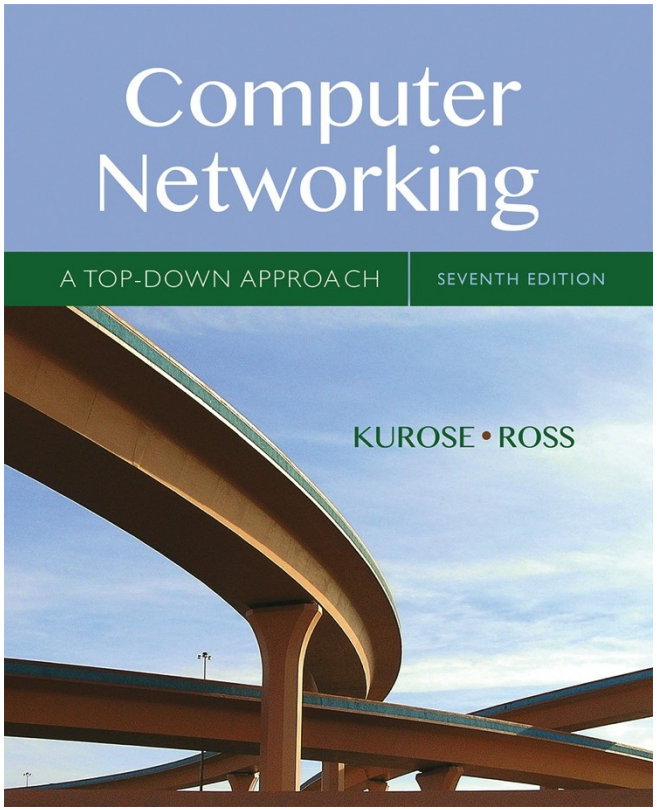
Chapter 2
Application Layer



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

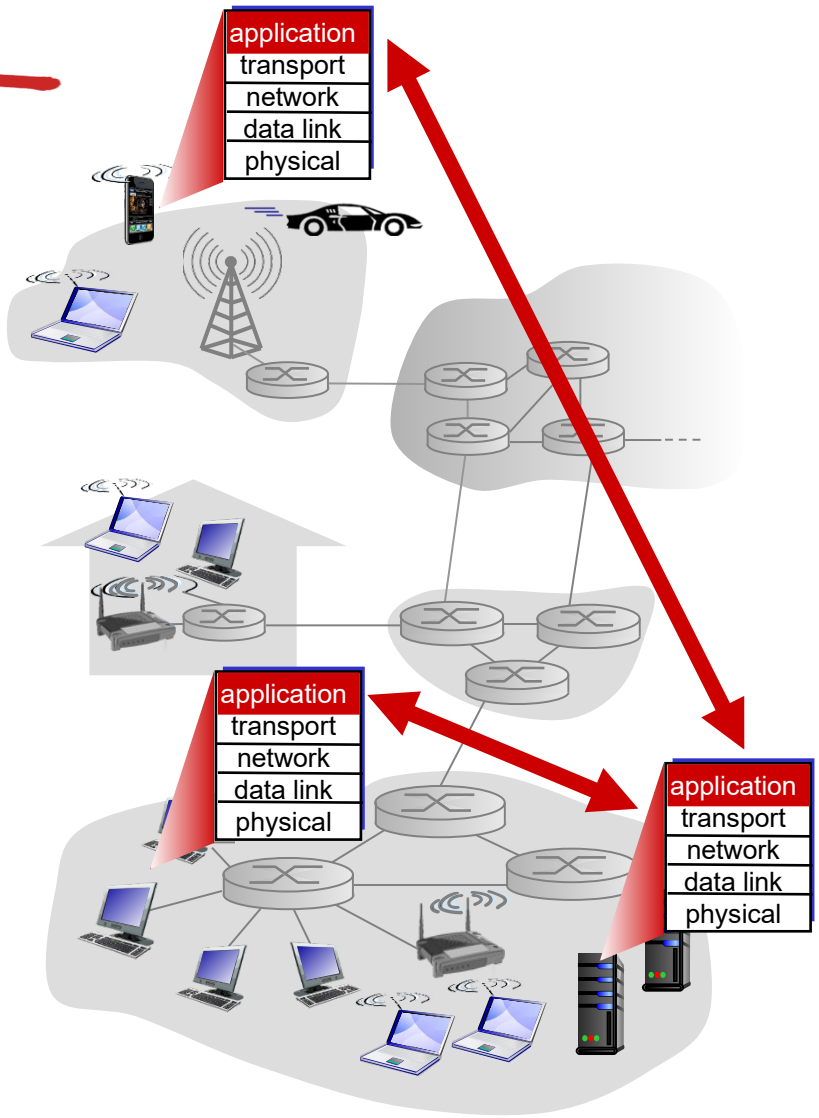
Netzanwendung entwickeln

Programme, die:

- auf (unterschiedlichen) *Endsystemen* laufen
- über das Netzwerk kommunizieren
- z.B., Webserver-Software kommuniziert mit Browser-Software

Keine Software für Komponenten im Kern des Netzwerks (*network-core*)

- auf *core devices* laufen keine User-Anwendungen
- schnelle Anwendungsentwicklung (RAD, rapid application development) und -verbreitung



Prozesse kommunizieren…

Prozess: Programm, läuft auf einem Host

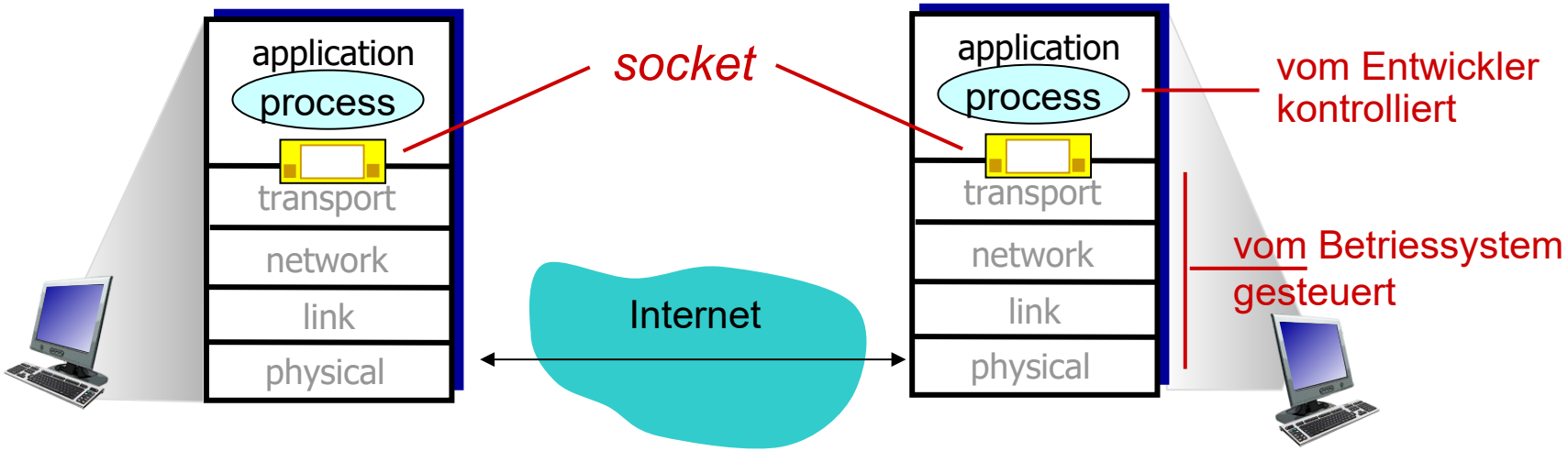
- Prozesse auf verschiedenen Hosts: Austausch von Nachrichten

Clients, Server

client process: initiiert die Kommunikation

server process: wartet auf Kontaktanfragen

… über Sockets



Prozesse adressieren

- für Empfang von Nachrichten: eindeutige *Kennung* (*identifier*)!
- Endgeräte haben eindeutige IP-Adresse (32-Bit oder 128-Bit)
- *F:* reicht IP-Addresse eines Hosts, um einen darauf laufenden Prozess zu identifizieren?
  - *A:* Nein, es können *viele* Prozesse auf ein und demselben Host laufen

- *identifier* beinhaltet sowohl IP-Adresse als auch Portnummern
- Beispiele für Portnummern:
  - HTTP-Server: 80
  - Mailserver: 25
- für HTTP-Nachricht an gaia.cs.umass.edu Webserver:
  - IP-Adresse: 128.119.245.12
  - Portnummer: 80

Protokolle der Anwendungsschicht definieren…

- den Typ der ausgetauschten Nachrichten
  - z.B. Request, Response
- die Syntax der Nachrichten
  - was steht wie in den Nachrichten
- die Semantik der Nachrichten
  - was bedeutet die Information in den Nachrichten
- Regeln für das Senden und Empfangen von Nachrichten

Es gibt offene Protokolle:

- definiert und standardisiert in RFCs
- erlaubt Kompatibilität verschiedener Anwendungen
- z.B., HTTP, SMTP

und proprietäre Protokolle:

- z.B., Skype

Anwendung	Protokoll der Anwendungsschicht
e-mail	SMTP [RFC 2821]
remote access	Telnet [RFC 854]
Web	HTTP [RFC 2616]
file transfer	FTP [RFC 959]

Web und HTTP

Webseiten bestehen aus aus *Objekten*

- z.B. HTML- oder Audiodatei, JPEG-Grafik, …
- jedes Objekt wird adressiert durch einen *URL (uniform resource locator),* z.B.,

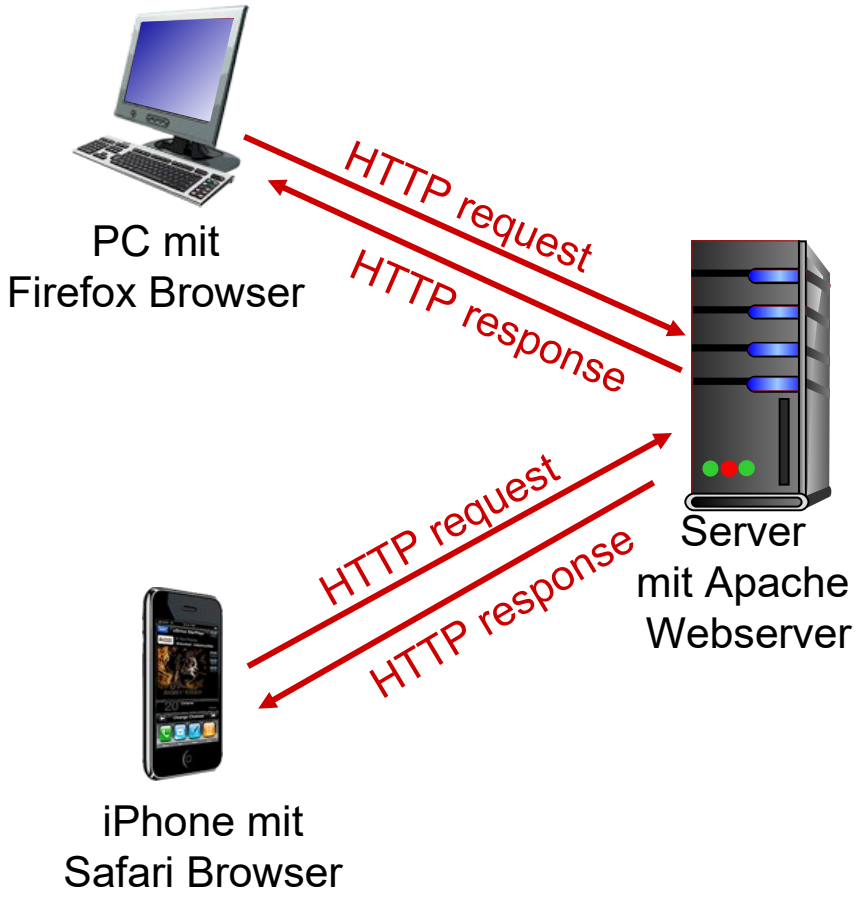
`www.someschool.edu/someDept/pic.gif`

host name | path name

HTTP

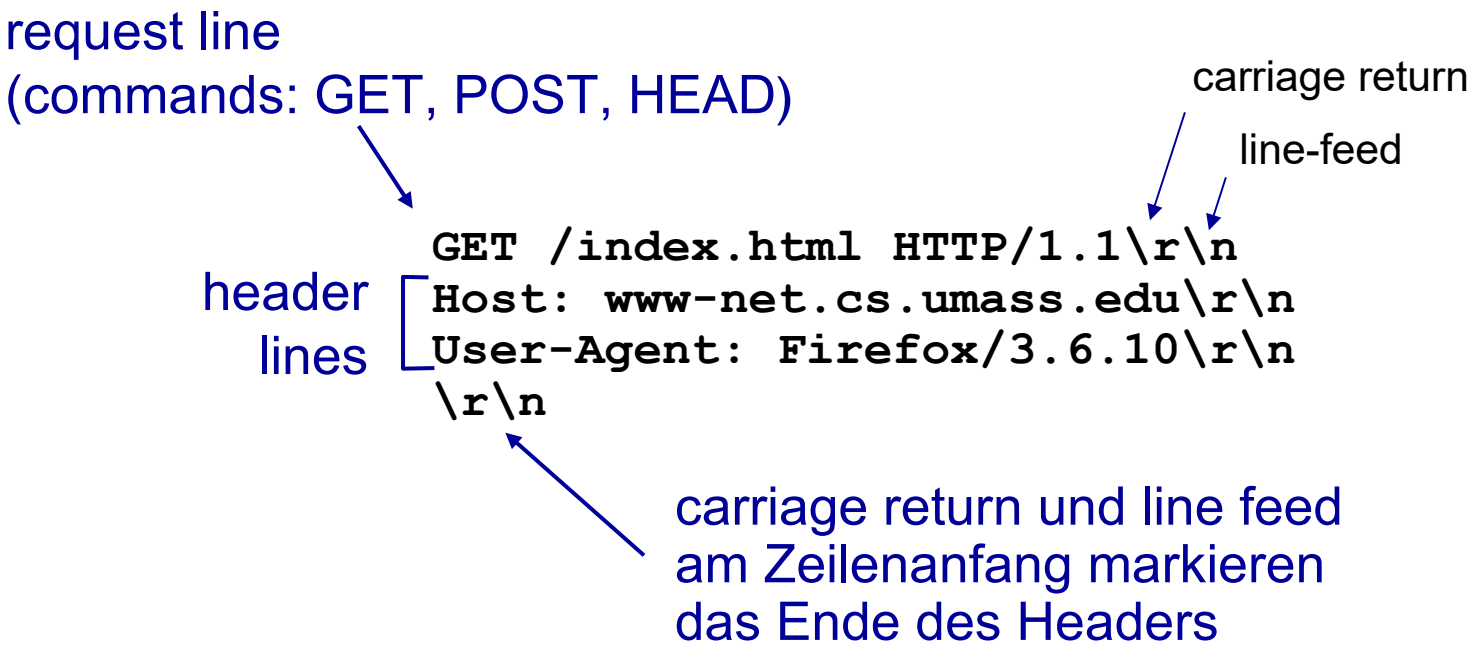
hypertext transfer protocol

- Client-Server Modell
  - *Client:* Browser, der Objekte anfordert, empfängt und darstellt
  - *Server:* Webserver, der Objekte auf Anfrage sendet
- Zwei Typen von HTTP Nachrichten:
  - *Request*
  - *Response*



HTTP Request Nachricht

- ASCII (human-readable format)



HTTP Response Nachricht

status line

header lines

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
\r\n
data data data data data ...
```

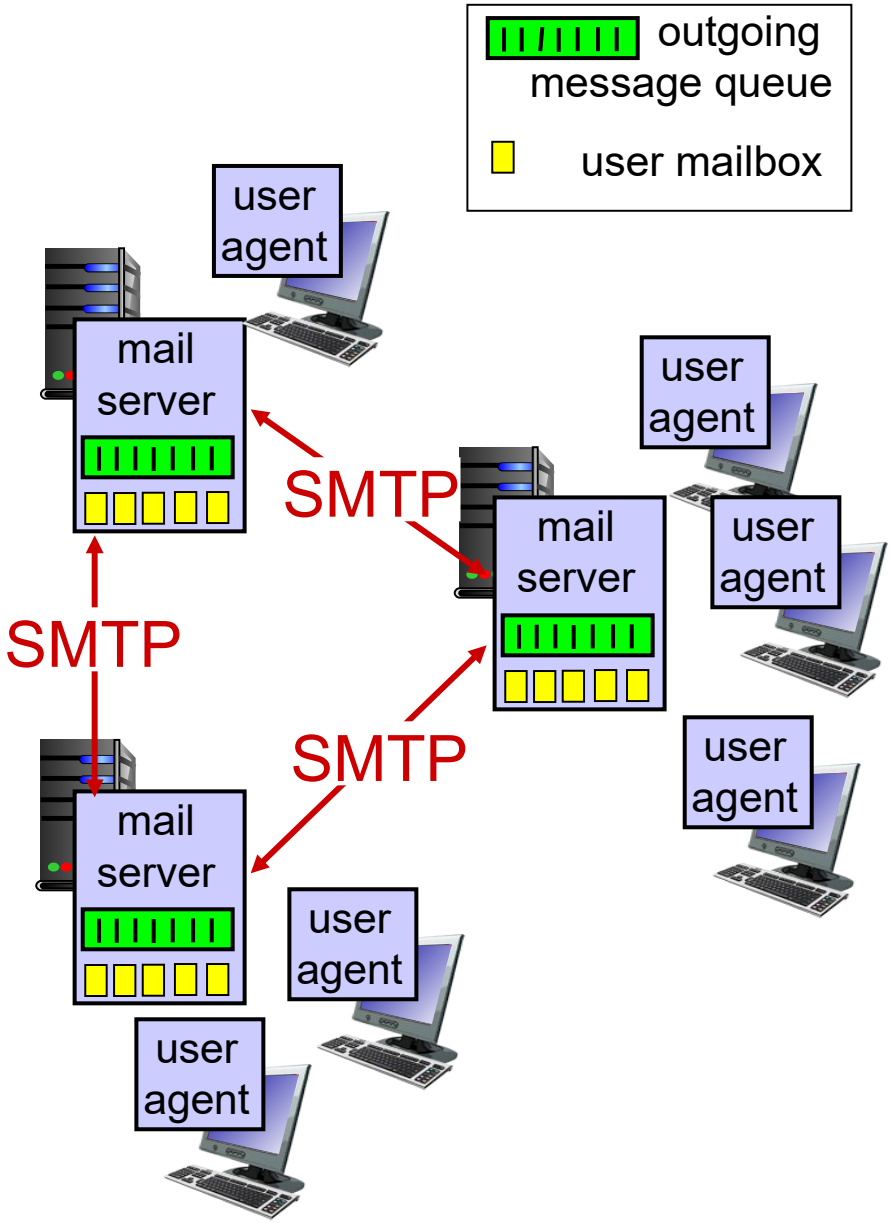
data, z.B. die angeforderte HTML-Datei


B

Email

Drei Komponenten:

- User Agents
- Mailserver
- SMTP: Simple Mail Transfer Protocol



- *Mailbox* für eingehende Nachrichten
- *Message Queue* für ausgehende Nachrichten
- *SMTP* für den Nachrichtenaustausch mit und zwischen Mailservern

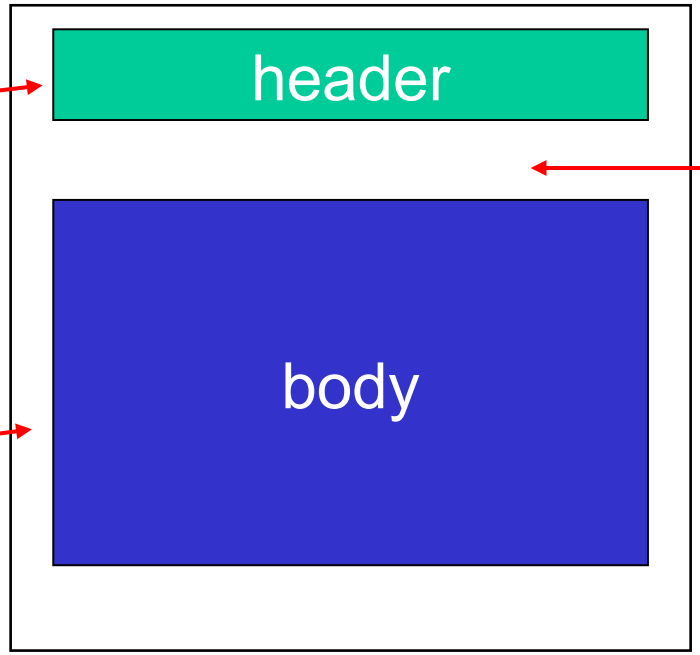
Beispielverlauf: SMTP

```
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
```

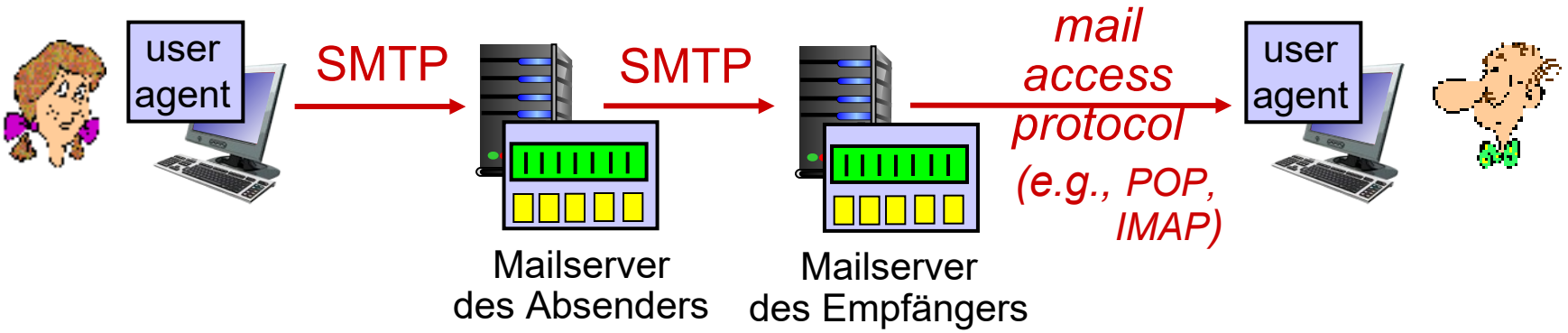
Format der Mail-Nachricht

RFC 822: Standard für das Nachrichtenformat

- header
  - To:
  - From:
  - Subject:
  - …
- body
  - die "eigentliche Nachricht", Mailtext
  - ASCII !!!



Mailzugriffsprotokolle



- SMTP: liefern und speichern im Server des Empfängers
- Mailzugriffsprotokoll (mail access protocol): Abholen der Nachrichten aus dem Postfach (mailbox) des Servers
  - POP: Post Office Protocol [RFC 1939]
  - IMAP: Internet Mail Access Protocol [RFC 1730]
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.