



# Systeme II

## 6. Die Anwendungsschicht

Christian Schindelhauer

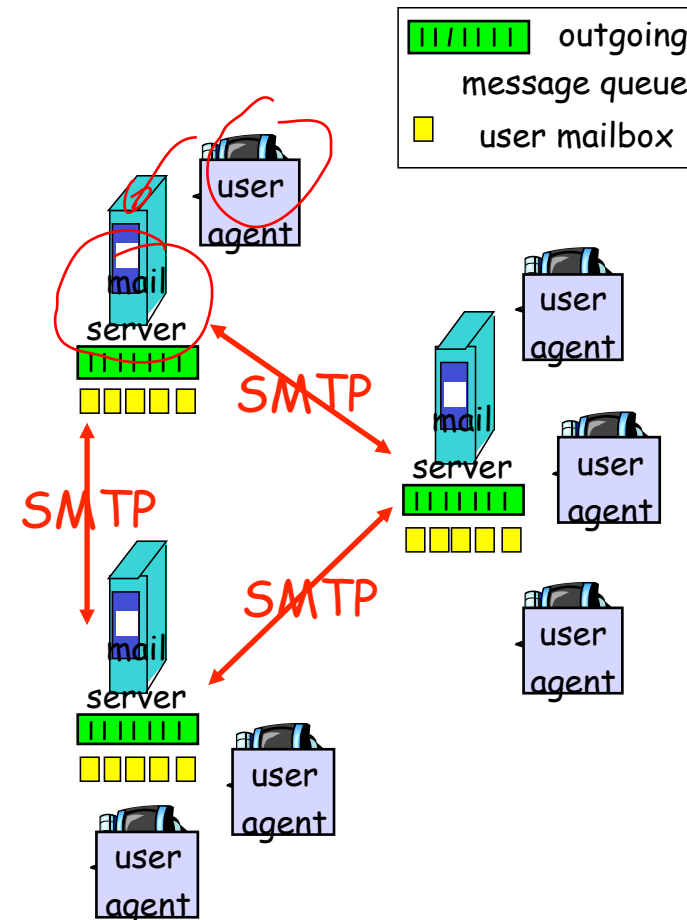
Technische Fakultät

Rechnernetze und Telematik

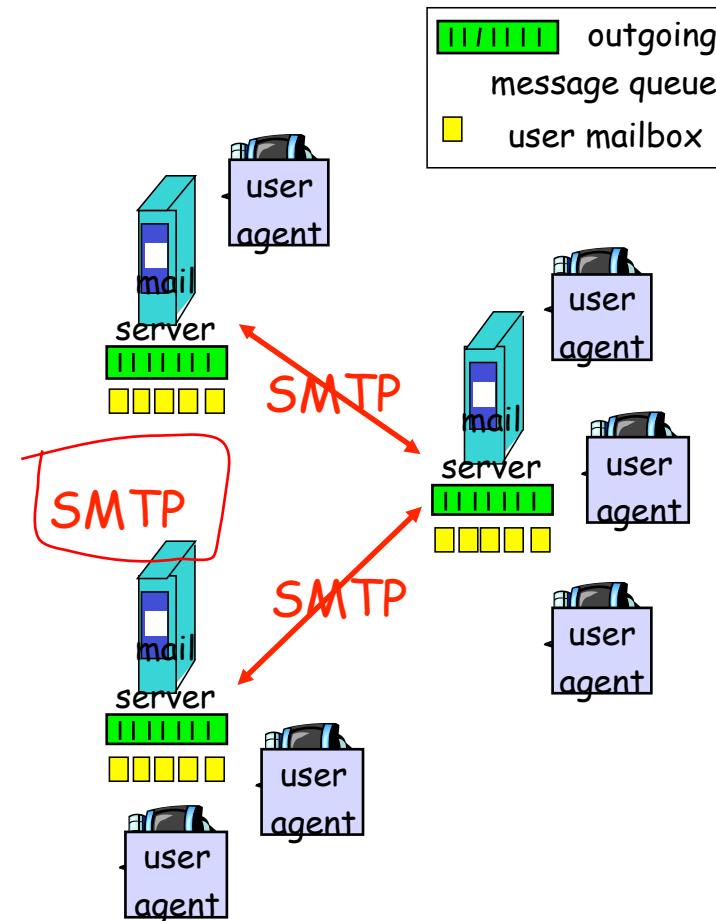
Albert-Ludwigs-Universität Freiburg

Version 30.06.2014

- Hauptkomponenten
  - user agents
  - mail servers
  - simple mail transfer protocol: SMTP
- User Agent
  - Mail Client
  - ⓐ Erstellen, ändern und lesen von E-Mail-Nachrichten
  - ⓐ z.B. Eudora, Outlook, pine, Mozilla Thunderbird
  - ⓐ abg ehende und ankommende Nachrichten werden auf dem Server gespeichert



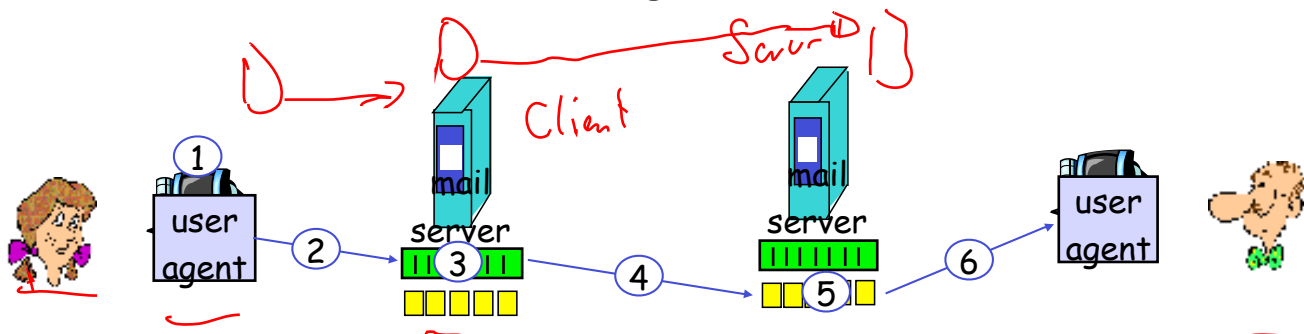
- Mailbox speichert eingehende Nachrichten für den User
- Nachrichten-Warteschlange (queue) der zu versendenden Nachrichten
- SMTP-Protocol zwischen Mail-Servern um E-Mail-Nachrichten zu schicken



- verwendet TCP um zuverlässig E-Mail-Nachrichten vom Client auf Port 25 zu verschicken
- Direkte Übertragung von Absender-Server zum Empfangs-Server
- 3 Phasen in der Übertragung
  - Handshake
  - Transfer der Nachricht
  - Abschluss
- Befehle und Antwort
  - Befehle als ASCII text
  - Antwort: Status-Code und Kurzbeschreibung
- Nachrichten sind in 7-bit ASCII

# Beispiel: Alice sendet eine Nachricht an Bob

- 1) Alice verwendet UA um die Nachricht zu erzeugen mit Eintrag "to" bob@some school.edu
- 2) Alice UA sendet die Nachricht zu ihren Mail-Server
  - Nachricht wird in der Nachrichtenwartenschlange platziert
- 3) Client-Seite des SMTP öffnet TCP-Verbindung mit Bobs Mail-Server
- 4) SMTP Client sendet Alice Nachricht über die TCP-Verbindung
- 5) Bobs Mail-Server schreibt die Nachricht in Bobs Mailbox
- 6) Bob ruft seinen User Agent auf, um die Nachricht zu lesen

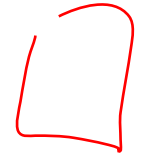


# Beispiel SMTP Interaktion

FROM  
[>FROM]  
[?FROM]

```
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
```

Server



## ■ SMTP

- verwendet persistente Verbindungen
- verlangt Nachrichten (header & body) in 7-bit ASCII
- SMTP-Server verwenden „CRLF.CRLF“ um das Ende einer Nachricht zu beschreiben

## ■ Vergleich mit HTTP:

- HTTP: pull
- SMTP: push
- beide haben ASCII Befehls- und Antwort-Interaktion und Status-Codes

## ■ HTTP

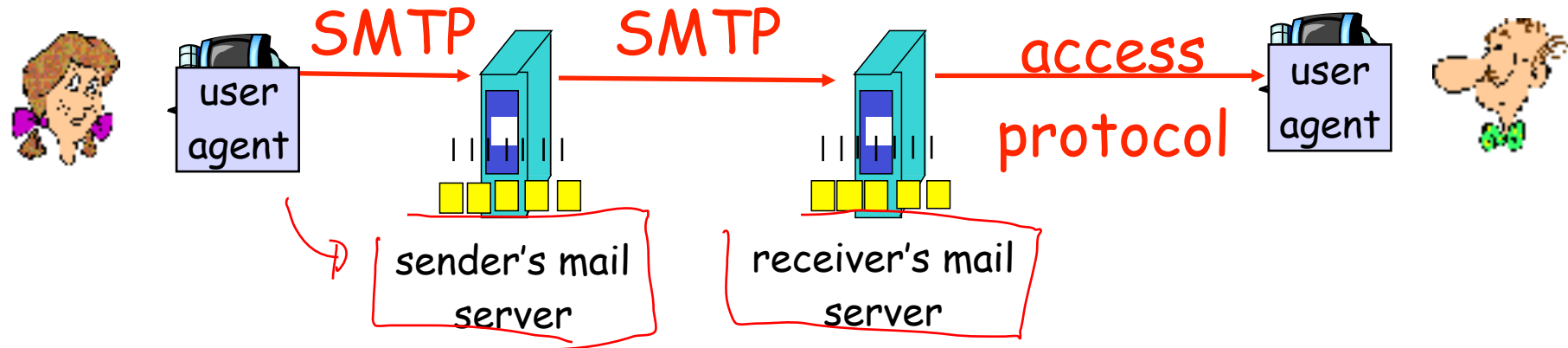
- jedes Objekt wird in eigener Nachricht verpackt

## ■ SMTP

- ~~verschiedene~~ Objekte werden in einer Multipart-Nachricht verschickt



# Mail-Zugriffsprotokolle



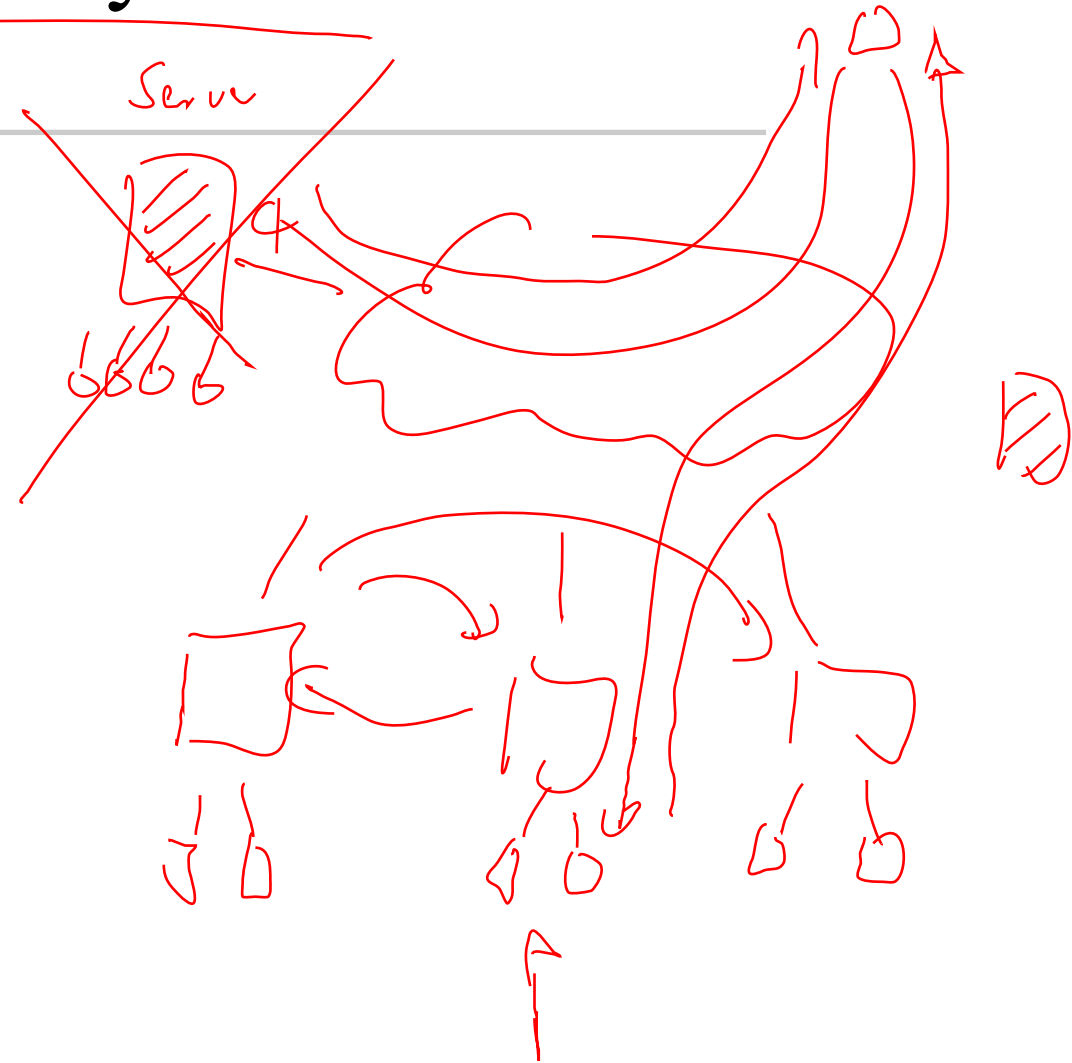
- SMTP: Auslieferung und Speicher zum Server des Empfängers
- Mail-Zugriffsprotokolle: E-Mail-Abruf vom Server
  - POP: Post Office Protocol [RFC 1939]
    - Authentifizierung (zwischen Agent und Server) und Download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - mehr Features und komplexer
    - Bearbeitung von gespeicherten Nachrichten **auf dem Server**
  - HTTP gmail, Hotmail, Yahoo! Mail, web.de, etc.



- POP3 (Post-Office-Protocol)
  - User kann im “download and delete” Modus E-Mails einmalig herunterladen
  - User kann E-Mails noch einmal lesen, wenn er den Client wechselt:
    - “Download-and-keep”: Kopien der Nachricht auf verschiedenen Clients
  - POP3 ist zustandslos (stateless) von einer Sitzung zur nächsten
- IMAP (Internet Message Access Protocol)
  - hält alle Nachrichten an einem Ort: dem Server
  - erlaubt dem User die Nachrichten in Ordnern zu organisieren
  - IMAP speichert den Benutzer-Status zwischen Sitzungen
  - Namen der Ordner und Zuordnung zwischen Nachrichten-ID und Ordernamen

# Meilensteine P2P Systeme

- Napster 1999-2000 
  - Filesharing, nur rudimentäres P2P
- Gnutella 2000
  - 1. echtes P2P-Netzwerk
- Edonkey 2000
  - Mehr Filesharing als P2P
- ○ FreeNet 2000
  - Anonymisiertes P2P-Netzwerk
- FastTrack 2001
  - KaZaa, Morpheus, Grokster
- ⊗ ○ Bittorrent 2001
- ⊗ ■ Skype 2003
  - VoIP (voice over IP), Chat, Video



- Distributed Hash-Tables (DHT) (1997)

- Ziel: Lastbalancierung für Web-Server

- CAN (2001)

- DHT-Netzwerk-Struktur

- Chord (2001)

- Erstes effiziente P2P-Netzwerk
- Logarithmische Suchzeit

- Pastry/Tapestry (2001)

- Effizientes verteiltes P2P-Netzwerk unter Verwendung des Plaxton-Routing

- Und viele andere Ansätze

- Viceroy, Distance-Halving, Koorde, Skip-Net, P-Grid, ...

- In den letzten fünf Jahren:

- Network Coding for P2P
- Game theory in P2P
- Anonymity, Security

Kademlia

Bittorrent