Quelle: Computernetzwerke: Der Top-Down-Ansatz, 6. Ausgabe
Jim Kurose, Keith Ross; Pearson, März 2014

Technische Grundlagen der Informatik 2 – Teil 3: Layer 7

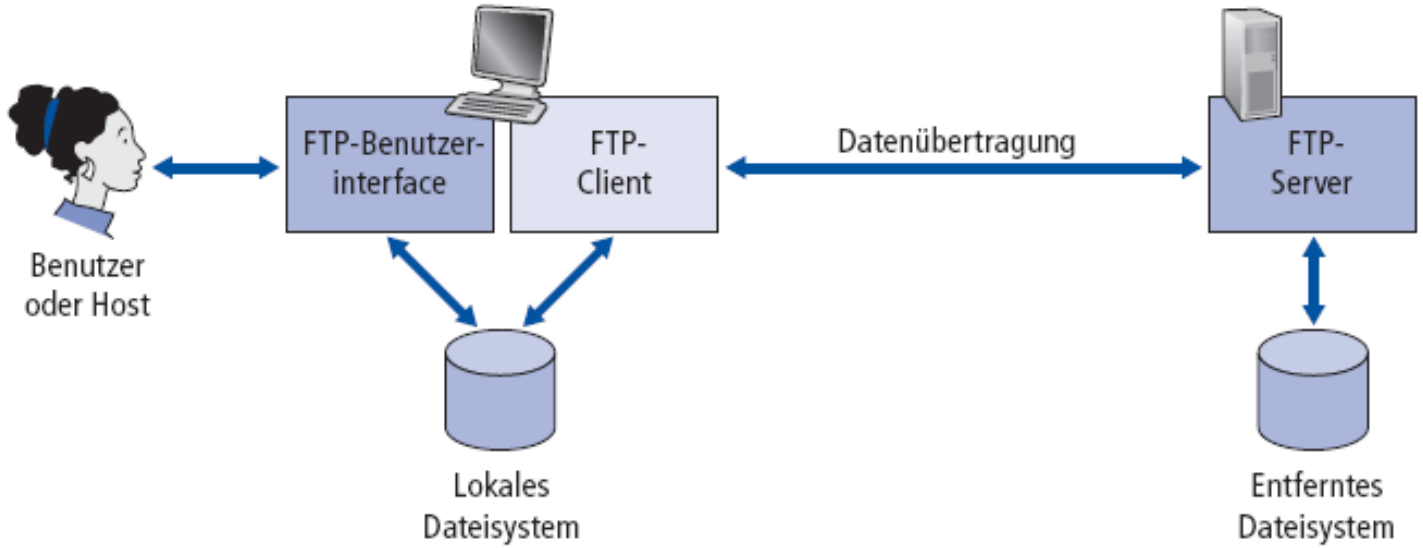
Philipp Rettberg / Sebastian Harnau

Block 4/18

Anwendungsschicht (Layer 7)

FTP: File Transfer Protocol

- Übertragen einer Datei von/zu einem entfernten Rechner
- Client/Server-Modell
- Client: Seite, die den Transfer initiiert (vom oder zum entfernten Rechner)
- Server: entfernter Rechner
- FTP: RFC 959
- FTP-Server: TCP Port 21



FTP: Verschiedene Kanäle für Kontroll- und Datenverbindungen

1. FTP-Client kontaktiert den FTP-Server auf Port 21, wobei er TCP als Transportprotokoll nutzt
2. Client autorisiert sich über die Kontrollverbindung
3. Client betrachtet das entfernte Verzeichnis, indem er Kommandos über die Kontrollverbindung schickt
4. Empfängt der Server ein Kommando für eine Dateiübertragung, öffnet der Server eine TCP-Datenverbindung zum Client
5. Nach der Übertragung einer Datei schließt der Server die Verbindung

- Server öffnet eine zweite TCP-Datenverbindung, um noch eine Datei zu übertragen
- FTP Server hält „Statusinformationen“ vor: aktuelles Verzeichnis, frühere Authentifizierung

FTP: Kommandos & Antworten

Kommandos (als ASCII-Text über die Kontrollverbindung):

- USER *username*
- PASS *password*
- LIST : gibt eine Liste der Dateien im aktuellen Verzeichnis zurück
- RETR *filename* : lädt eine entfernte Datei auf den lokalen Rechner
- STOR *filename* : überträgt eine lokale Datei auf den entfernten Rechner

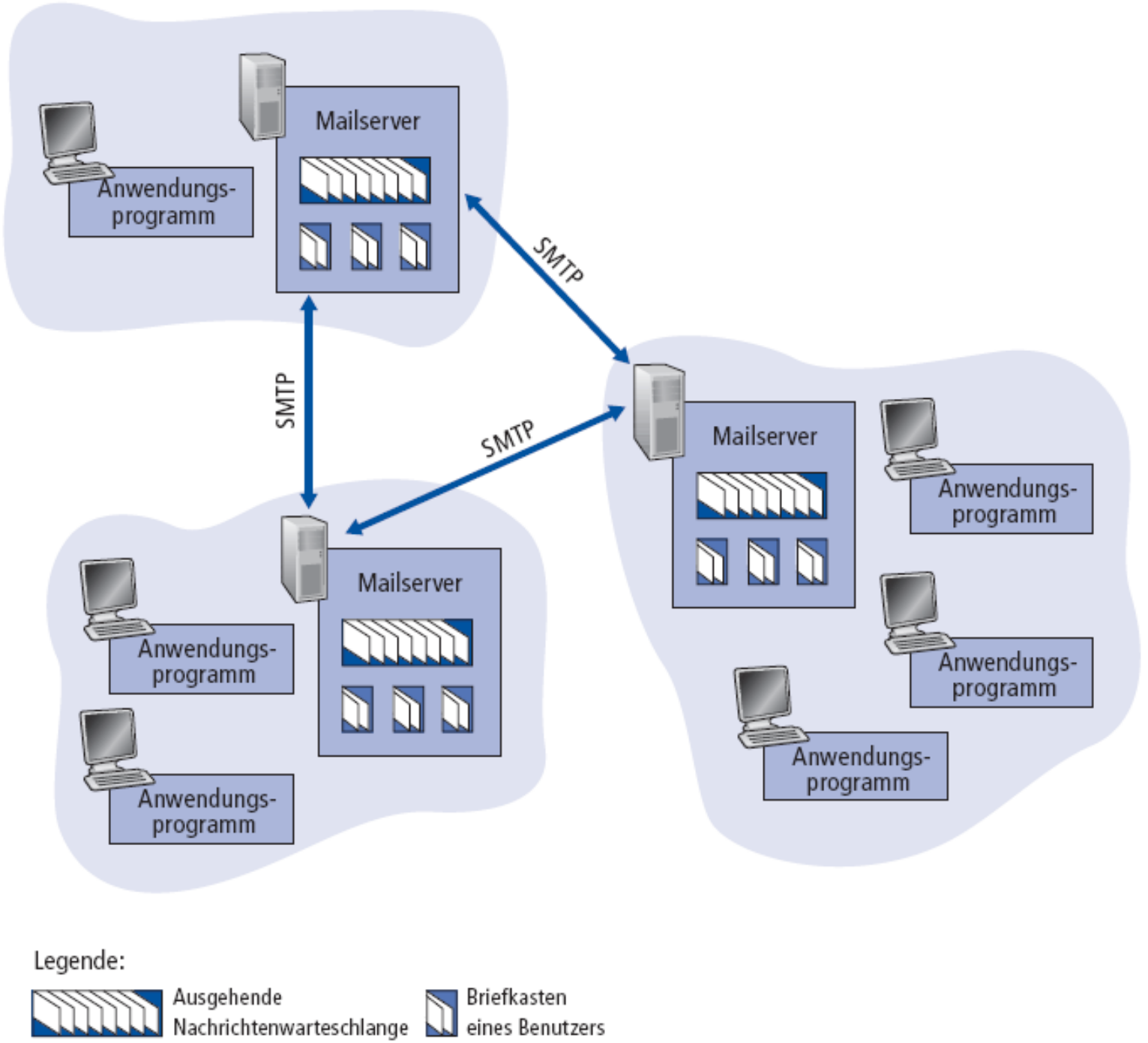
Mögliche Antworten (Statuscode und Erläuterung (wie bei HTTP)):

- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can’t open data connection
- 452 Error writing file

eMail: Electronic Mail

Drei Hauptbestandteile:

- Anwendungsprogramm
- Mailserver
- Übertragungprotokoll: SMTP (Simple Mail Transfer Protocol)

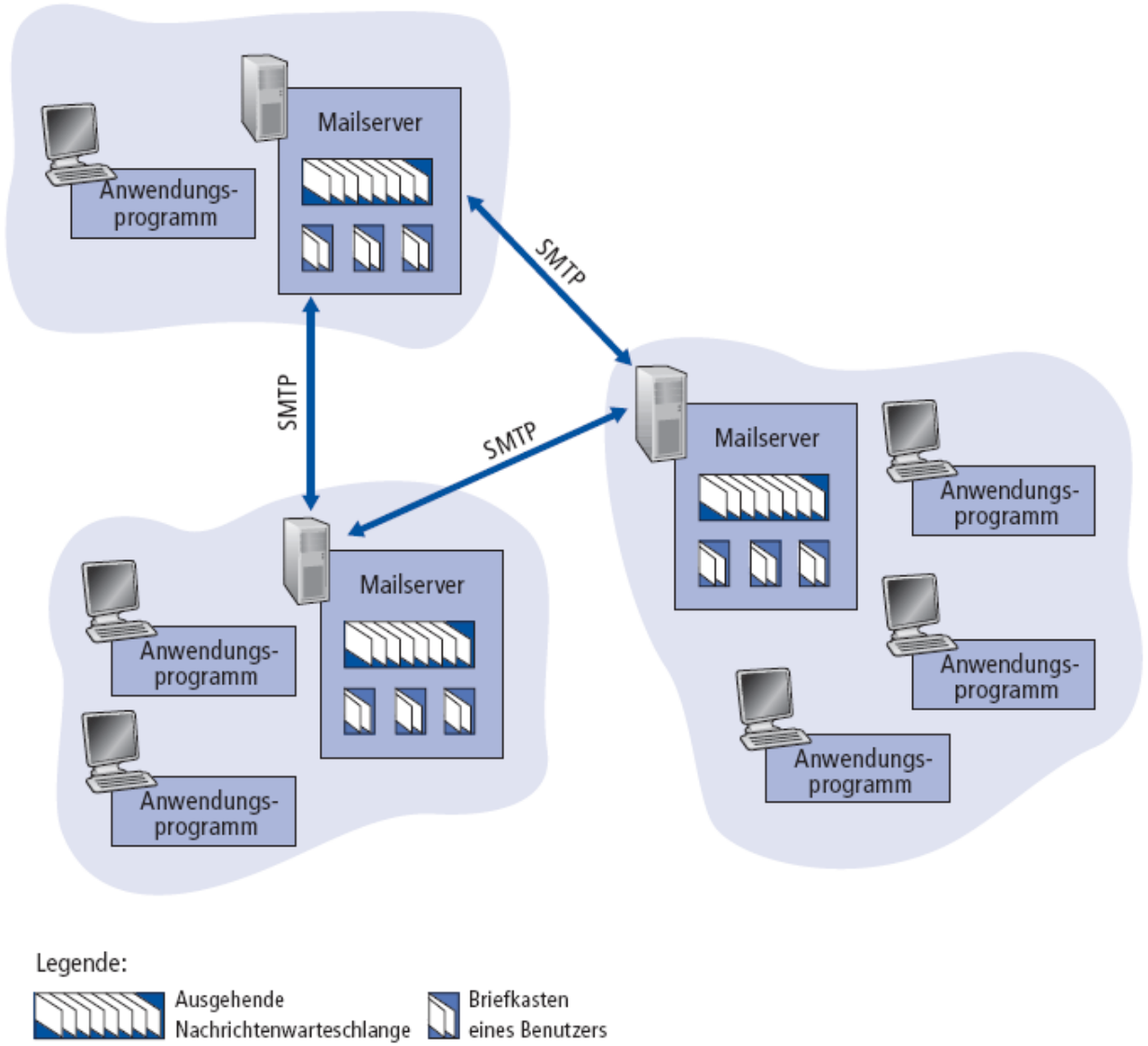


eMail: Anwendungsprogramm

- Auch “Mail Reader” genannt
- Erstellen, Editieren, Lesen von E-Mail-Nachrichten
- z.B. Outlook, Mozilla Thunderbird, Apple Mail, …
- Eingehende und ausgehende Nachrichten werden auf dem Server gespeichert

eMail: Mailserver

- Die Mailbox enthält die eingehenden Nachrichten eines Benutzers
- Die Warteschlange für ausgehende Nachrichten enthält die noch zu sendenden E-Mail-Nachrichten
- SMTP wird verwendet, um Nachrichten zwischen Mailservern auszutauschen
- Client: sendender Mailserver
- Server: empfangender Mailserver

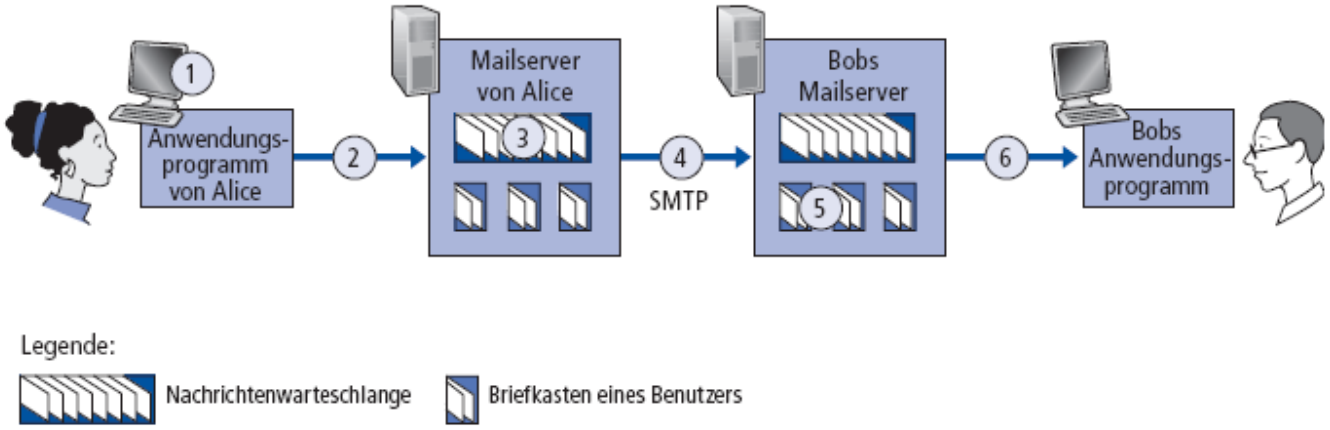


eMail: SMTP

- Ursprüngliche Version aus dem Jahr 1982
- TCP wird zum zuverlässigen Transport von E-Mail-Nachrichten vom Client zum Server (Port 25) verwendet
- Direkter Transport der Nachrichten: vom sendenden Server zum empfangenden Server
- Drei Phasen des Mail-Versands: analog zu einer Unterhaltung
  - Handshaking (Begrüßung)
  - Transfer of Messages (Austausch von Informationen)
  - Closure (Verabschiedung)
- Interaktion basiert auf dem Austausch von Befehlen (Commands) und Antworten (Responses)
  - Command: ASCII-Text
  - Response: Statuscode und Bezeichnung
- *Nachrichten müssen in 7-Bit-ASCII kodiert sein*

eMail: Alice schickt Bob eine Nachricht

1. Alice verwendet ihr Anwendungsprogramm, um eine Nachricht an bob@someschool.edu zu erstellen
2. Alices Anwendung versendet die Nachricht an ihren Mail-Server; Nachricht wird in der Warteschlange gespeichert
3. Alices Mailserver öffnet als Client eine TCP-Verbindung zu Bobs Mailserver
4. SMTP-Client versendet die Nachricht von Alice über die TCP-Verbindung
5. Bobs Mailserver empfängt die Nachricht von Alices Mailserver und speichert diese in Bobs Mailbox
6. Bob verwendet (irgendwann) sein Anwendungsprogramm und liest die Nachricht



eMail: SMTP-Beispiel

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

eMail: Lab SMTP

> telnet <*servername v. Mailserver*> 25

- Der Server sollte mit dem Code 220 antworten
- Eingeben der Befehle HELO, MAIL FROM, RCPT TO, DATA, QUIT

So kann man eine E-Mail ohne Verwendung eines Anwendungsprogramms versenden.

eMail: SMTP-Zusammenfassung

- SMTP verwendet eine dauerhafte Verbindung für den Versand von E-Mails
- SMTP verwendet sowohl für Header als auch für Daten 7-Bit-ASCII
- Ein SMTP-Server verwendet CRLF.CRLF, um das Ende einer Nachricht zu signalisieren

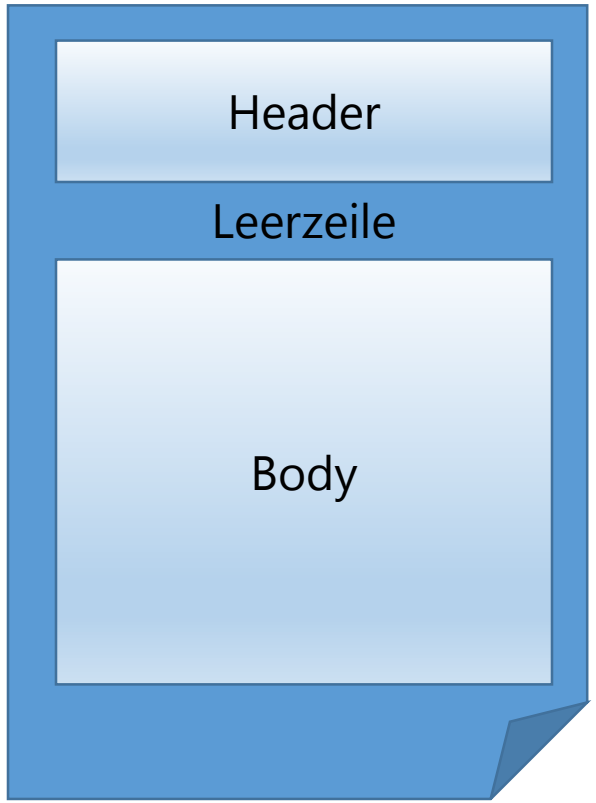
HTML	SMTP
PULL	PUSH
Interaktion mittels ASCII-Befehl/Antwort-Paaren sowie Statuscodes	
jedes Objekt ist in einer eigenen Antwortnachricht gekapselt	mehrere Objekte können in einer Nachricht (multipart msg) versendet werden

eMail: Format der Nachricht

SMTP: Protokoll für den Austausch von E-Mail-Nachrichten

RFC 822: Standard für Textnachrichten:

- Header-Zeilen, z.B.
  - To:
  - From:
  - Subject:
  - Keine SMTP-Befehle!
- Body
  - Die eigentliche Nachricht in ASCII



eMail: Multimedia-Erweiterung (MIME)

- MIME: Multimedia Mail Extension, RFC 2045, 2056
- Zusätzliche Zeilen im Header deklarieren den MIME-Typ des Inhaltes:
  - MIME-Version
  - Methode, die zum Kodieren der Daten verwendet wurde
  - Deklaration der Datentypen und Untertypen
  - Codierte (Multimedia-)Daten

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding:
base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
```

MIME: Datentypen

Text
- Beispiele für Subtypen:
  - plain
  - html

Bilder
- Beispiele für Subtypen:
  - jpeg
  - gif
  - png

Audio
- Beispiele für Subtypen:
  - basic (8-bit mu-law encoded),
  - 32kadpcm (32 kbps coding)

Video
- Beispiele für Subtypen:
  - mpeg
  - quicktime

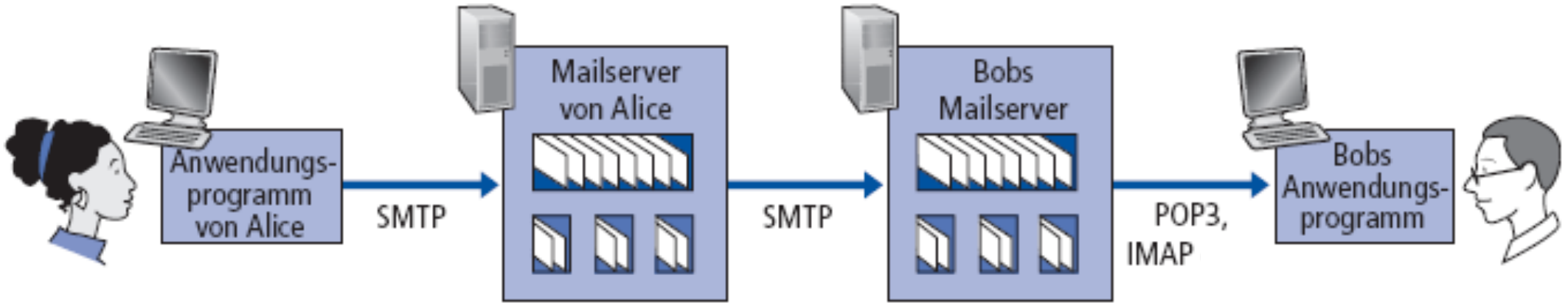
Anwendungen
- Daten müssen von der Anwendung vor der Wiedergabe interpretiert werden
- Beispiele für Subtypen:
  - Msword
  - octet-stream

eMail: Multipart-Typ

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=StartOfNextPart

--StartOfNextPart
Dear Bob, Please find a picture of a crepe.
--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.........................
......base64 encoded data
--StartOfNextPart
Do you want the recipie?
```

eMail: Zugriffsprotokolle



Zugriffsprotokoll: Protokolle zum Zugriff auf E-Mails
Abruf vom Server:

- POP: Post Office Protocol [RFC 1939]
  - Autorisierung (Anwendung <--> Server) und Zugriff/Download
- IMAP: Internet Mail Access Protocol [RFC 1730]
  - Größere Funktionalität (deutlich komplexer)
  - Manipulation der auf dem Server gespeicherten Nachrichten

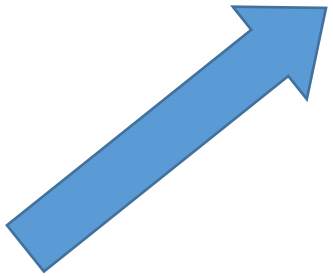
eMail: Beispiel POP3-Protokoll

Autorisierungsphase:

- Befehle des Clients:
  - user: Benutzername
  - pass: Passwort
- Antworten des Servers:
  - +OK
  - -ERR

Transaktionsphase:

- list: Nachrichten auflisten
- retr: Nachrichten herunterladen
- dele: Löschen von Nachrichten
- quit: Ende



```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 2 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

eMail: POP3 vs. IMAP

Mehr zu POP3

- Vorheriges Beispiel nutzte den “Download-and-Delete”-Modus, d.h., andere E-Mail- Clients haben danach keine Möglichkeit mehr, die Mails zu lesen
- Der “Download-and-Keep”-Modus ermöglicht den reinen Lesezugriff auf Nachrichten, d.h. verschiedene Clients haben Zugriff
- POP3 ist zustandslos zwischen einzelnen Sitzungen

IMAP

- Alle Nachrichten bleiben an einem Ort: auf dem Server
- Nachrichten können auf dem Server in Ordnern verwaltet werden
- IMAP bewahrt den Zustand zwischen einzelnen Sitzungen:
  - Namen von Ordnern und Zuordnung von Nachrichtennummer und Ordnername bleiben erhalten

eMail: Push-Dienste

- Alle Clients fragen den Server regelmäßig, ob es neue Nachrichten gibt. Das verursacht unnötig Last, ohne das neue Informationen vorlägen.
- Fragt ein Programm alle 5 Minuten nach neuen Mails und Sie erhalten 20 Mails am Tag, so sind mindestens 93% der Anfragen ergebnislos und damit überflüssig.
- Besser: Der Client registriert sich mit seiner aktuellen IP/Port-Nummer und der Server sagt bescheid, wenn es neues gibt.
- Beispiele:
  - Push-IMAP (P-IMAP): Entwickelt von Oracle, basiert auf IMAP
  - Yahoo! Mail benutzt eine spezielle UDP-Nachricht, um eine eMail-Sychronisation anzustoßen.
  - Apples push email benutzt eine Variante von XMPP (Extensible Messaging and Presence Protocol)

Quelle: http://en.wikipedia.org/wiki/Push-IMAP
Quelle: http://en.wikipedia.org/wiki/Push_email

eMail: Spam

Spam/Junk
- unerwünschte elektronisch übermittelte Nachrichten, die dem Empfänger unverlangt zugestellt werden und häufig werbenden Inhalt enthalten

Auswirkungen
- Aussortieren und Lesen von Spam kostet Arbeitszeit.
- Leitungsbelastung/Übertragungsvolumen
- Spam belegt Speicherplatz (Mailboxen/Archiv/Backups)

eMail: Erkennung von Spam

Spam-Mails weisen einige Besonderheiten auf, die es ermöglichen, diese automatisiert zu klassifizieren:

- Ggf. syntaktische Fehler im Aufbau des Headers bzw. des Bodies (Malformed Mail)
- Auswertung der in der Mail enthaltenen Formate, Worte bzw. Textbausteine
- Bewertung der in einer Mail vorhandenen Links
- Bewertung der „Absender“ der Mails

Diese verschiedenen „Faktoren“ können durch Hilfsprogramme wie z.B. „Spamassassin“ einen Spam-Score für eine Mail berechnen, der dann z.B. in den Mail-Header (X-Spam-Flag) eingefügt werden kann.

```
X-Spam-Level: **
X-Spam-Status: No, score=2.8
required=7.0 tests=FORGED_MUA_OUTLOOK,
HTML_MESSAGE,MIME_HTML_MOSTLY,RCVD_IN_DN
SWL_NONE autolearn=no version=3.3.2
X-Spam-Checker-Version: SpamAssassin
3.3.2 (2011-06-06) on <server>
```

Der Mailserver ist dann in der Lage, diesen Header bei der Auslieferung der Mail in die Mailbox des Benutzers zu berücksichtigen und die Mail in einen anderen Ordner zu sortieren oder sogar gleich zu löschen.

eMail: Vermeidung von Spam

Im Gegensatz zur automatischen Klassifizierung von Spam ist ein Ansatz zu bevorzugen, der einen großen Teil der potentiellen Spam-Mails beim serverseitigen Empfang gleich abweist.

DNSBL:

- eine DNS-based Blackhole List bietet eine Echtzeit-Abfrage einer Blacklist von Spam-Absendern, so dass der Mailserver diese Mail ggf. direkt abweisen kann

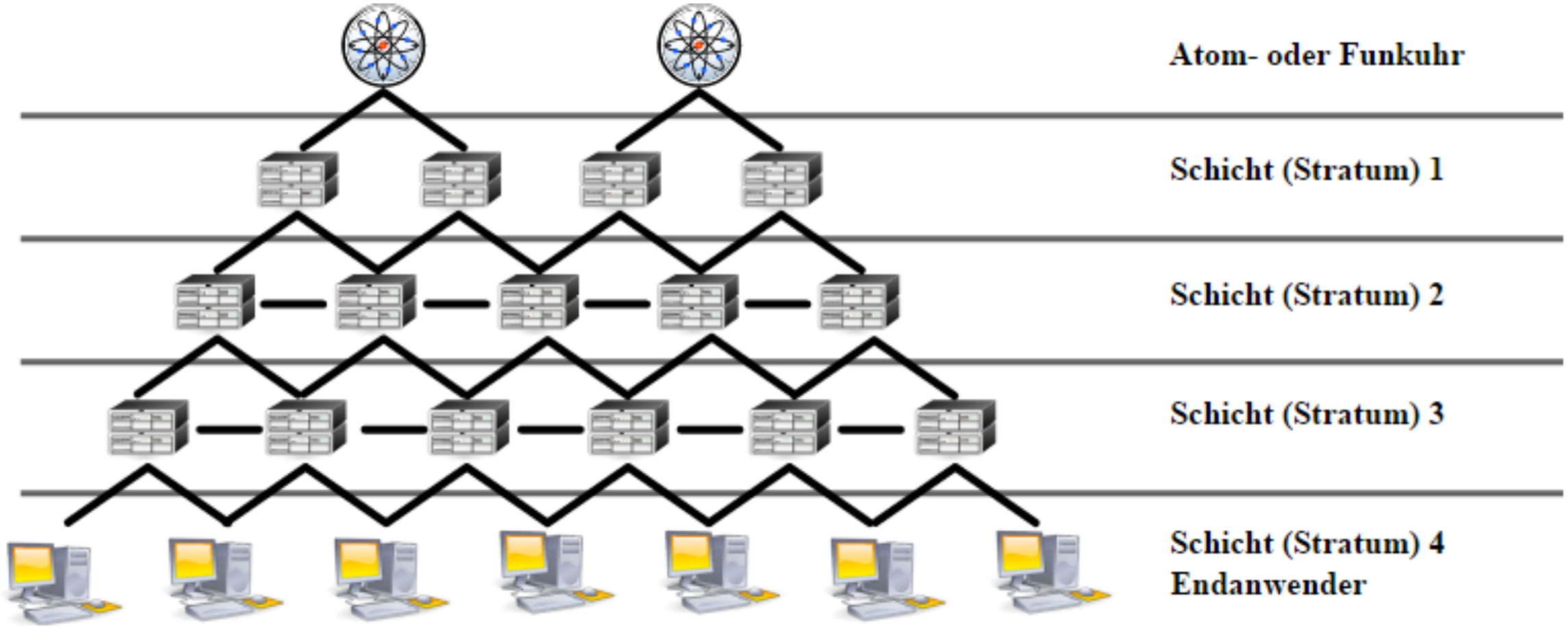
Greylisting:

- das Greylisting weist die erste Mail eines unbekannten Absenders ab
- beim zweiten „Zustellversuch“ wird diese Mail und alle zukünftigen angenommen
- Massenversender verzichten oft auf eine erneute Zustellung, so dass diese Spam-Mails den Empfänger nicht belasten

Zeitsynchronisation: NTP

- UDP, Port 123
- Synchronisiert die Lokale Uhr mithilfe externer Zeitsignale
- Korrigiert „Phase“ und „Frequenz“ des lokalen Zeitgebers für eine möglichst stabile Zeit auch zwischen den Synchronisierungszeitpunkten
- Hierarchische Strukturen (Client der darüber liegenden Schicht ist normalerweise ebenso Server für die darunterliegenden Clients)

Zeitsynchronisation: NTP



DNS: Domain Name System

Adressierung von Internet-Hosts:

- IP-Adresse (32 Bit) – für die Adressierung in Paketen
- “Name”, z.B., www.yahoo.com – von Menschen verwendet

Domain Name System:

- Verteilte Datenbank, implementiert eine Hierarchie von Nameservern
- Protokoll der Anwendungsschicht, wird von Hosts verwendet, um Namen aufzulösen (Abbildung zwischen Adresse und Name)
- zentrale Internetfunktion, implementiert als Protokoll der Anwendungsschicht
- Grund: Komplexität nur am Rand des Netzwerkes!

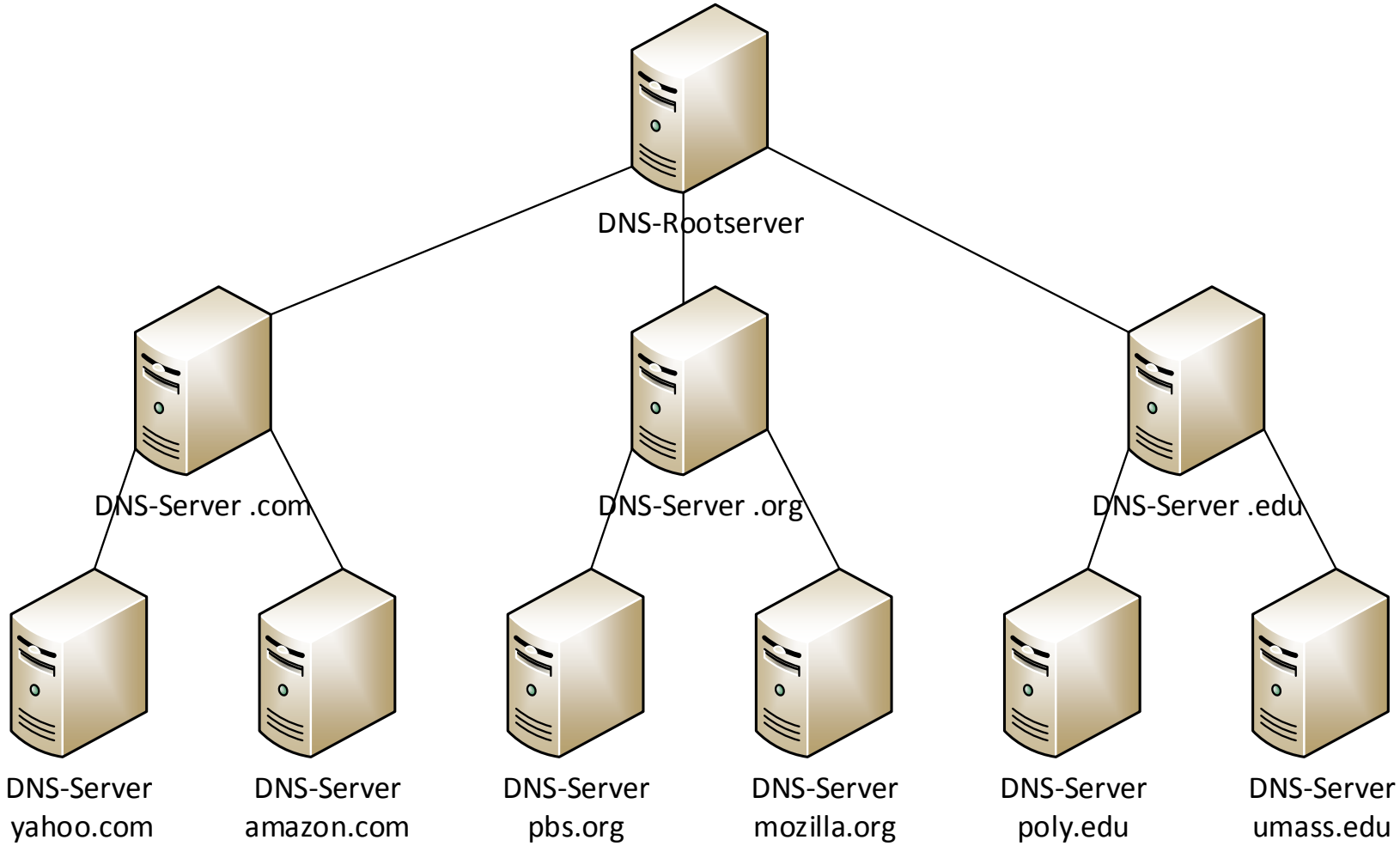
DNS: Funktionen und Dienste

- Übersetzung von Hostnamen in IP-Adressen
- Aliasnamen für Hosts
- Kanonische Namen und Aliasnamen
- Aliasnamen für Mailserver
- Lastausgleich
- Replizierte Webserver: mehrere IP-Adressen von einem kanonischen Namen

DNS: verteilte, hierarchische Datenbank

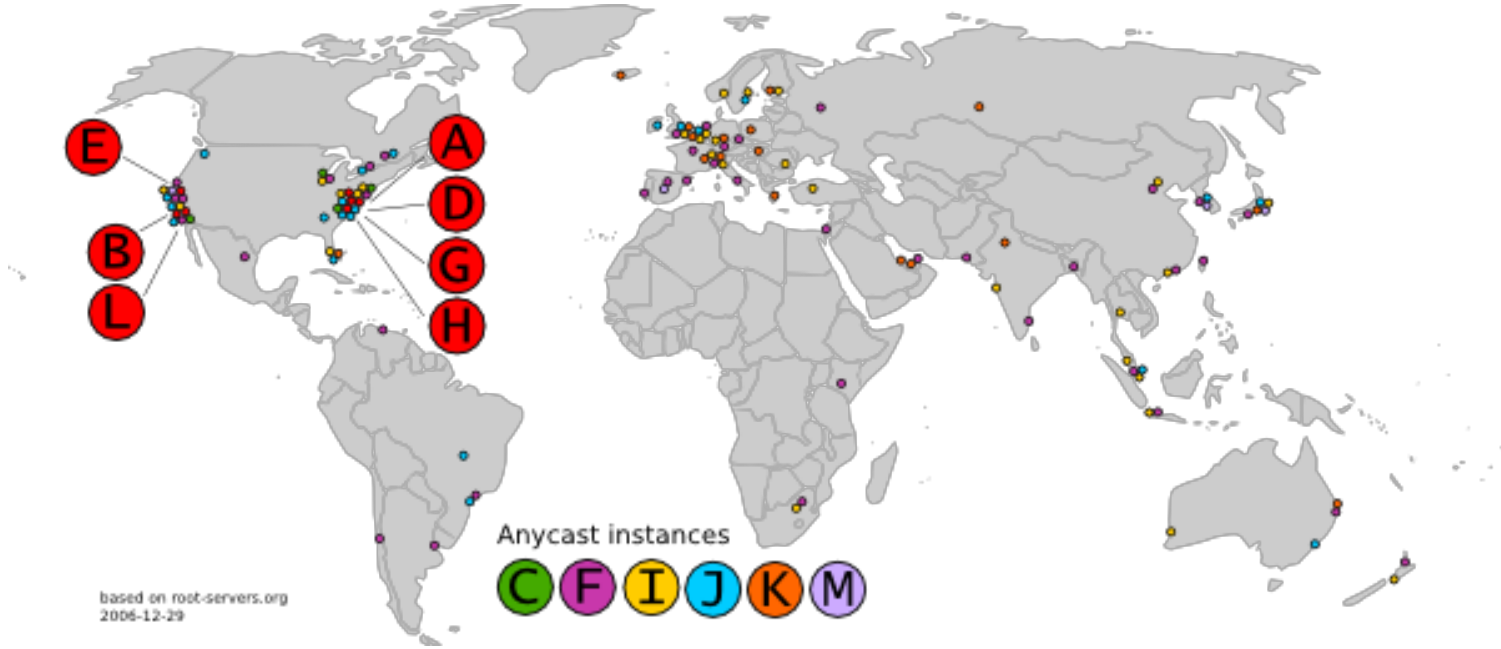
Client sucht die IP-Adresse von www.amazon.com – erste Annäherung:

- Client fragt seinen lokalen DNS-Server
- dieser fragt einen DNS-Rootserver, um den DNS-Server für com zu finden
- danach fragt er den com-DNS-Server, um den amazon.com-DNS-Server zu finden
- dann wird der amazon.com-DNS-Server gefragt, um die IP-Adresse zu www.amazon.com zu erhalten



DNS: Root-Nameserver

- wird vom lokalen Nameserver kontaktiert, wenn dieser einen Namen nicht auflösen kann
- kennt die Adressen der Nameserver der Top-Level-Domains (com, net, org, de, uk, …)
- gibt diese Informationen an die lokalen Nameserver weiter



- 13 logische Root-Nameserver
- aktuell weit über 500 tatsächliche Server aktiv in Betrieb (Stichwort: Anycast)

DNS: TLD- und Autoritative Server

Top-Level-Domain (TLD)-Server:

- verantwortlich für com, org, net, edu etc. sowie für alle Länder-Domains, z.B. de, uk, fr, ca, jp
- Network Solutions ist verantwortlich für den com-TLD-Server
- Educause hat die Verantwortung für den edu-TLD-Server

Autoritativer DNS-Server:

- DNS-Server einer Organisation, der eine autorisierte Abbildung der Namen dieser Organisation auf IP-Adressen anbietet
- verwaltet von der entsprechenden Organisation oder einem Service Provider

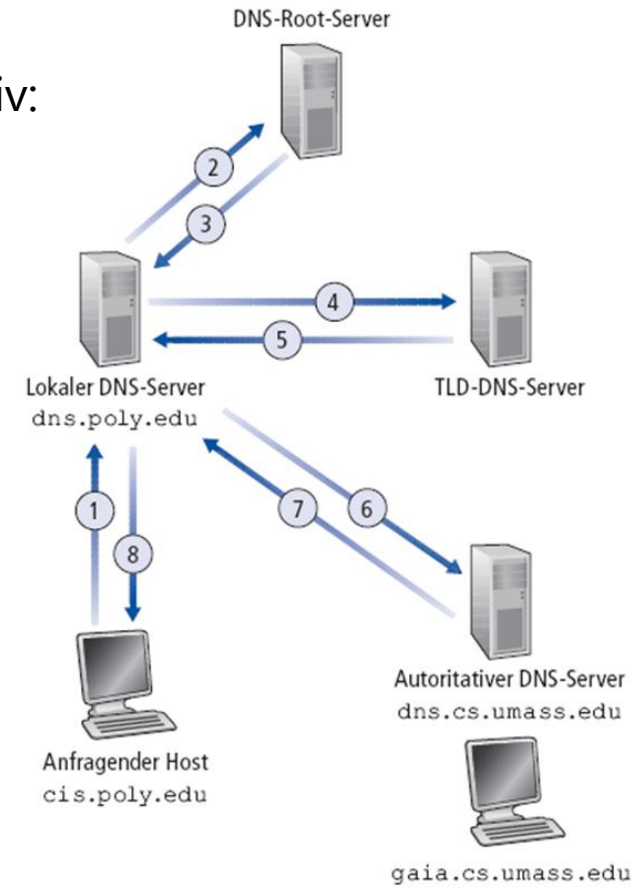
DNS: lokale Nameserver

- gehören nicht zur Hierarchie der DNS-Server
- jeder ISP (ISP für Privatkunden, Firmen, Universität) besitzt einen lokalen Nameserver
- werden auch “Default-Nameserver” genannt
- wenn ein Host eine DNS-Anfrage startet, dann schickt er diese an seinen lokalen Nameserver
  - dieser kümmert sich um die Anfrage so lange, bis eine endgültige Antwort vorliegt
  - dazu kontaktiert er bei Bedarf Root-Nameserver, TLD-Nameserver und autoritative Nameserver
  - dann schickt er die Antwort an den Host zurück

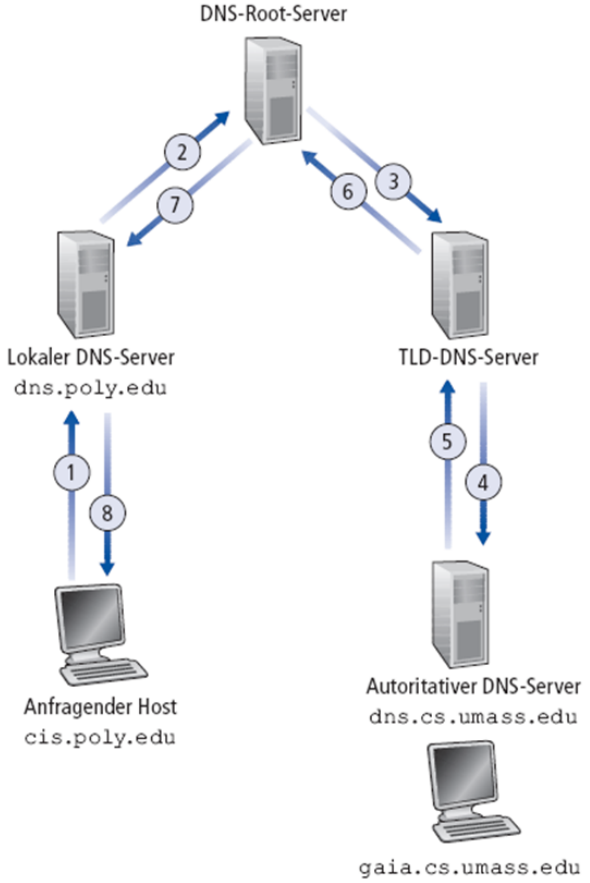
DNS: Beispiel Namensauflösung

Host cis.poly.edu fragt nach der IP-Adresse von gaia.cs.umass.edu

iterativ:



rekursiv:



DNS: Caching

- Sobald ein Nameserver eine Abbildung zur Namensauflösung kennenlernt, merkt er sich diesen in einem Cache
  - Die Einträge im Cache werden nach einer vorgegebenen Zeit wieder gelöscht
  - Die Adressen der TLD-Server werden üblicherweise von den lokalen Nameservern gecacht
    - Root-Nameserver werden eher selten angesprochen

- Mechanismen zur Pflege von Cache-Einträgen und zur Benachrichtigung bei Änderungen werden derzeit von der IETF entwickelt
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS: Ressource Records

Ressource Record Format: `(name, value, typ, ttl)`

Typ=A

- name ist der Hostname
- value ist die IP-Adresse

Typ=NS

- name ist eine Domain (z.B. foo.com)
- value ist der Hostname des autoritativen Nameservers für diese Domain

Typ=CNAME

- name ist ein Alias für einen kanonischen (echten) Namen:
- www.ibm.com ist ein Alias für servereast.backup2.ibm.com
- value ist der kanonische Name

Typ=MX

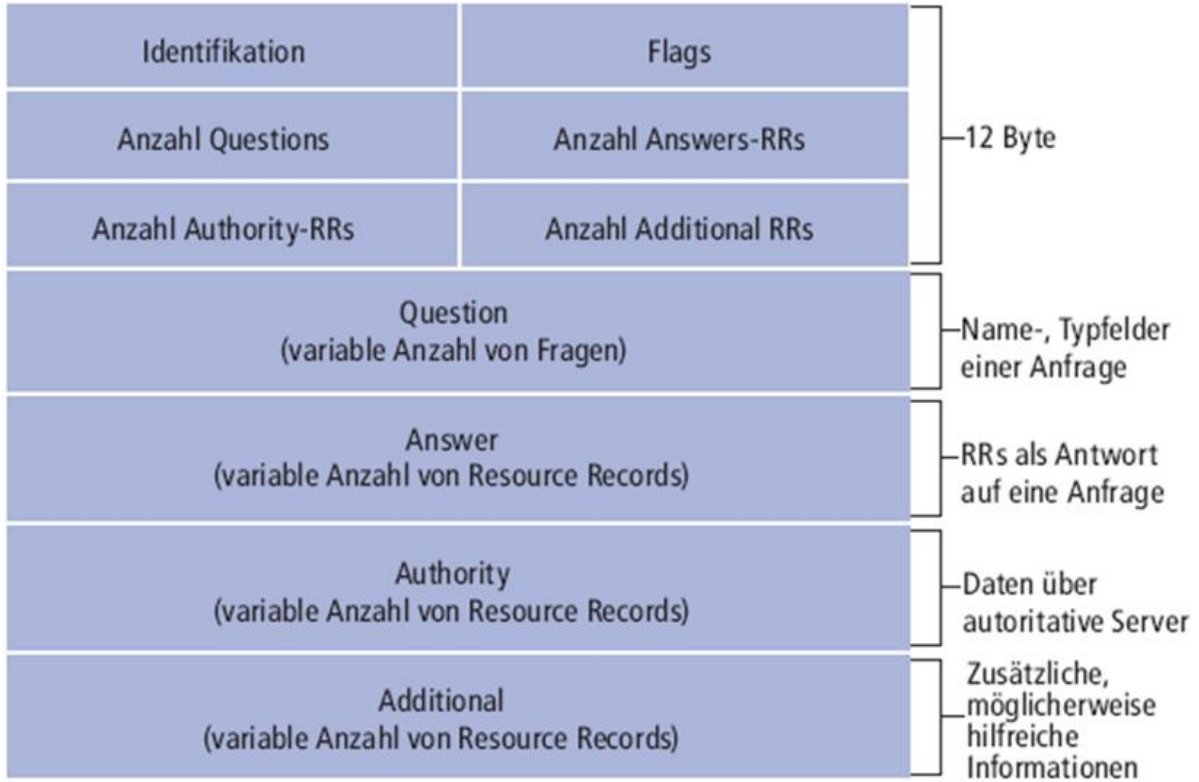
- value ist der Name des Mailservers für die Domain name

DNS: Nachrichtenformat

DNS-Protokoll: Query- und Reply-Nachrichten, beide mit demselben Nachrichtenformat

Header-Felder:

- identification: 16-Bit-ID, wird für die Query-Nachricht vergeben, die Reply-Nachricht verwendet dieselbe ID
- flags:
  - query/reply
  - recursion desired
  - recursion available
  - reply is authoritative



DNS: Lab

https://dnsquery.org

- Wie viele Mailserver sind für „nordakademie.de“ veröffentlicht?
- Wie lauten die autoritativen Nameserver von nordakademie.de?