutzerfreundliche Namen statt IP-Adressen
- Benennung beliebiger Objekttypen
- Dezentrale Namensauflösung mit sehr guter Skalierbarkeit
- Hohe Performance und Verfügbarkeit
- Dezentrale Instanzen zur Namensvergabe
  → hierarchischer Namensraum

→ Domain Name System (DNS)
  - definiert in RFCs 1034 und 1035

Begriffe

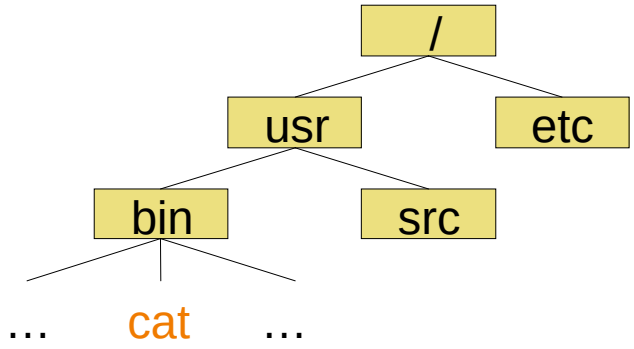
- Namen
  - Name *„Was wird gesucht?“*
  - Adresse *„Wo ist es?“*
  - Weg *„Wie gelangt man dahin?“*
  - Attribute *„Welche Eigenschaften sind vorhanden?“*
- Namen identifizieren Objekte im verteilten System (Beispiele)
  - Ressourcen wie Rechner, Drucker und Dateien
  - Benutzer und Benutzergruppen
  - Dienste und Prozesse, etc.
- Entwurfsfragen
  - Struktur der Namen und Namensraum
  - Generierung und Eindeutigkeit
  - Auflösung (= Abbildung auf andere Namen / Adressen)
  - Transparenz

Namen und Namensraum

- Syntax
  - Unstrukturiert (*z.B. DHDIBM1)*
  - Strukturiert (*z.B. kbs.cs.tu-berlin.de)*
  - Attributiert (*z.B. [country=de, org=tub, department=cs, group=kbs])*
    - Reihe von [Attributname, Attributwert]-Paaren
- Namensraum
  - Menge aller möglichen Namen bei gegebener Namenssyntax
  - Struktur und Aufbau
    - Flach → unstrukturiert
    - Hierarchisch → hierarchische Anordnung von Kontexten, relative Namen
    - Wegorientiert → Namenskomponenten bestimmen Weg zum Objekt

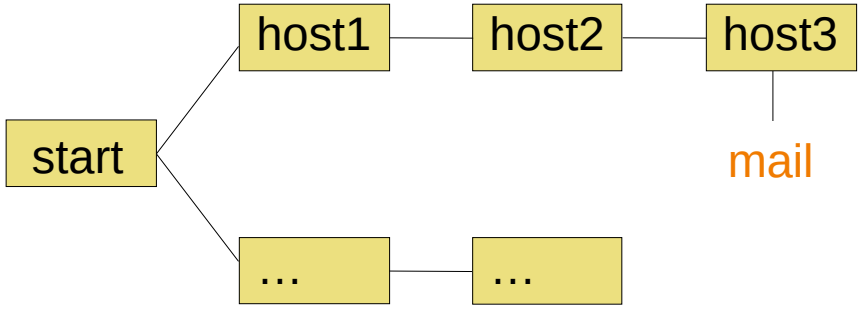
Beispiele

- Hierarchisch



/usr/bin/cat

- Wegorientiert



host1!host2!host3!mail

Verzeichnis (Namensdienst)

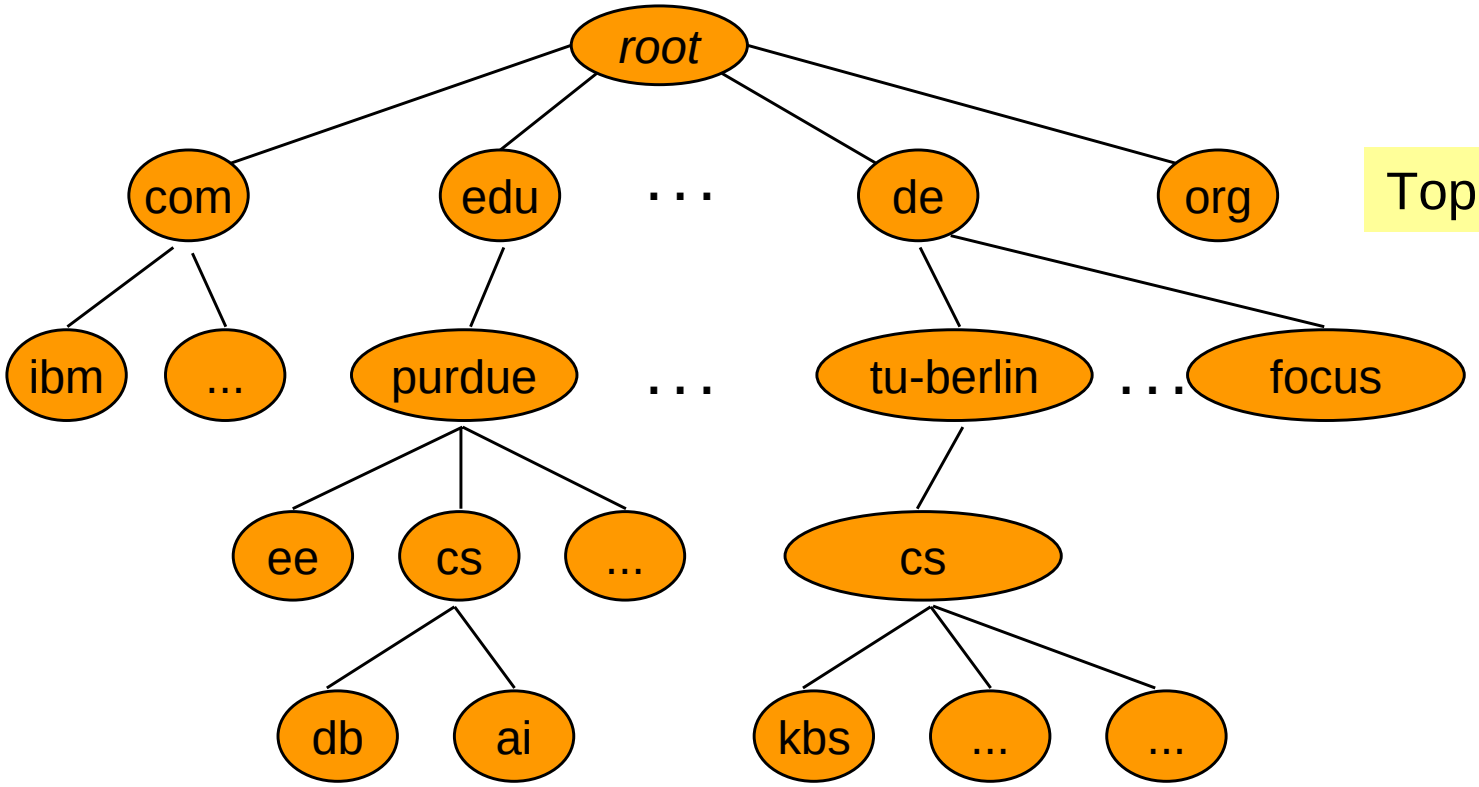
- Telefonbuch (weiße Seiten)
  - Name → Telefonnummer
- Branchenbuch (gelbe Seiten)
  - Attribute (und evtl. Namen) → Telefonnummer
- Namensdienst (Name Service, Directory Service)
  - Bildet Namen auf Adressen ab
  - Finden einer Adresse bei vorgegebenem Namen → weiße Seiten
  - Finden einer Adresse auf Basis bestimmter Attributwerte → gelbe Seiten
  - Verwaltung von Querverweisen → Alias
  - Verwaltung von Gruppen von Namen → Verteilerliste

Domain Name Service (DNS)

- Verteiltes System zur Abbildung der Namen auf IP-Adressen
- Hierarchisches Domänen-Konzept
  - Beispiel: `cs.tu-berlin.de`
  - `tu-berlin` ist Subdomain von `de`
- Syntax der Namen
  - <256 Zeichen, Teile <64 Zeichen
  - Groß-/Kleinschreibung unerheblich
- Hierarchische Autorität zur Namensvergabe
- Benennung beliebiger Objekttypen
- Domänen unabhängig von physischen Netzen

DNS Namensraum – Top Level Domains

- Organisatorisch COM, EDU, GOV, MIL, ORG, NET, INT, ...
- Geographisch DE, UK, US, AU, TO, ...



Top Level Domains

Name Server

- Domänen nutzen miteinander kooperierende Name Server
  - Bilden ein Verteiltes System
  - Replikation der Name Server steigert die Zuverlässigkeit
  - Jeder Name Server kennt mindestens einen übergeordneten Name Server
  - Primary Name Server kennen die Root Name Server
- Anfragen der Clients an den Name Service
  - Per UDP an Port 53 des lokalen Name Servers
  - Anfrage nach Name, Objekttyp, ..., iterativ / rekursiv
- Caching der Namensinformation im Name Server
  - Einträge im Cache: Name / IP-Adresse / Name Server
  - Möglichkeit veralteter Information wird mitgeteilt
  - Bereinigung des Caches mit TTL (time-to-live) für Einträge

Datenbank im Name Server

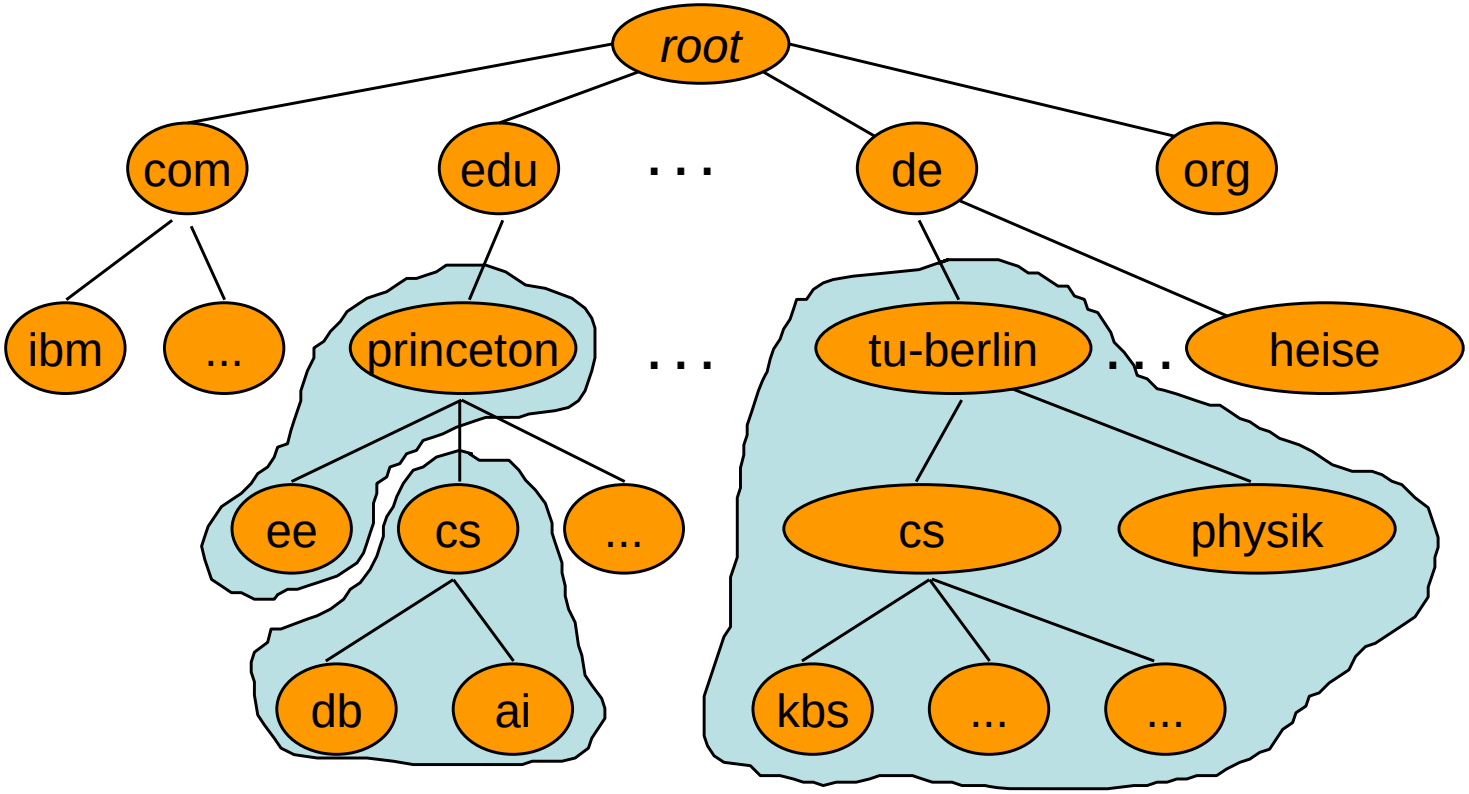
- Name Server speichert seine Information in einer Datenbank
- Einträge heißen Resource Records (RR)
- Neue Resource Records sind in Antworten auf *Resolve()*-Anfragen an Name Server enthalten
- Jeder Resource Record ist ein 5-Tupel
  - Domain Name — Domain-Name, für den dieser RR gilt
  - Class — Klassifizierung der Information (meist „IN“ = Internet)
  - Type — Typ des Eintrags
  - Value — Datenwert (abhängig vom Typ)
  - Time-to-Live — Lebensdauer des Eintrags (in Sekunden)

Resource Record Typen (Beispiele)

Typ	Bedeutung	Wert
SOA	NS zuständig für	mehrere Parameter
A	IP-Adresse f. Namen	32-Bit-Integer
MX	E-Mail Knoten	Priorität, Empfänger für ...
NS	Name Server	Name eines NS
CNAME	Alias	Name einer Domäne
PTR	inverse Abbildung	Name
HINFO	Host-Beschreibung	ASCII-Text
TXT	Beliebiger Text	ASCII-Text

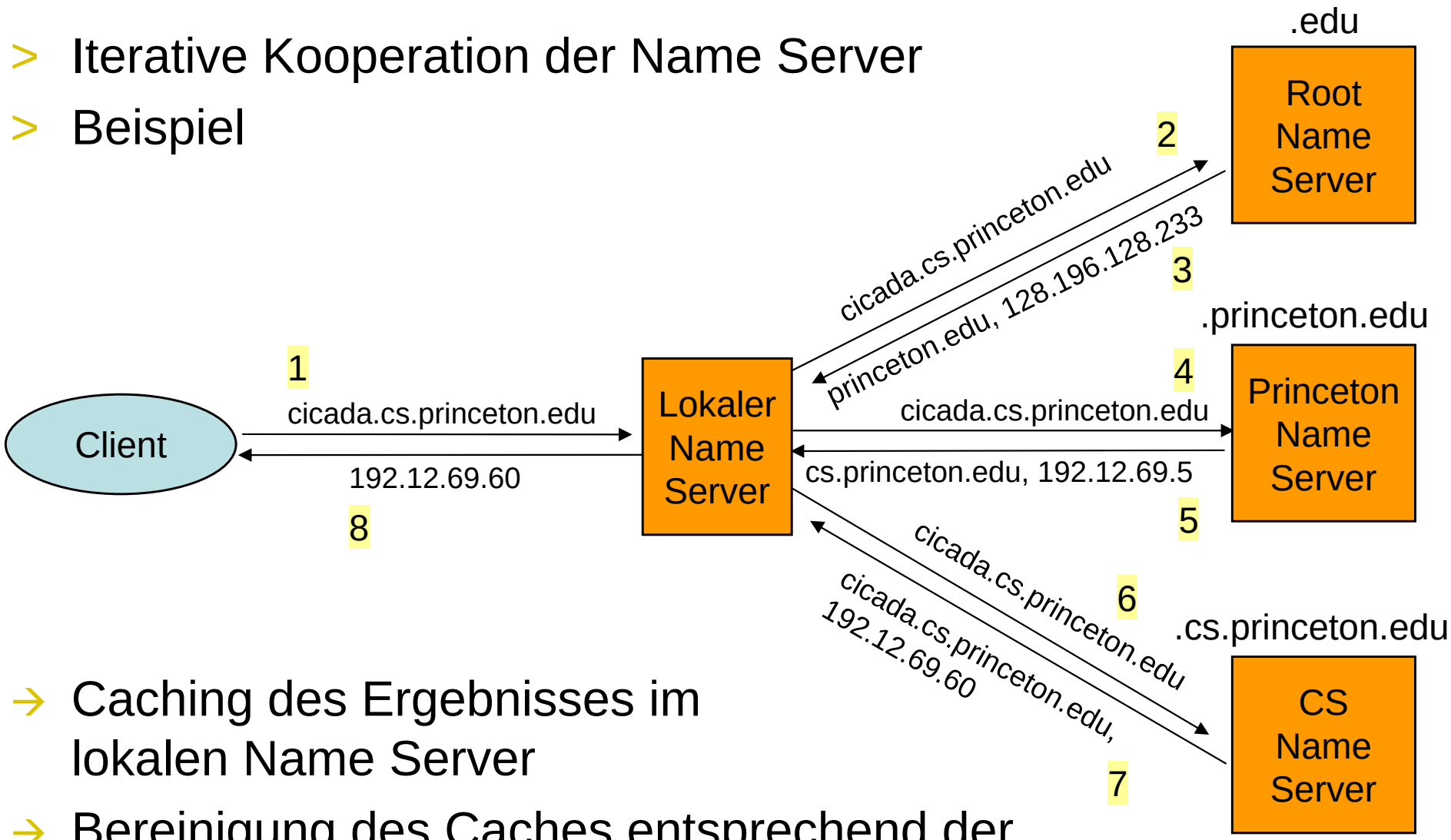
Platzierung von Name Servern

- DNS-Namensraum wird in nicht überlappende Zonen eingeteilt
- Jede Zone enthält einen Primary Name Server und evtl. weitere Secondary Name Server



Namensdienst

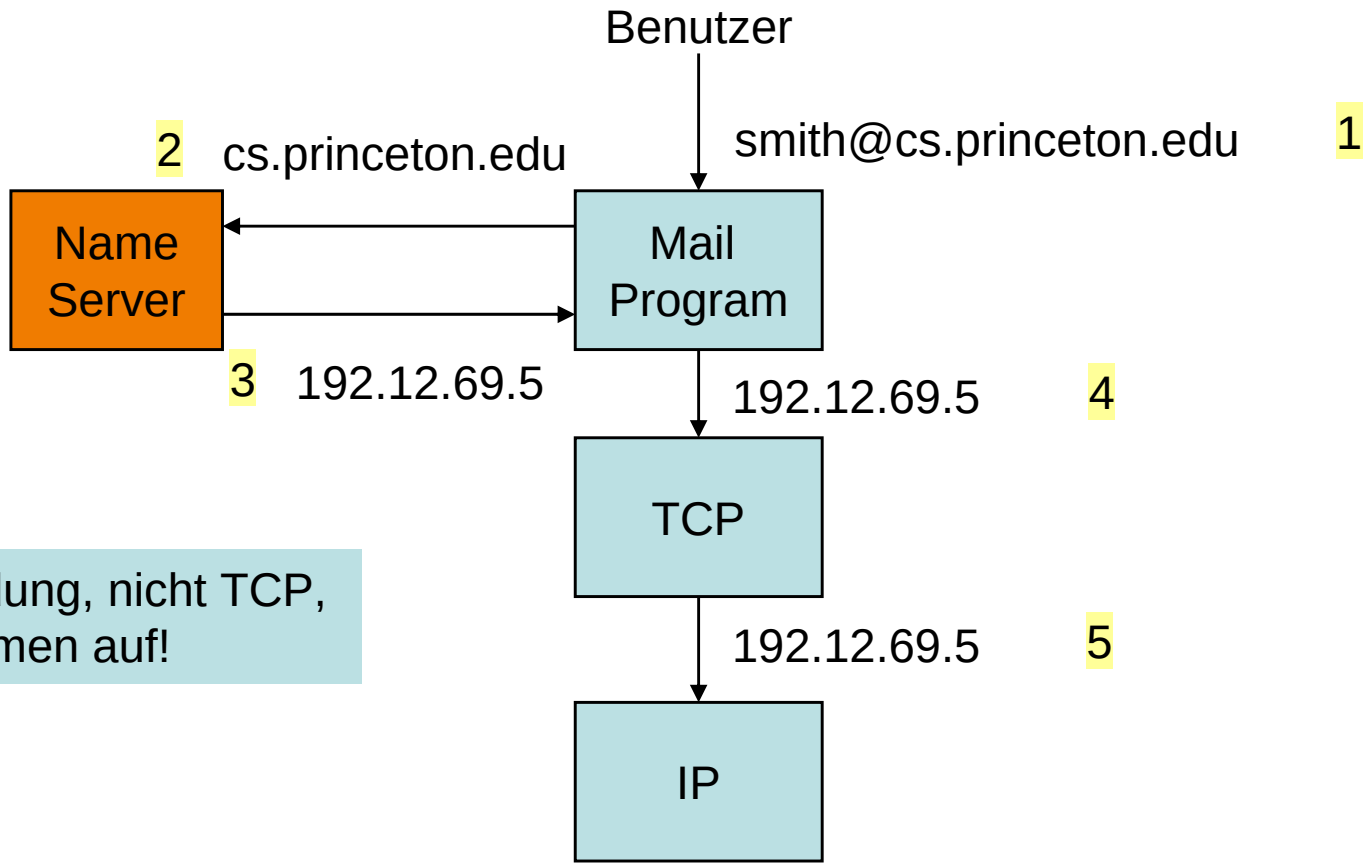
- Iterative Kooperation der Name Server
- Beispiel



- Caching des Ergebnisses im lokalen Name Server
- Bereinigung des Caches entsprechend der Time-to-Live des Eintrags

Namensdienst

- Beispiel: E-Mail an „**`smith@cs.princeton.edu`**“



Die Anwendung, nicht TCP, löst den Namen auf!

Zusammenfassung

- DNS ist eine verteilte Datenbank
  - Name Service bildet Namen auf Adressen ab
  - Daten werden lokal gepflegt und sind global zugreifbar

- DNS Skalierbarkeit beruht auf
  - Replikation → Robustheit
  - Caching → Performance
  - Hierarchische Organisation → dezentrale Namensvergabe

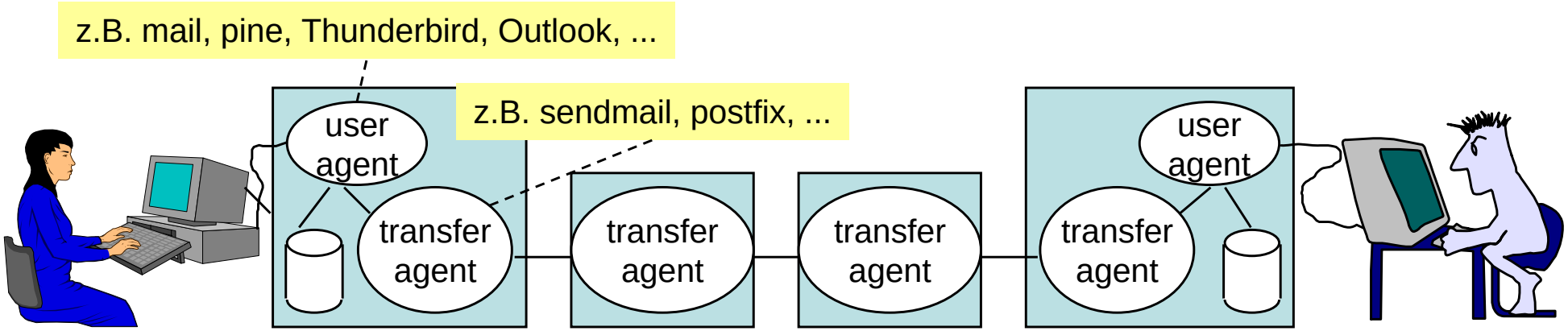
Weitere Aspekte

- LDAP (= Lightweight Directory Access Protocol)
  - Modifizierter OSI X.500 Namensdienst über TCP/IP
  - Bietet sehr viel mehr Funktionalität als DNS (weiße Seiten, gelbe Seiten, …)
- DNS Datenbank wurde ursprünglich statisch vom Administrator angelegt
  - Neue UPDATE-Option im DNS-Protokoll [RFC 2136]
  - Dynamische Erweiterung und Modifikation der Datenbankeinträge
- IPv6 verlangt auch eine neue Version von DNS

Electronic Mail (E-Mail)

Anforderungen

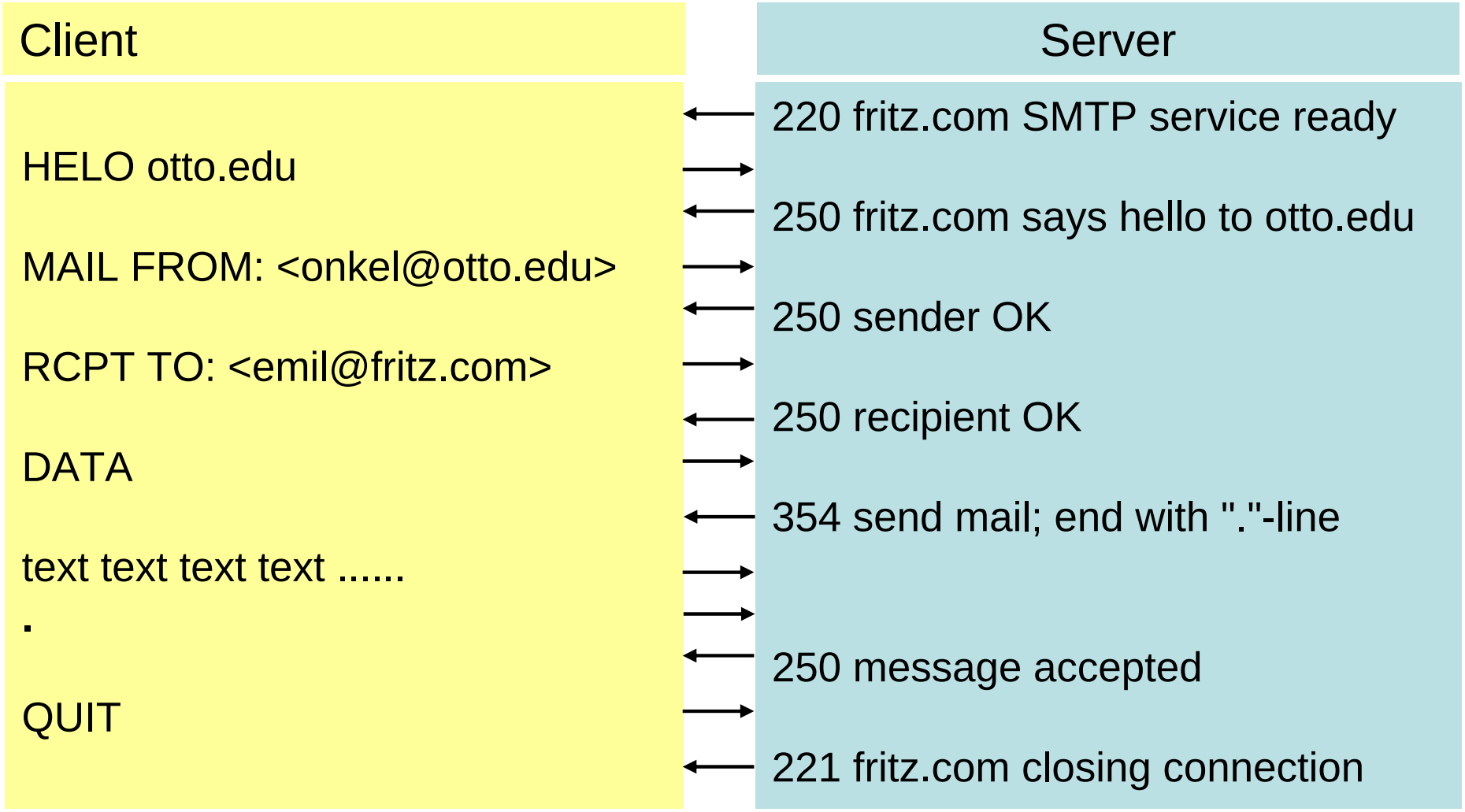
- Zuverlässiges Verschicken von Text und Daten
- Entkoppelte, Store-and-Forward-Übertragung, ggf. mit Auslieferungsbestätigung
- Adressierung von Einzelpersonen und Gruppen
- Strukturierte Nachrichten mit klarer Trennung der Einheiten
- Weiterleiten und Kopieren von Nachrichten



Simple Mail Transfer Protocol (SMTP)

- Spezifiziert in RFC 821 bzw. 2821
- Auslieferung von Mail über eine TCP-Verbindung (Port 25)
- Der Absender (= SMTP Client) kommuniziert mit dem SMTP Daemon des Empfängers (= Server)
- SMTP ist ein einfaches ASCII-Protokoll
  - Client-Kommandos als Zeichen
  - Replies als Ziffern und Text
- Client-Kommandos
  - HELO, MAIL FROM, DATA, QUIT, ...
- Server-Antworten
  - 220 (= service ready)
  - 250 (= other party OK)
  - 354 (= send mail)
  - ....

SMTP – Beispieldialog



Nachrichtenformate [RFC 822, 2822]

- Aufbau einer Nachricht
  - Umschlag (Envelope) + Briefkopf (Header) + Rumpf (Body)

- Umschlag wird von den Transfer Agents benutzt und gebildet
- Header-Felder werden für den Transport benötigt
  - To: (primäre) Empfänger
  - Cc: (sekundäre) Empfänger
  - Bcc: (unsichtbare) Empfänger
  - From: Briefschreiber
  - Sender: Absender
  - Received: pro Zwischen-Transfer-Agent eine Zeile
  - Return-Path: Rückweg für Antworten

Nachrichtenformate [RFC 822, 2822]

- Zusätzliche Felder im Header (optional)
  - Date: Datum und Zeit des Abschickens
  - Reply-To: Adresse für Antworten
  - Message-Id: eindeutige Nummer
  - In-Reply-To: Bezug ("Message-Id")
  - References: andere relevante Message-Ids
  - Keywords: vom Benutzer angegebene Keywords
  - Subject: Betreff
  - X-?????: benutzerdefinierte Felder

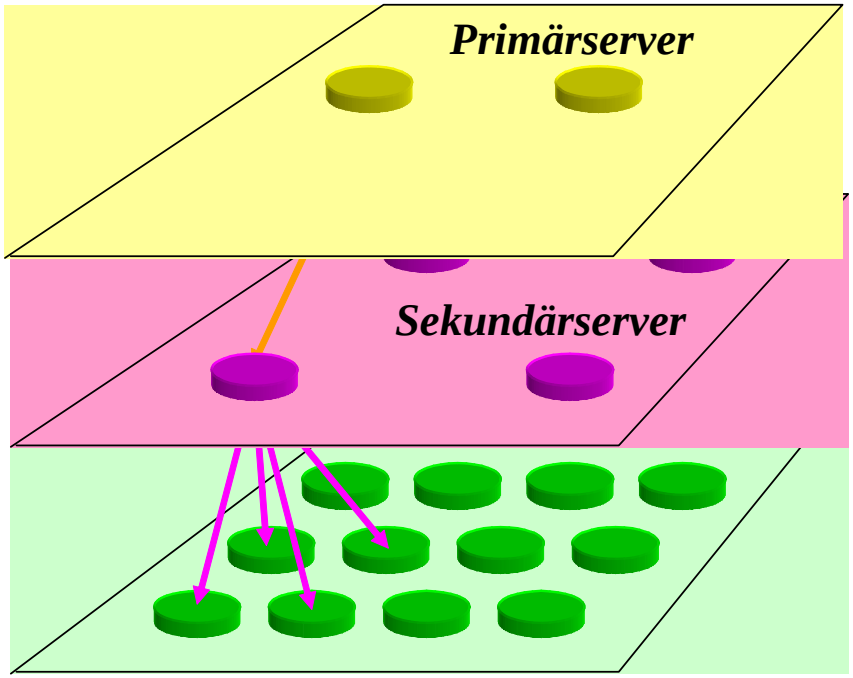
Network Time Protocol (NTP)

Network Time Protocol (NTP)

- Zeitsynchronisierung wichtig in Netzen
  - Datensynchronisierung nach Zeit
  - Zeitstempel in Sicherheitsinfrastrukturen (z.B. Kerberos)
  - …
- Entwurfsziel
  - Weltweite Synchronisierung mit Atomuhren
  - Auch über längere Zeiten ohne Verbindung
  - Kompensierung der für Uhren typische Drift (clock drift)
  - Schutz vor absichtlicher Verfälschung

Network Time Protocol (NTP)

- Hierarchische Architektur
  - Verteilung auf verschiedene Strata
  - Dynamische Rekonfigurierung



Network Time Protocol (NTP)

- NTP Synchronisierungsmodi
  - Über Multicast im LAN
  - Symmetrischer Modus für Austausch von Zeiten in derselben Ebene
- Genauigkeit im LAN im Millisekundenbereich.
- Liste von Zeitservern: http://www.ntp.org.
- Konfiguration unter Linux in `/etc/ntp.conf`

```
server hora.cs.tu-berlin.de
server ntps1-0.uni-erlangen.de
driftfile /etc/ntp.drift # vom Dämon verwaltet
```

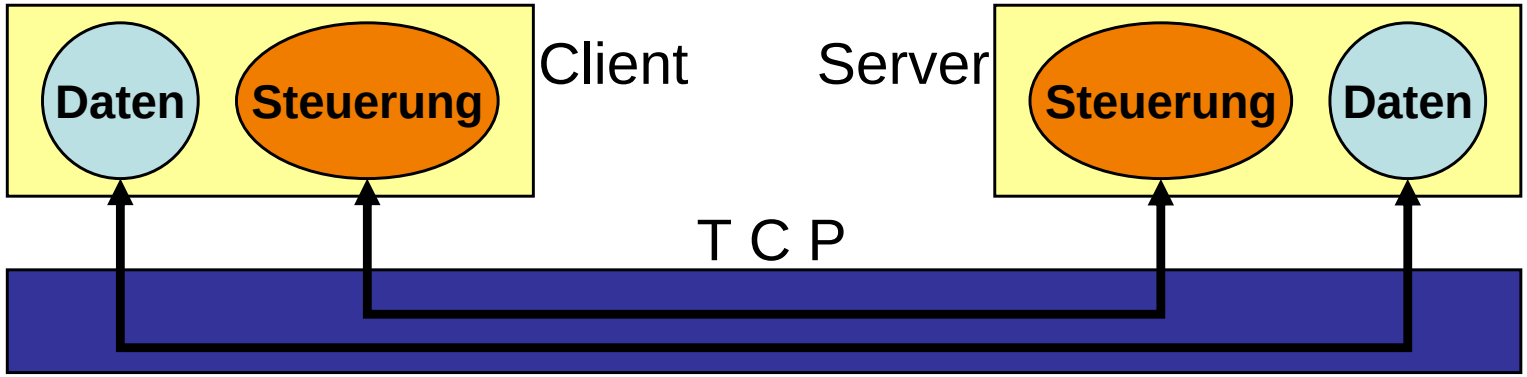
File Transfer Protocol (FTP)

File Transfer Protocol (FTP) [RFC 959]

- Ermöglicht den Transfer ganzer Dateien zwischen Systemen
  - Einfacher als die Einbindung von entfernten Dateien in das lokale Dateisystem (z.B. mittels NFS)
  - Bietet einfache Möglichkeit auf gemeinsame Daten zuzugreifen
  - Wird meistens für nur lesbare Dateien benutzt
  - Falls die Datei verändert wird, muss sie explizit per erneutem Transfer zurückgeschrieben werden
- Zugang über Benutzerkonto (Name + Passwort)
  - Benutzerkonto, Passwort und Daten werden im Klartext übertragen
  - Bietet interaktiven Zugang als auch Programmierschnittstelle
  - Verschiedene Datenformate (ASCII und binär)

FTP-Verbindungen

- Arbeitet mit TCP
  - 2 Verbindungen pro Übertragung
  - Kontrollport = 21
  - Datenport = 20



- Zusätzlich existiert ein *Trivial File Transfer Protocol (TFTP)*
  - Aufbauend auf UDP, kleiner Umfang, einfache Implementierung

FTP-Sitzung

```
$ ftp diamant.vsb.informatik.uni-frankfurt.de
Connected to diamant.vsb.informatik.uni-frankfurt.de
220 diamant.vsb.informatik.uni-... FTP server ready.
Name (diamant:otto): anonymous
331 Guest login ok, send ident as password.
Password: onkel@otto.edu
230 Guest login ok, access restrictions apply.
ftp> cd pub
200 PORT command ok.
ftp> get some.file.data my.local.data
150 Opening data connection for /bin/ls (141.2.2.1, 2363)
(290455 bytes).
226 Transfer complete.
300000 bytes received in 99 seconds (30 Kbytes/s)
ftp> close
221 Goodbye.
ftp> quit
$
```

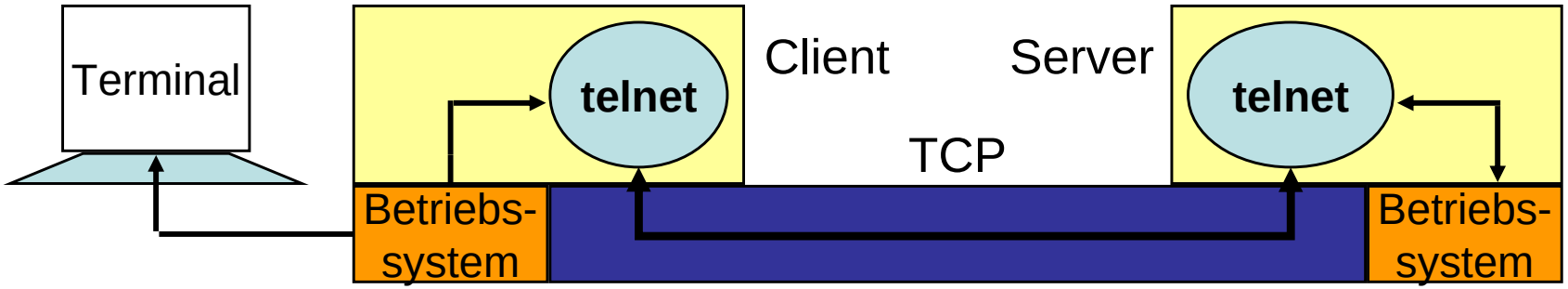
Telnet

Telnet [RFC 854]

- Eine der ersten und erfolgreichsten Internetanwendungen
- Telnet liefert interaktiven Terminalzugang über das Netz
  - Definiert ein Network Virtual Terminal (NVT) → eine Standardschnittstelle für den Zugang zu entfernten Rechnersystemen
  - Erlaubt das Aushandeln von Optionen zwischen Client und Server, z.B. Transferformate
  - Bietet sowohl eine interaktive Bedienung als auch eine Programmierschnittstelle
- Muss sorgfältig in das Betriebssystem integriert werden
  - Was geschieht bei **`Ctrl-c`**?
  - Was geschieht bei **`CR/LF`**?

Eigenschaften

- Client verbindet sich mittels TCP zu Port 23 des Servers
- Client registriert Tastaturanschläge, schickt sie zum Server, empfängt die Ergebnisse und leitet sie zum Bildschirm
- Server leitet alle Daten an das lokale System weiter und gibt Ergebnisse zurück
- Unsicher, da Daten, Benutzerkennungen und Passwörter im Klartext übertragen werden
- Secure Shell (SSH) ermöglicht die sichere Übertragung (Port 22)



Network Virtual Terminal (NVT)

- NVT ist das Transferformat für Daten und Befehle
- Befehle
  - Die üblichen ASCII-Steuerbefehle
    - CR Wagenrücklauf
    - LF nächste Zeile
    - HT Tabulator
    - BS Backspace
    - ...
  - Plus spezifische Befehle
    - IP Interrupt Process
    - AO Abort Output
    - SYNCH Rücksetzen und auf Befehle warten
    - ...

Network Virtual Terminal (NVT)

- Daten
  - Übertragung als 7-Bit-ASCII Zeichen
  - Auch optional (nach Aushandlung) 8-Bit-ASCII oder binär
- Optionen
  - Die Aushandlung von Optionen kann von Client und Server initiiert werden
  - NVT legt einen Mindestumfang unterstützter Optionen fest
  - "neue" Terminals mit mehr Optionen können so mit "alten" Systemen interagieren

Dienste und Anwendungen

USENET NEWS

USENET NEWS

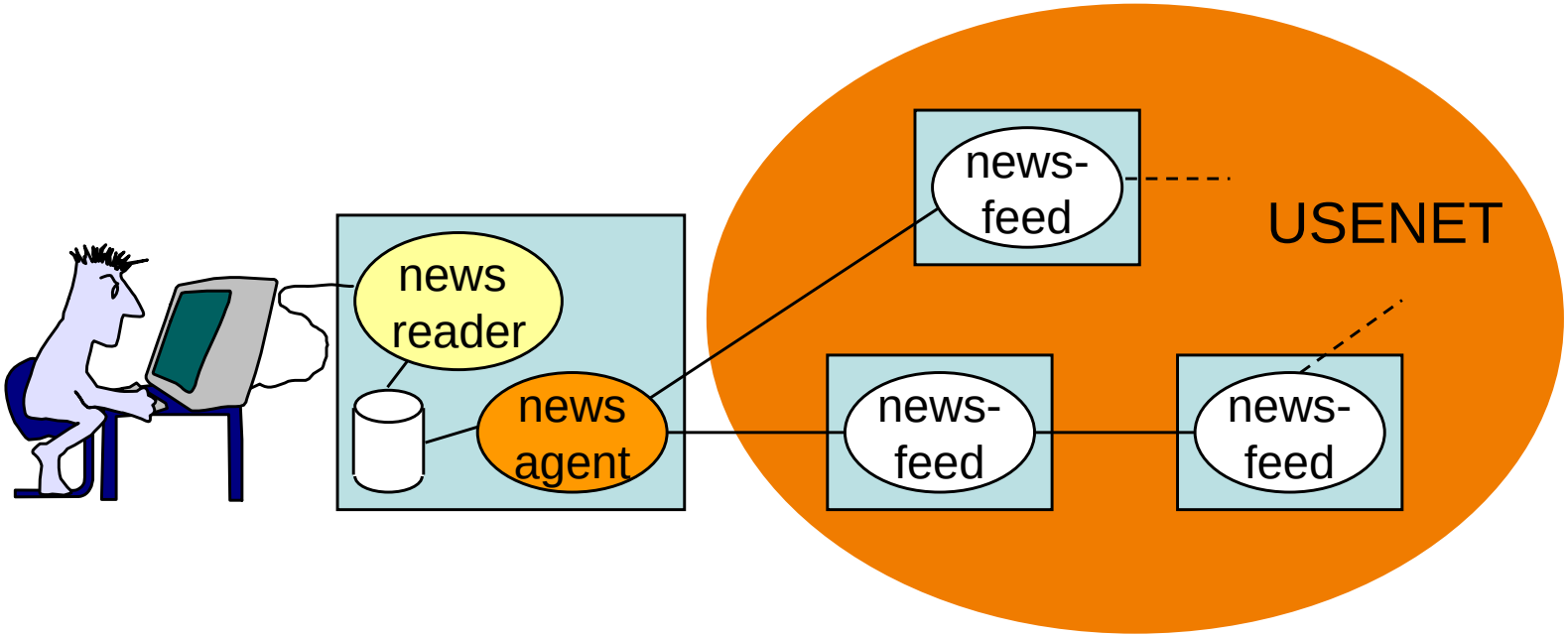
- Weltweite Diskussionsforen zu beliebigen Themen
- Die Nachrichten werden mittels NNTP (Network News Transfer Protocol) übertragen [RFC 977]
- NNTP ähnelt stark dem E-Mail-Transferprotokoll SMTP
- Nachrichten werden hierarchisch angeordneten News Groups zugeordnet

News Groups

- Oberste Hierarchieebene der News Groups, u.a.
  - comp — Computer
  - sci — Natur- und Ingenieurwissenschaften
  - humanities — Literatur und humanistische Themen
  - news — Themen, die das USENET betreffen
  - rec — Freizeit und Erholung
  - misc — "Vielfältigste Themen"
  - soc — Soziale Themen
  - talk — Geschwätz
  - alt — "life, universe, and all the rest"
- News Groups werden von den Teilnehmern abonniert
- Abonnenten sehen alle Nachrichten der Gruppe und können eigene Nachrichten anhängen

Implementierung

- Anschluss an USENET geschieht über sog. *Newsfeed(s)*
- Je nach Konfiguration schicken Newsfeeds neue News (*push*) oder der Abnehmer fragt nach neuen News (*pull*)



Implementierung

- Format der Artikel
  - Gleiches Format wie RFC 822-Mails (Kompatibilität!)
  - Mit einigen Erweiterungen
- Protokoll
  - Newsfeed erreichbar mit TCP unter Well-Known-Port 119
  - NNTP steuert Dialog zwischen News-Anbieter und Benutzer-Agent
  - News-Agent und News-Anbieter tauschen Information aus
    - Gibt es neue Artikel ?
    - Gibt es neue News Groups?
  - News Groups können selektiv angefragt werden
  - News-Agent gibt lokal aufgegebene Nachrichten an seinen News-Anbieter weiter

Implementierung

- NNTP Protokolldaten und Befehle in ASCII
- NNTP Kommandos
  - LIST — Liste Deiner Newsgroups und Artikel
  - NEWSGROUPS datum zeit — Alle Gruppen neuer als ...
  - GROUP grp — Liste aller Artikel in Gruppe grp
  - NEWNEWS grps date time .. — Liste neuer Artikel in ...
  - ARTICLE n — Gib mir Artikel n
  - POST — Artikel veröffentlichen
  - IHAVE n — Ich habe Artikel n. Möchtest Du ihn?
  - QUIT — Und tschüss.
- Artikel werden lokal abgespeichert
- Alte Artikel werden nach Verfallsdatum gelöscht

Fragen?