

openHPI-Kurs: Wie funktioniert das Internet?
Einführung Internetanwendungen

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Internetanwendungen (1/4)

... eine bunte Welt ohne die heute nicht mehr viel geht

■ WWW

- Suchmaschinen
- Soziale Netzwerke
- Shopping, Banking
- Karten

■ Email

■ Mediastreaming

■ Clouddienste

■ VoIP

■ ...



OpenStreetMap®



Screenshot Microsoft Edge mit geöffneter openHPI-Webseite



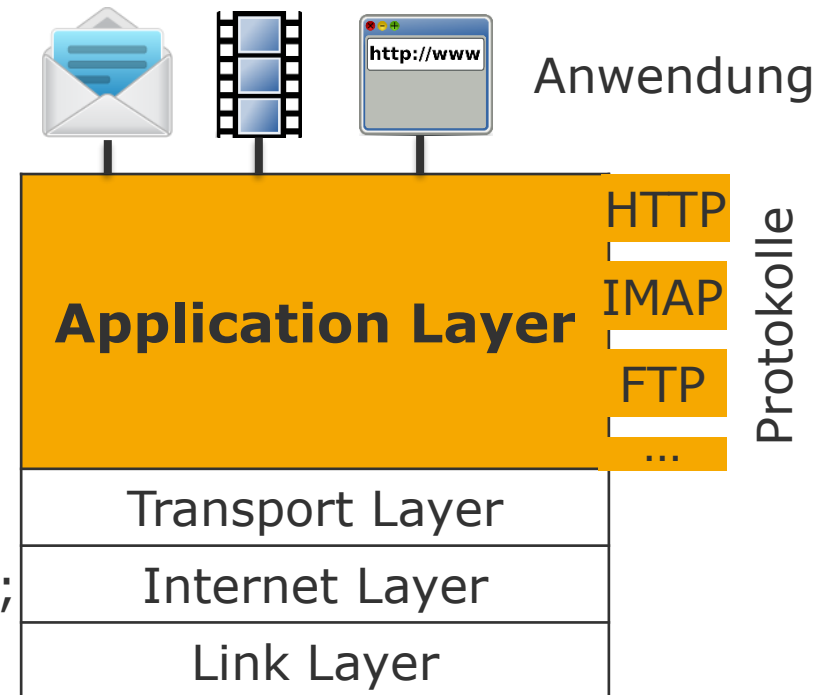
Internetanwendungen (2/4)

Internetanwendungen ziehen Nutzer an

- Popularität des Internets ist das Resultat der Vielfalt und Attraktivität der Internetanwendungen
 - Anfangs hatten das Internet nur wenige, technikaffine Menschen genutzt / nutzen können, etwa in Forschungszentren
 - Heutzutage nutzt schon die Hälfte aller Menschen das Internet
- Internetnutzung (eigentlich Nutzung der Internetanwendungen) ist in unserer Gesellschaft zum selbstverständlichen Teil des privaten und beruflichen Lebens geworden
- Im Gegensatz zu frühen Internetzeiten („online gehen“) erfolgt die Internetnutzung heute meist vollkommen unbewusst und (leider auch) unreflektiert

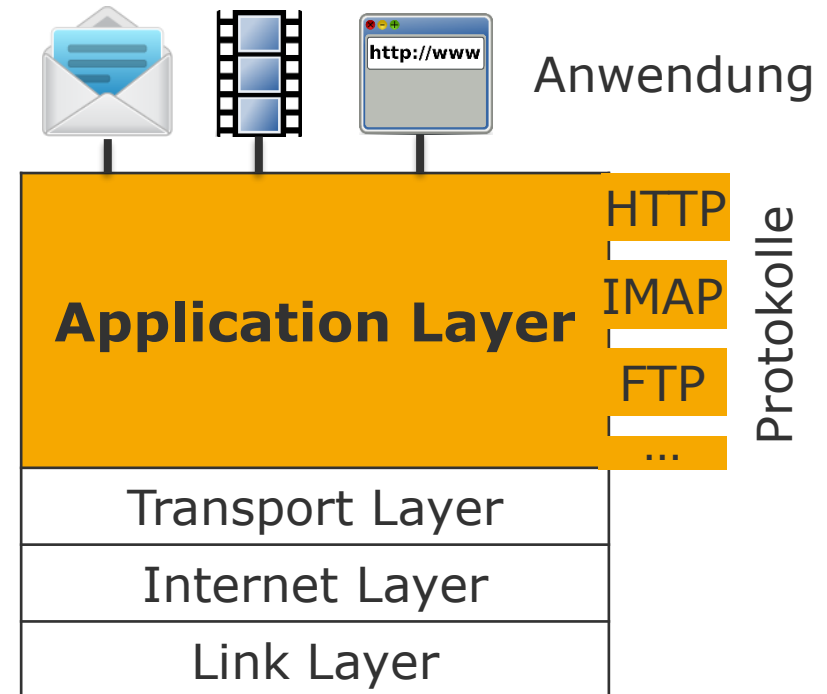
Internetanwendungen (3/4)

- Damit **Internetanwendungen** das Internet für ihre Zwecke nutzen können, braucht es Kommunikationsprotokolle in der **Anwendungsschicht** des TCP/IP-Protokollstapels
- Die **Anwendung** selbst, z.B. Web-Browser, Email-Client, ... steht außerhalb des TCP/IP-Stapels; sie benutzt die von den Anwendungsprotokollen bereitgestellten Schnittstellen



Beispiele

- WWW – HTTP, ...
- Email – IMAP, POP3, SMTP
- Filetransfer – FTP
- Streaming – RTP, ...
- Instant Messaging – XMPP, ...
- Voice over IP – SIP, ...
- ...





openHPI-Kurs: Wie funktioniert das Internet?

Architektur der Internetanwendungen

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Internetanwendungen (1/5)

Internetanwendungen brauchen Anschluss, um Internet nutzen zu können:



4	Application Layer
3	Transport Layer
2	Internet Layer
1	Link Layer

Internetanwendungen (2/5)

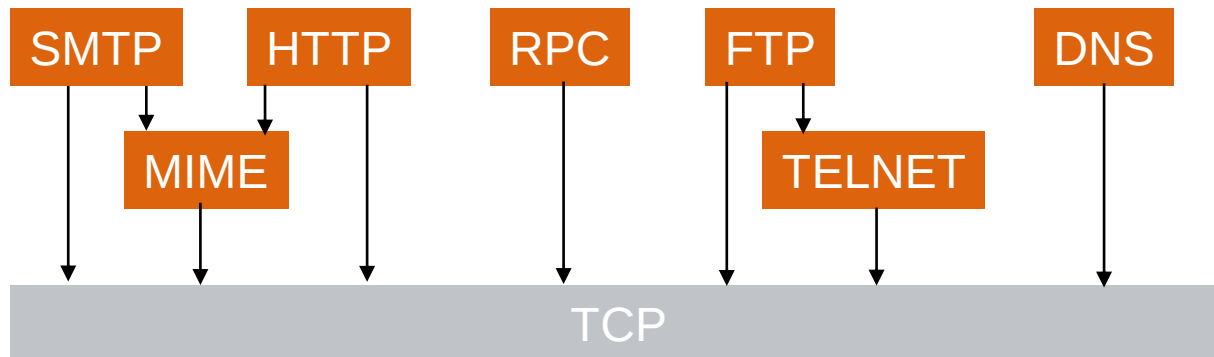
Erinnerung: TCP (Transmission Control Protocol):

- verbindungsorientiert und zuverlässig

Internetdienste, die auf dem Transportprotokoll TCP aufsetzen

- **SMTP:** Simple Mail Transfer Protocol
- **HTTP:** Hypertext Transfer Protocol
- **RPC:** Remote Procedure Call
- **MIME:** Multipurpose Internet Mail Extension
- **FTP:** File Transfer Protocol
- **TELNET:** Teletype Network
- **DNS:** Domain Name Service

Internetdienste, die auf dem Transportprotokoll TCP aufsetzen:



Internetanwendungen (4/5)

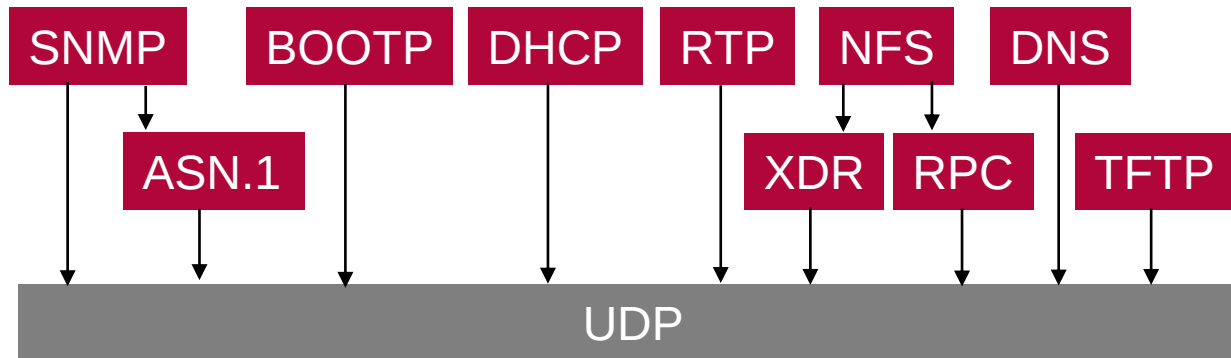
Erinnerung: UDP (User Datagram Protocol):

- verbindungslos und unzuverlässig

Internetdienste, die auf dem Transportprotokoll UDP aufsetzen

- **TFTP:** Trivial File Transfer Protocol
- **DNS:** Domain Name Service
- **SNMP:** Simple Network Management Protocol
- **BOOTP:** Boot Protocol
- **DHCP:** Dynamic Host Configuration Protocol
- **RTP:** Realtime Transfer Protocol
- **RPC:** Remote Procedure Call
- **NFS:** Network File System

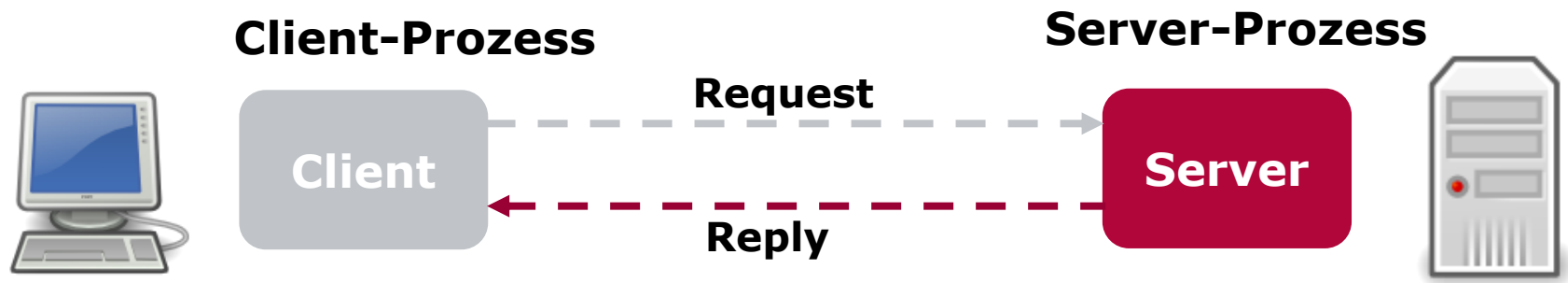
Internetdienste, die auf dem Transportprotokoll UDP aufsetzen:



Client-Server Prinzip (1/2)

Internetanwendungen sind die Hauptakteure, die über das Internet miteinander kommunizieren

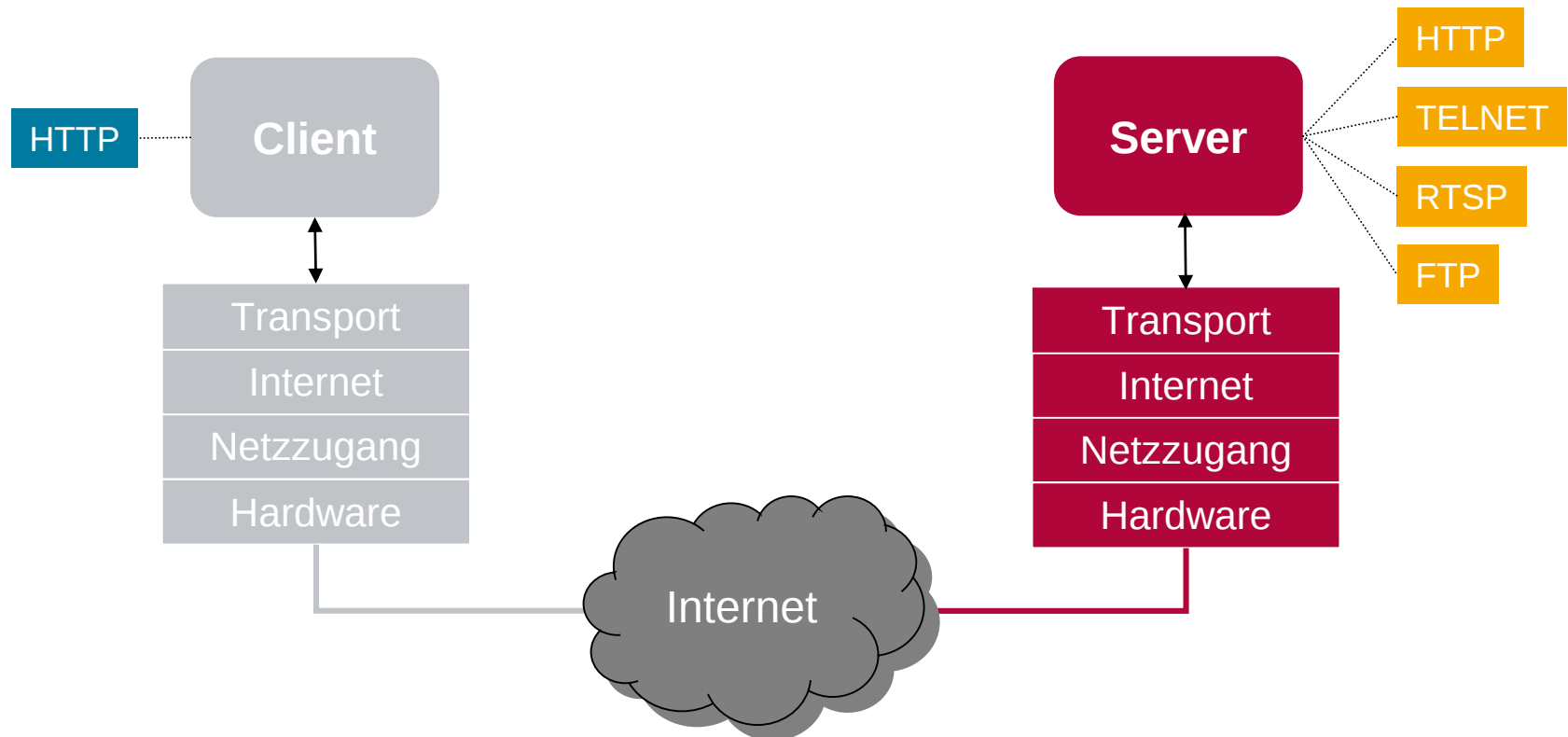
- Internetanwendungen basieren auf dem Client-/Server-Prinzip



- | | |
|--|--|
| <ul style="list-style-type: none"> ■ aktive Komponente ■ stellt Anfrage an Anwendung, die auf einem anderen Rechner läuft – „Request“ | <ul style="list-style-type: none"> ■ passive Komponente ■ antwortet auf Anfrage mit der gewünschten Information – „Reply“ |
|--|--|

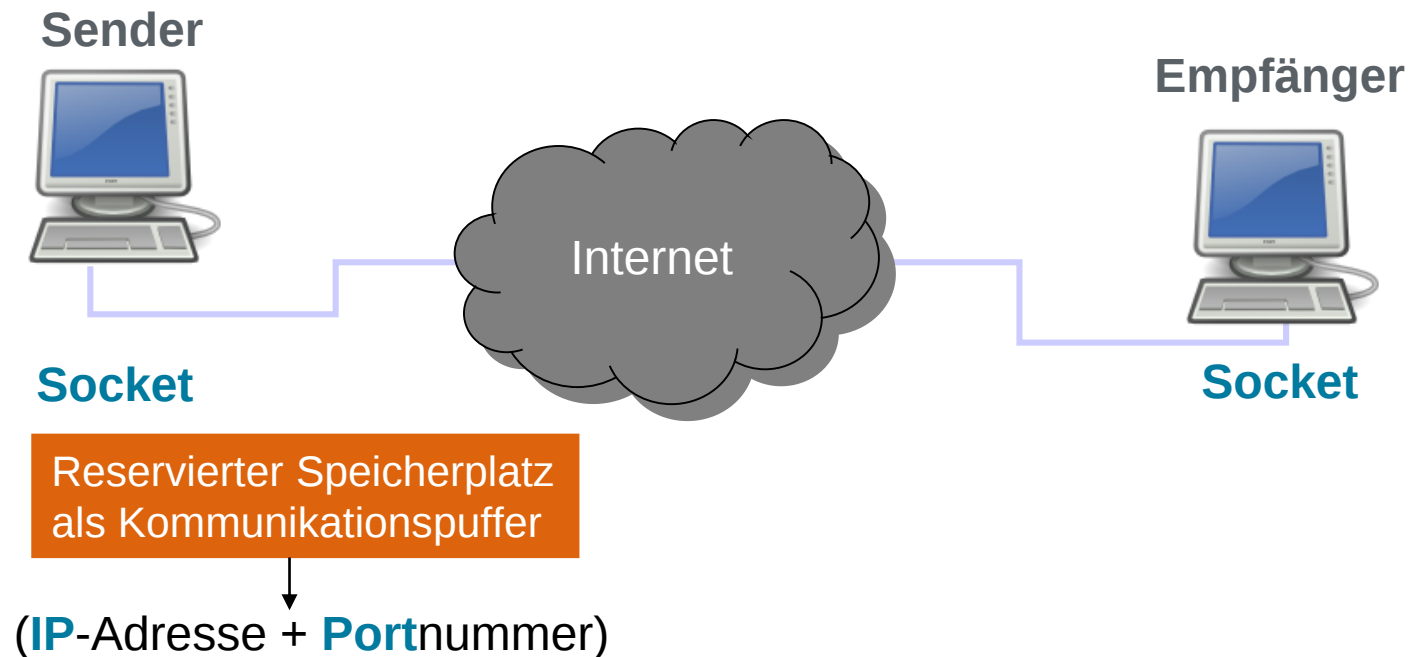
Client-Server Prinzip (2/2)

Ein Rechner (Server-Hardware) kann zugleich verschiedene Dienste (Services, Software) anbieten



Socket-Schnittstelle (1/2)

Um Verbindung zwischen zwei Endsystemen einzurichten und aufrechtzuerhalten, müssen auf Sender- und Empfängerseite eindeutig identifizierbare Endpunkte der Kommunikation – **Sockets** – eingerichtet werden



Socket-Schnittstelle (2/2)

Sockets

- Definieren eine bidirektionale Schnittstelle zur Interprozesskommunikation
- Vergleichbar mit UNIX-Sockets zur lokalen Interprozesskommunikation
- 1983 eingeführt mit BSD-Unix
- Tupel aus
 - Quell-/Zieladresse
 - Quell-/Zielport
 - Netzwerkprotokoll
- Programm greift auf Socket ähnlich wie auf eine Datei zu
 - Operationen read / write / flush /...



openHPI-Kurs: Wie funktioniert das Internet?

DNS – Domain Name Service

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

DNS – Domain Name Service (1/2)

Zu lösendes **Problem**:

Kein Mensch kann sich leicht IP-Adressen merken

- 136.199.35.24
- 23.234.13.1
- 213.3.21.9
- 127.44.32.96
- 192.56.93.3

Aussagekräftige Namen lassen sich dagegen besser behalten

- hpi.de
- google.com
- stanford.edu
- europa.eu
- www.w3c.org

DNS – Domain Name Service (2/2)

Idee:

- Einführung eines hierarchisch strukturierten Namensraum – „Domänennamen“
 - Beispiel: pc55.open.hpi.de
- Bereitstellung eines DNS-Namensdienstes im Internet, der Domain-Namen in IP-Adressen übersetzt
- DNS ist eine Client-Server-basierte, verteilte Anwendung

Geschichte:

- DNS wurde im 1983 von Paul Mockapetris vorgeschlagen
- RFC 882, jetzt RFC 1034 / 1035



Paul Mockapetris

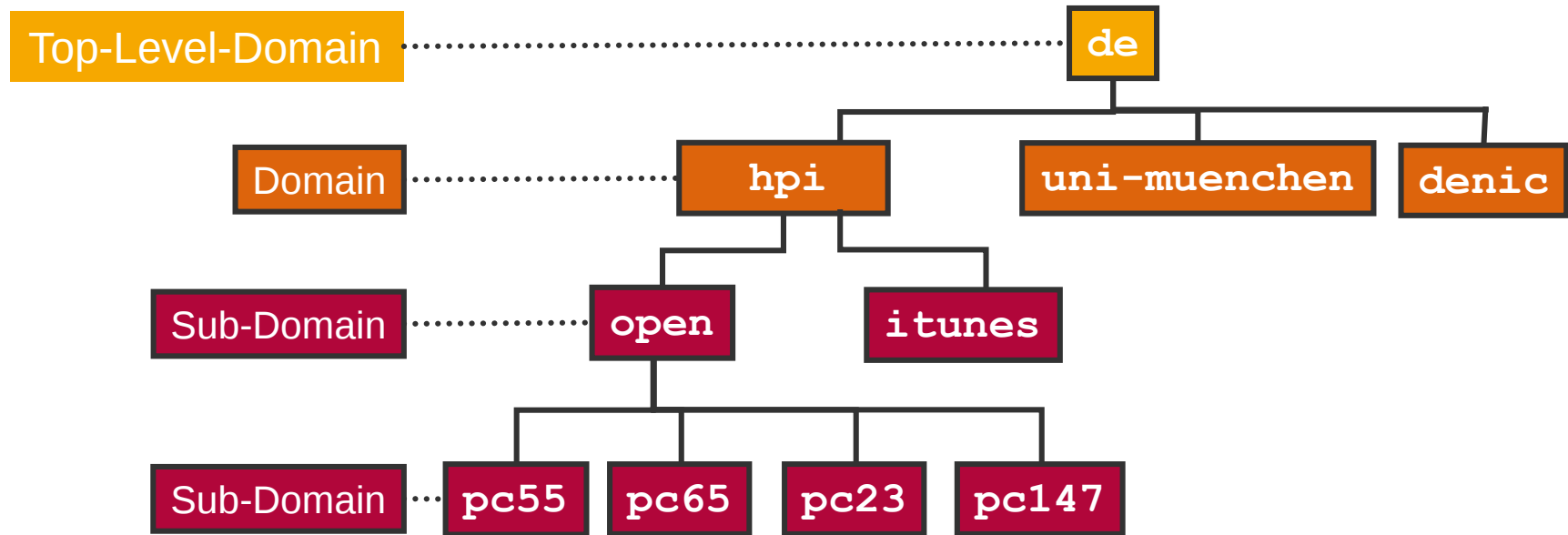
DNS-Namensraum



DNS-Namensraum lässt sich als Baumstruktur darstellen

DNS-Namensraum (2/5)

Ausschnitt aus dem DNS Namensraum



DNS-Namensraum (3/5)

Top-Level-Domains

- Werden zentral von ICANN vergeben

Beispiele:

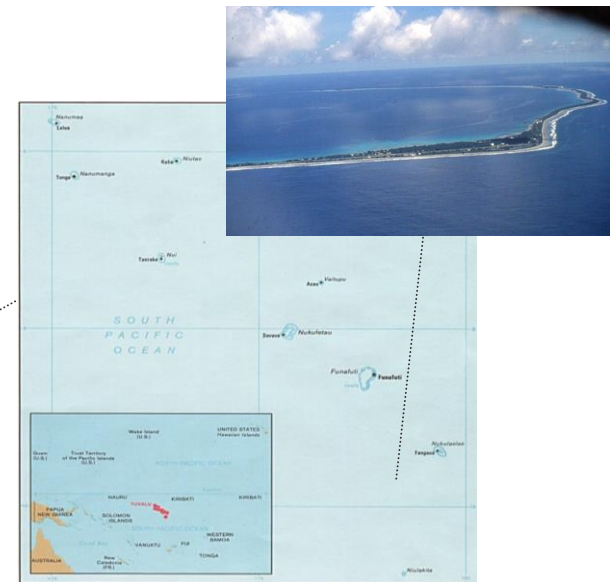
- Generic Domains: Institutionen und Organisationen
 - com kommerzielle Unternehmen
 - edu (US)-Ausbildungseinrichtungen
 - gov (US)-Regierung
 - mil (US)-Militär
 - org gemeinnützige Organisationen
 - int Internationale Organisationen
 - net Netzwerk Provider
 - ...

DNS-Namensraum (4/5)

Weitere Top-Level-Domains:

■ Country Domains: Länderkennung (nach ISO3166)

- ☐ de Deutschland
- ☐ fr Frankreich
- ☐ uk England
- ☐ ru Russland
- ☐ ch Schweiz
- ☐ tv Tuvalu
- ☐ eu Europa
- ☐ ...



Aktuell Entwicklung:

- In letzter Zeit wurden sehr viele weitere Top-Level-Domains eingeführt, wie z.B. .books, .singles, .berlin,

DNS-Namensraum (5/5)

- Verwaltung einer Top-Level-Domain und der ihr zugeordneten Domains erfolgt zentral durch ein zuständiges Unternehmen
- Für .de Top-Level-Domain ist → DE-NIC (Deutsches Network Information Center) in Frankfurt zuständig



DNS bietet Übersetzung von DNS-Namen in IP-Adresse

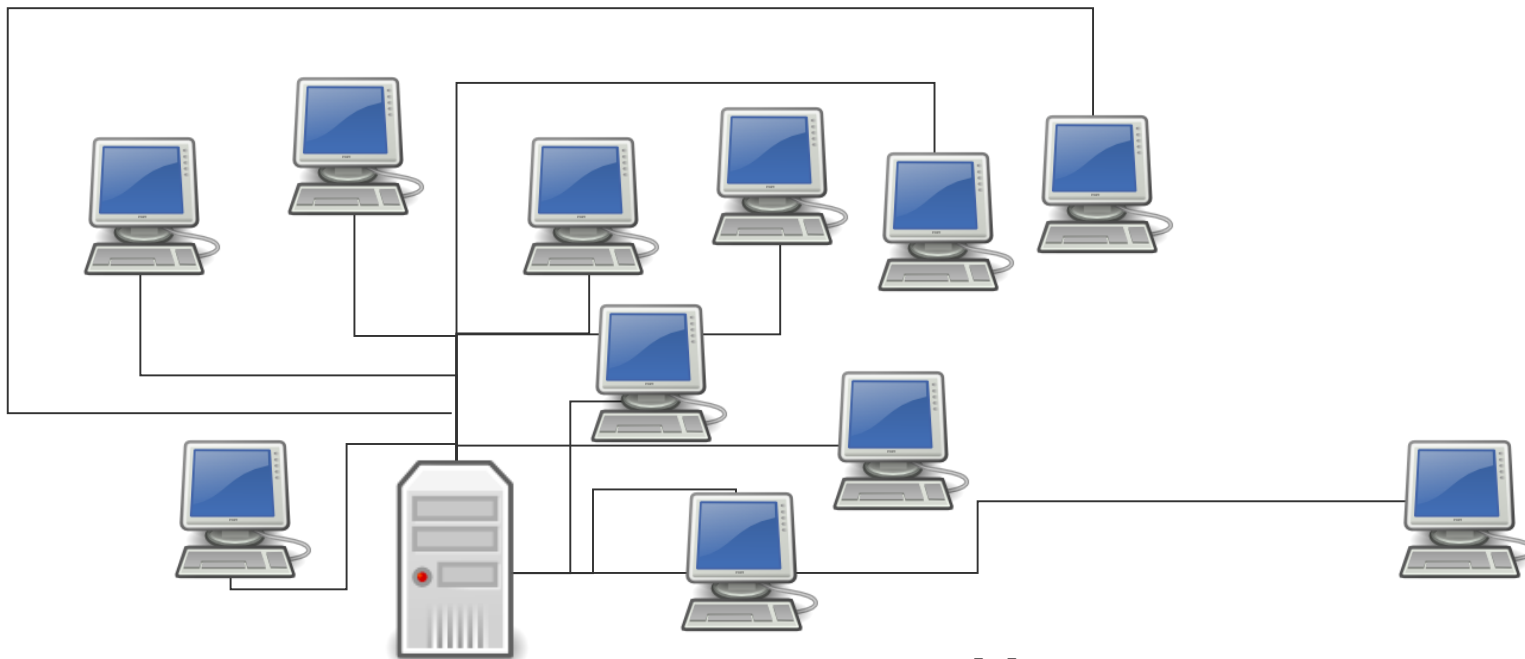
DNS – Funktionsweise (1/5)

Namensauflösung via DNS:

- 1) Internetnutzer gibt Domainnamen einer Ressource ein
- 2) Betriebssystem sucht in Hosts-Datei nach passendem Eintrag. Annahme: findet keinen
- 3) User-Client (Resolver) sendet Anfrage an DNS-Server
- 4) DNS-System verarbeitet Anfrage, findet die zum Domainnamen passende IP-Adresse und sendet diese zurück an Resolver
 - Domain-Informationen wird vom DNS Nameserver in Form von Ressource Records gespeichert
- 5) Client kann jetzt Ressource im Internet via IP-Adresse abrufen und Internetnutzer kann jetzt auf Ressource zugreifen...

DNS – Funktionsweise (2/5)

Macht es Sinn, einen einzigen, zentralen Server zu betreiben, der alle IP-Adressen und die ihnen zugeordneten DNS-Namen verwaltet?



zentraler
DNS-Server



Problem:

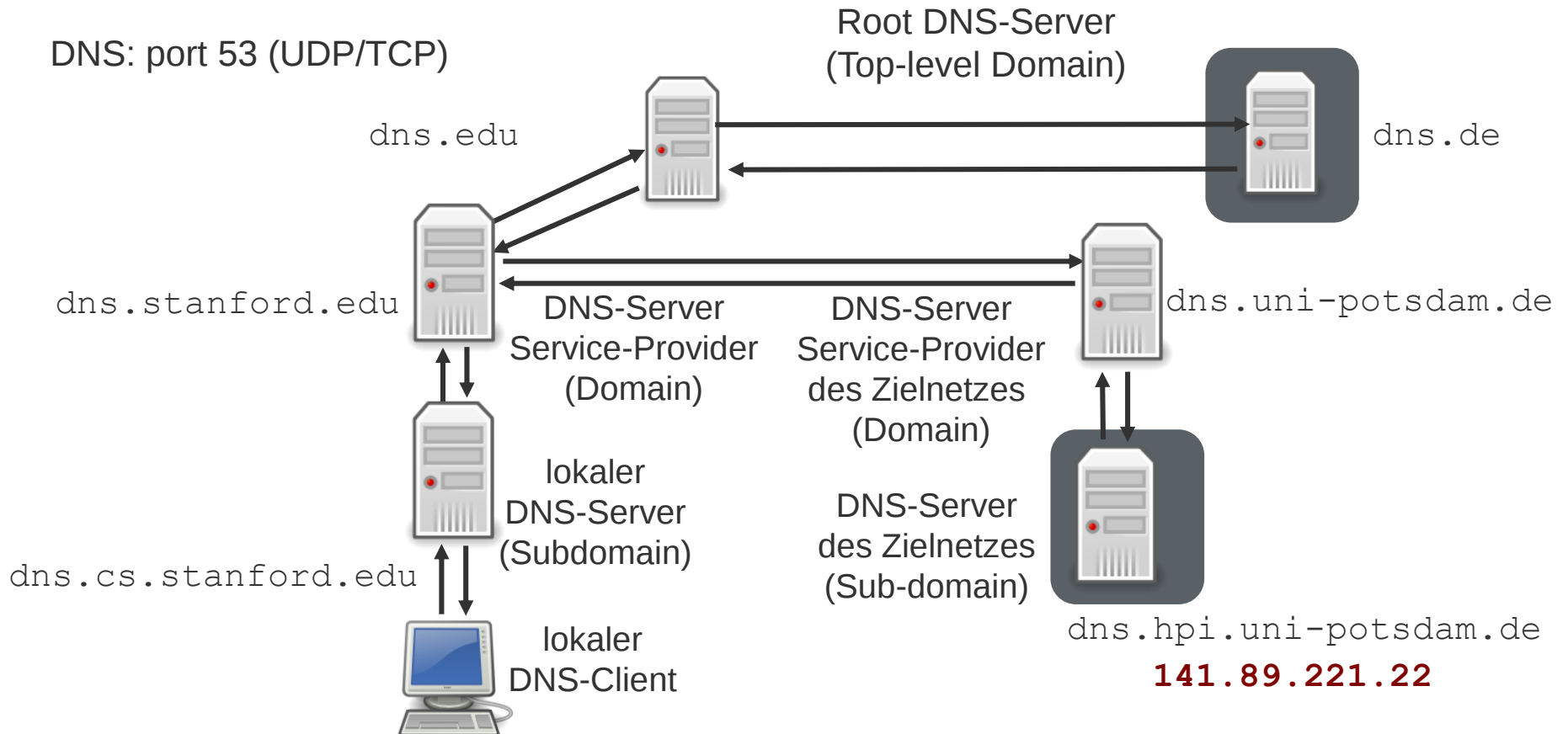
- etwa 1 Milliarde Internet-Hosts
- permanente Überlastung!

DNS – Funktionsweise (3/5)

- Mit nur einem zentralen Nameserver zu arbeiten, wäre sehr problematisch!
- DNS-Namensraum wird deshalb hierarchisch in nicht überlappende Zonen aufgeteilt
- Jede Zone wird von eigenem DNS-Server verwaltet
- DNS-Server kennt jeweils
 - alle Server/Hosts auf der direkt unter ihm liegenden Hierarchiestufe
 - den Server, der direkt über ihm in der Hierarchie steht

DNS – Funktionsweise (4/5)

DNS: port 53 (UDP/TCP)



Suche in Stanford nach: **www.hpi.uni-potsdam.de**

DNS – Funktionsweise (5/5)

Löst verteilter DNS-Service ursprüngliches Überlastungsproblem?

- Preis für die Verteilung des DNS-Services auf verschiedene Domain Name Server:
 - Namensauflösung kann ganze Reihe kaskadierende DNS-Abfragen erfordern ...

Aber:

- Wenn verteilter DNS-Service Vielzahl von Namensauflösungen von DNS-Abfragen erfordern würde, wäre nichts gewonnen, denn Server in den oberen Hierarchie-Ebenen blieben ständig überlastet ...

DNS – Caching

Idee:

- Die zuletzt angefragten IP-Adressen werden zusammen mit dem Domain Name in einem Zwischenspeicher – **DNS-Cache** – abgelegt
- Bei erneuter Anfrage erfolgt zuerst Lookup im DNS-Cache
 - Aufgrund des **Lokalitätsprinzips** sind meist keine DNS-Client-Server Interaktionen zum nächsthöheren Server mehr notwendig
- DNS-Caching ist sehr wirkungsvoll, da sich DNS-Einträge nur selten ändern
- DNS-Cache-Einträge bleiben für vorgegebene Zeit – Time-to-Live (TTL) - aktiv, bevor sie wieder aus dem Cache gelöscht werden

DNS-Datensätze

Domain Name	TTL	Klasse	Typ	Wert
-------------	-----	--------	-----	------

- TTL: Lebensdauer in Sekunden
 - statisch (z.B. 86.400 = Tag)
 - flüchtig (z.B. 60)
- Klasse: IN = Internet
- Typ

SOA	Start of Authority	Parameter für betreffende Domain
A	Host IP	IP-Adresse
MX	Mail Exchange	Domäne für E-Mail
NS	Name Server	Nameserver der Domäne
CNAME	Kanonischer Name	Domänenname
PTR	Pointer	Eintrag für Reverse DNS Lookup
HINFO	Host Info	CPU / OS
TXT	Text	Nicht interpretierter Text

DNS-Lookup



openHPI-Kurs: Wie funktioniert das Internet?

Email

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Email – Elektronische Post (1/5)

Versenden elektronischer Post ist eine der ältesten Internetanwendungen

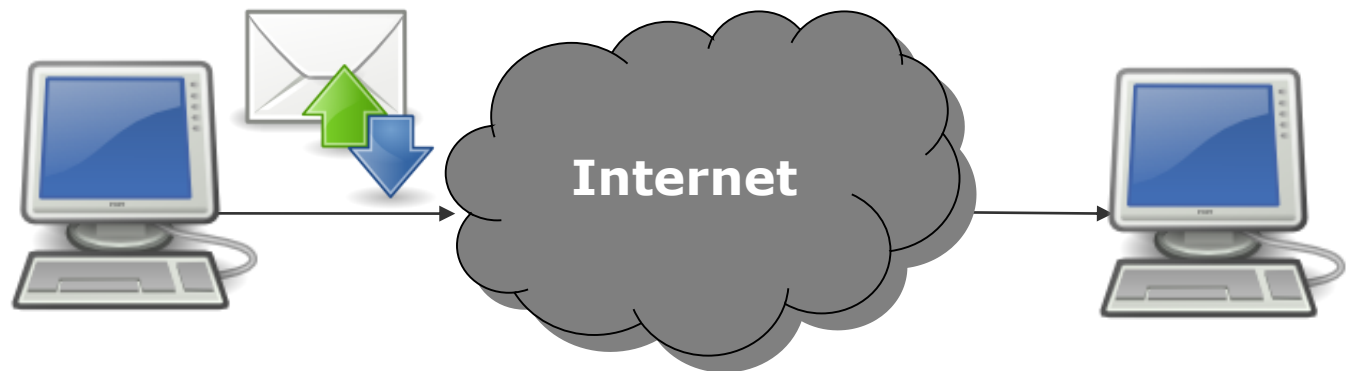
- War bereits möglich im ARPANET (1971)
- Ursprüngliche Standards von 1982 (RFC 821 und 822) wurden 2008 durch RFCs 5321 und 5322 ersetzt



AileenWang@Flickr-CC

Ray Tomlinson

(* 2. Oktober 1941, † 5. März 2016)



Email – Elektronische Post (2/5)

Email-Dienst zählt zu den **Message Handling Systems (MHS)**

MHS besteht aus zwei Komponenten

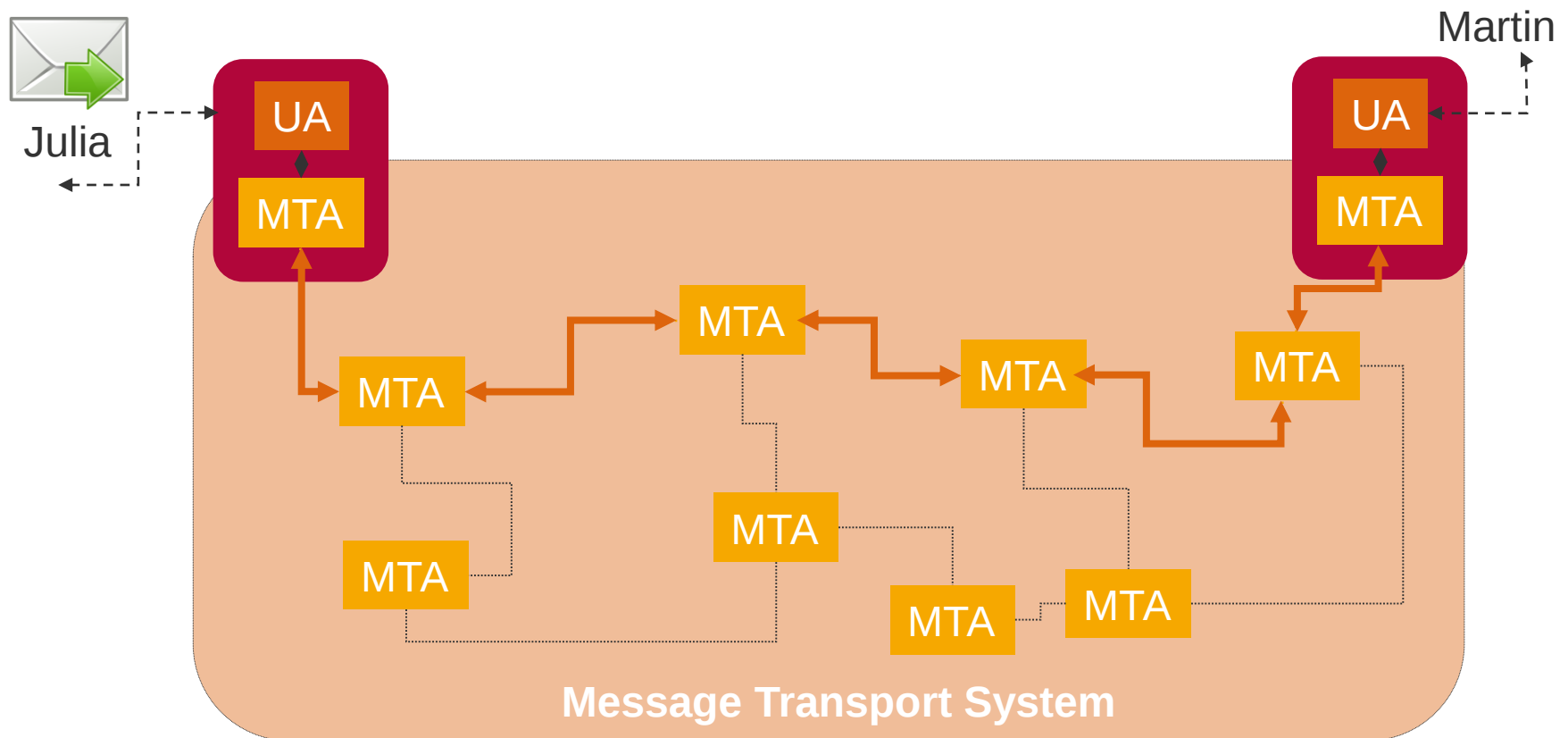
User Agents (UA) – RFCs 5322, 6854

- Lokales Anwendungsprogramm, das dem Anwender eine Benutzerschnittstelle zum Email-Dienst anbietet
 - Anwender kann mit UA Nachricht erzeugen und editieren
 - Nachrichten können gelesen, empfangen, gesendet werden
 - Beispiele: Outlook, Thunderbird, Gmail etc.

Message Transfer Agent (MTA) – RFC 5321

- Verantwortlich für Transport der Email-Nachrichten zwischen Sender und Empfänger
- Oft müssen dabei mehrere MTA zusammenarbeiten

Message Handling Systeme



Email – Elektronische Post (4/5)

Analogie zur herkömmlichen Post



Briefumschlag

Dingsbums GmbH
Hauptstr. 23
32105 Bad Salzuflen
10. Oktober 2016
Betrifft: MAHNUNG!

Lieber Herr Nickel,
leider sehen wir uns
gezwungen Ihnen
nachfolgende
Mahngebühr...

Header

Nachricht

Name: Dr. K. Nickel
Straße: Am Hasenbau 7
Ort: Waldstadt
PLZ: 12345
Priorität: Dringend

From: Dingsbums GmbH
Adresse: Hauptstr. 23
Ort: Bad Salzuflen
PLZ: 32105
Date: 10.10.2016
Subject: MAHNUNG!

Lieber Herr Nickel,
leider sehen wir uns
gezwungen Ihnen
nachfolgende Mahngebühr...

Email – Elektronische Post (5/5)

Basisfunktionen:

- Erzeugen einer Nachricht (**Composition**)
 - Texteditor
 - Sorgt für korrektes Erzeugen/Einfügen der Headerinformationen
- Zustellung der Nachricht (**Transfer**)
 - Verbindungsaufbau / Transfer / Verbindungsabbau
- Benachrichtigung über Verlauf der Zustellung (**Reporting**)
 - Positive/negative Rückmeldung / Fehlerursachen
- Darstellung der zugestellten Nachricht (**Displaying**)
 - Eventuelle Umformatierungen notwendig
- Ablegen der zugestellten Nachricht (**Disposition**)
 - Löschen oder Aufheben...

Email – Adresse

- Sender und Empfänger einer Nachricht werden über spezielle Email-Adressen (RFC 5321) identifiziert
- Email-Adresse besteht aus



Email - Nachrichtenformat

Beispiel:

```
Received: from 3MXMA1R.hpi.uni-potsdam.de ([141.89.224.242])
        by 3mxstulr.hpi.uni-potsdam.de
        with Microsoft SMTPSVC(6.0.3790.1830); Wed, 10 Oct 2016 08:28:20 +0200
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
Content-class: urn:content-classes:message
Return-Path: <Meinel@hpi.uni-potsdam.de>
X-Mozilla-Status: 0001
X-MimeOLE: Produced By Microsoft Exchange V6.5
X-OriginalArrivalTime: 10 Oct 2016 08:28:20.0212 (UTC) FILETIME=[2FB28B40:01C8C7D9]
Subject: WWW-Vorlesung in der nächsten Woche
Date: Wed, 10 Oct 2016 08:28:19 +0200
Message-ID: <5AC0F51A926CD442BC08D55DA7DAB1D40B5DEE37@3mxma1r.hpi.uni-potsdam.de>
From: "Prof. Dr. Christoph Meinel" <Meinel@hpi.uni-potsdam.de>
To: "Matthias Bauer" <matthias.bauer@hpi.uni-potsdam.de>

Lieber Herr Bauer,
...
```



openHPI-Kurs: Wie funktioniert das Internet?

MIME-Standard

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Email-Erweiterung MIME (1/7)

Im Email-Internetstandard RFC 5322 ist nur die Verwendung von 7-Bit (US) ASCII Zeichen erlaubt, also

- keine Umlaute
- keine Sonderzeichen
- nur Text
- keine Grafik / Audio / sonstigen Binärdaten

Will man trotzdem Sonderzeichen, Grafiken und sonstige Binärdaten per Email versenden, müssen die vorher in 7-Bit ASCII Zeichen umkodiert werden

→ **MIME-Standard** (1993) - Multipurpose Internet Mail Extension – legt Regeln fest, wie nicht 7-Bit ASCII Datentypen behandelt werden sollen

- Interpretation von MIME ist dann Aufgabe der User Agenten

Email-Erweiterung MIME (2/7)

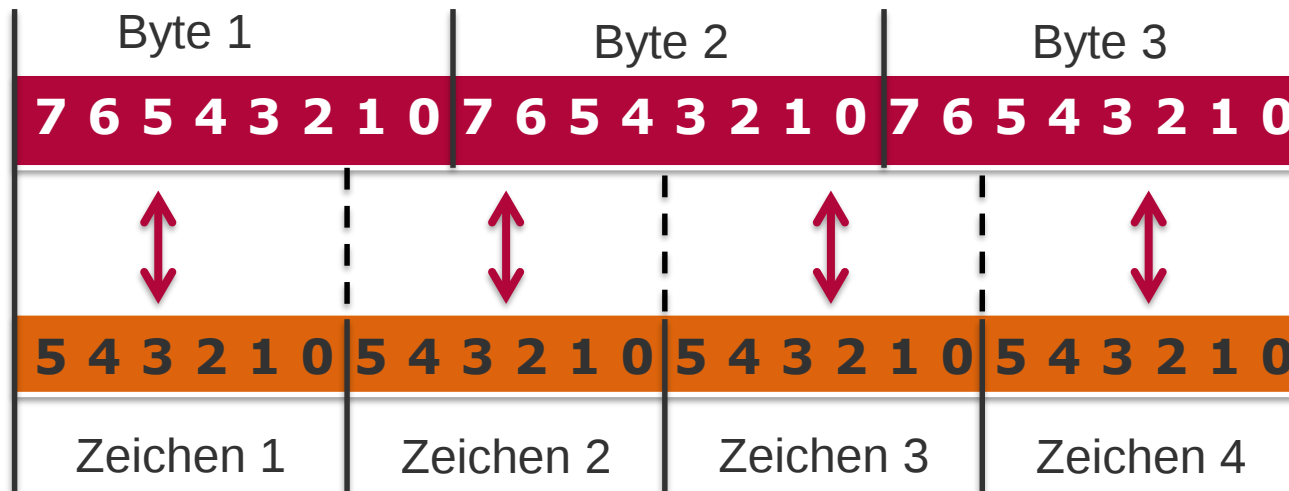
MIME bietet folgende Kodierungsvarianten:

- 7 Bit ASCII nach RFC 5322, max. 128 Zeichen
- 8 Bit ASCII inkl. nationale ASCII-Ergänzungen, max. 256 Zeichen
- Base64 Encoding
 - zerlegt jeweils 24-Bit Folgen der Originalnachricht in 6-Bit Sequenzen
 - 6-Bit Sequenzen werden mit 7 Bit ASCII (Werte 0-64) kodiert
- Quoted Printable Encoding
 - nur Sonderzeichen werden speziell kodiert, sonst 7 Bit ASCII
- User Defined Encoding
 - Sender und Empfänger übernehmen Dekodierung in eigener Regie

Email-Erweiterung MIME (3/7)

Beispiel: Base64 Encoding

- Verwendet Zeichen A-Z, a-z, 0-9, + und /
- Ende wird mit „=“ gekennzeichnet
- Erlaubt kodierungsunabhängigen Datenaustausch



Email-Erweiterung MIME (4/7)

MIME erlaubt folgende Typen von Binärdateien:

Typ	Subtyp
Text	Plain, RTF, HTML, etc.
Image	GIF, JPEG, TIFF, etc.
Audio	Basic, x-pn-realaudio, etc.
Video	MPEG, quicktime, etc.
Application	Octet-Stream, Postscript, etc.
Message	RFC5322, Partial, External
Multipart	Mixed, Alternative, Parallel, Digest

Email-Erweiterung MIME (5/7)

Beispiel für eine im MIME-Standard kodierte Grafikdatei:



Grafikdatei (binär):
run.jpg

MIME-
Kodierung



Content-Type: image/jpeg; name="run.jpg"
Content-Transfer-Encoding: Base64
Content-Disposition: attachment; filename="run.jpg"

```
/9j/4AAQSkZJRgABAgEAtAC0AAD/4Q9bRXhpZgAASUkqAAgAAAAKAA8BAgAG
AAAAhgAAABABAgAOAAAAjAAAAIBAwABAAAAQAAABoBBQABAAAAmgAAABsB
BQABAAAAogAAACgBAwABAAAAAgAAADEBAGAUAAAAqgAAADIBAGAUAAAAvgAA
ABMCAwABAAAAQAAAGmHBAABAAAA1AAAAJwDAABDYW5vbGBDYW5vbiBFT1Mg
RDMwALQAAAABAAAAtAAAAEAAABZG9iZSBQaG90b3Nob3AgNy4wADlwMDQ6
MDY6MDEgMTA6Mjg6MzYAAAAaAJqCBQABAAAAEgIAAJ2CBQABAAAAAGgIAACeI
AwABAAAAkAEAAACQBwAEAAAMDIxMAOQAgAUAAAAIglAAASQAgAUAAAAngIA
AAGRbWAEAAAAAQIDAARbQABAAAAAgIAAAGSCgABAAAAUglAAAKSBQABAAAA
WglAAASSCgABAAAAYglAAAWSBQABAAAAAgIAAAaSBQABAAAAAgIAAAeSAwAB
.....
.....
.....
```

ASCII-Datei (MIME Base64):
run.mime

Email-Erweiterung MIME (6/7)

Beispiel für eine Email, die MIME Kodierung enthält:

```
From: matthias.bauer@hpi.de
To: christoph.meinel@hpi.de
MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=Next Part
--NextPart

Content-Type: TEXT/PLAIN
Hallo Herr Professor Meinel,
anbei ein Screenshot der Hausaufgaben-Statistik aller
Studenten Ihres openHPI-Kurses.
Bis morgen!
Matthias Bauer

--Next Part
Content-Type: IMAGE/JPEG
Content-Transfer-Encoding: base64
...
Binäre Daten der jpeg-Datei in base64-Kodierung
```

Email-Erweiterung MIME (7/7)

MIME-Version:

- teilt dem User Agent Version der Kodierung mit

Content-Description:

- freie Beschreibung des Inhalts

Content-ID:

- eindeutige Identifikation des Inhalts

Content-Type:

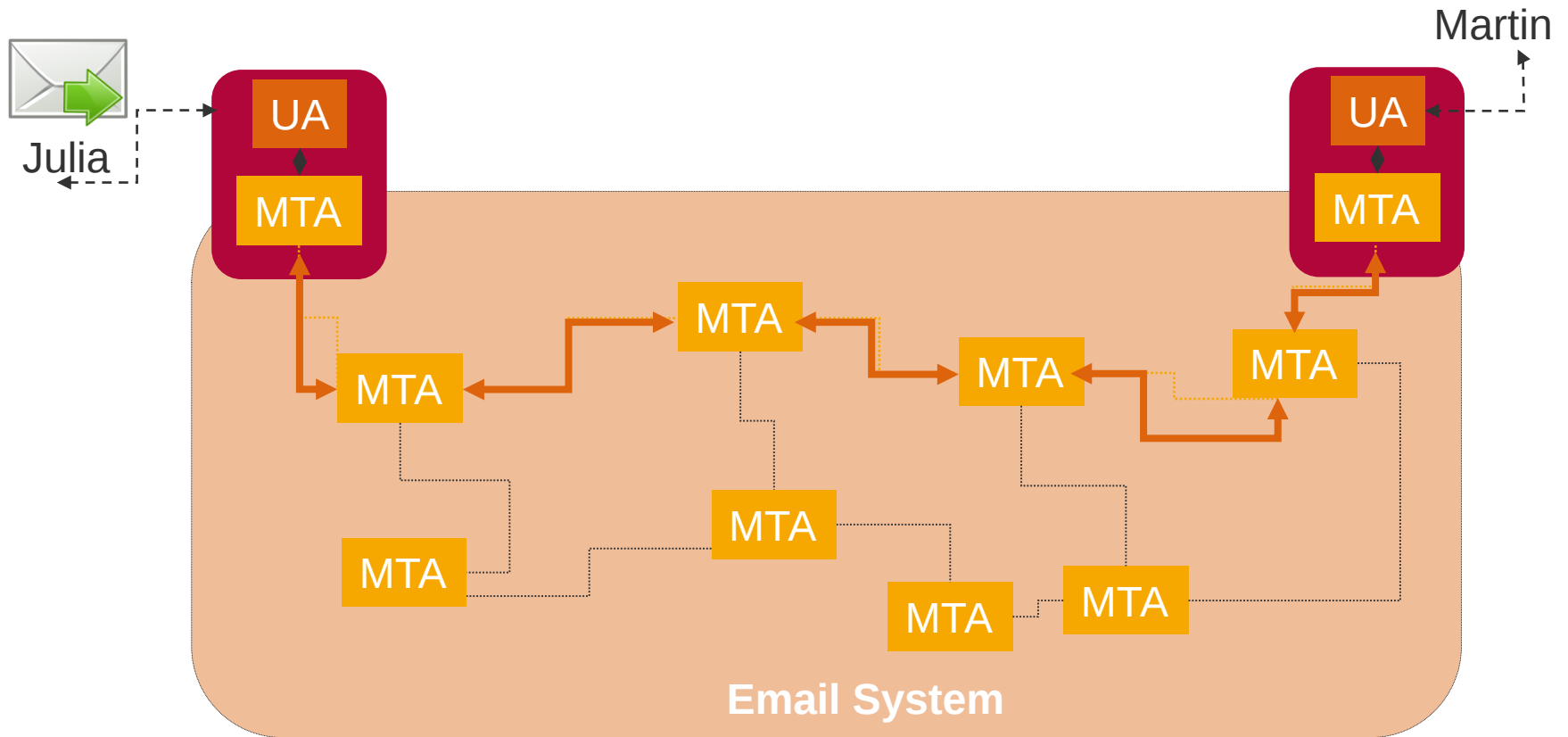
- Art der kodierten Daten, damit UA diese korrekt darstellen kann

Content-Transfer-Encoding:

- 7-Bit ASCII
- 8-Bit ASCII
- Base64 Encoding
- Quoted Printable Encoding
- User Defined Encoding

openHPI-Kurs: Wie funktioniert das Internet?
Email Protokolle

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam



Um einen den Transport einer Email über das Internet sicherzustellen, braucht es verschiedene Protokolle auf der Anwendungsschicht

SMTP - Email-Transport (1/2)

SMTP - Simple Mail Transfer Protocol

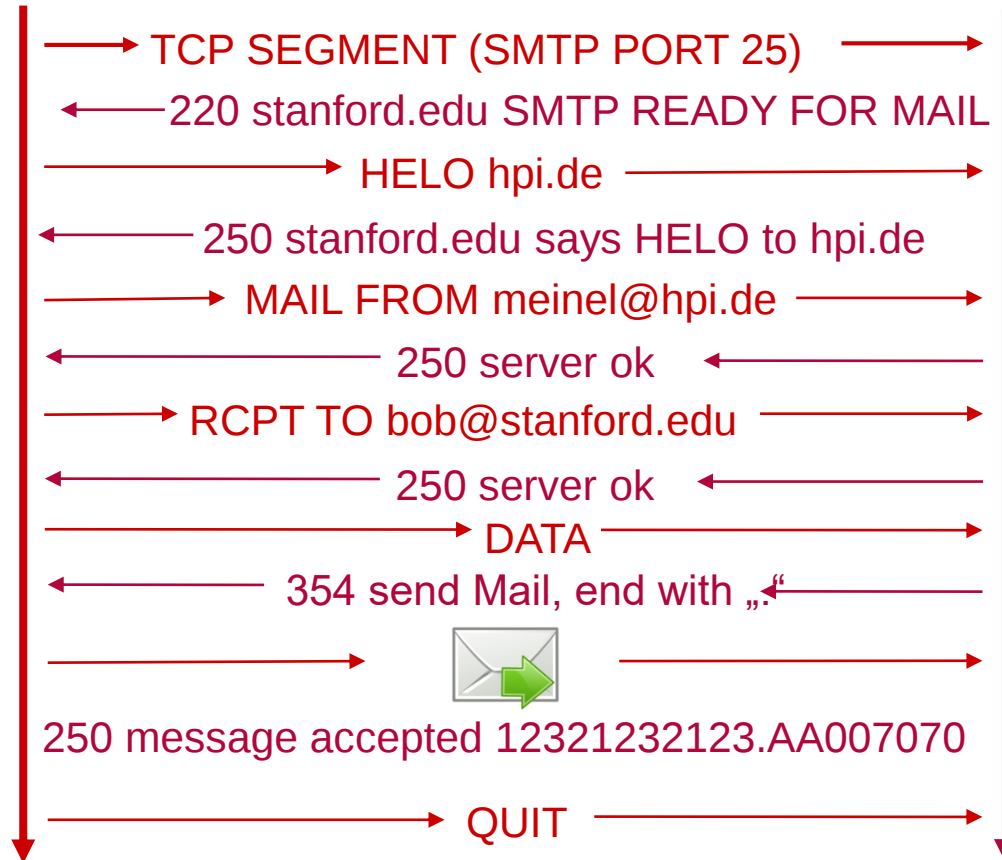
- Regelt Kommunikation zwischen MTAs, d.h. legt fest, wie eine Email-Nachricht übertragen wird
- Standard: RFC 5321 (seit 2008, vorher RFC 821 seit 1982)
- Ist stets mit TCP-Port 25 verknüpft
- Kommunikation via SMTP ist leicht nachzuvollziehen, da lediglich Nachrichten in 7-Bit-ASCII-Klartext ausgetauscht werden
- Hat mittlerweile viele Erweiterungen erfahren (ESMTP)

SMTP-Ablauf



Client
(Sender)

hpi.de



Server
(Receiver)

stanford.edu

POP3 - Email-Zustellung (1/3)

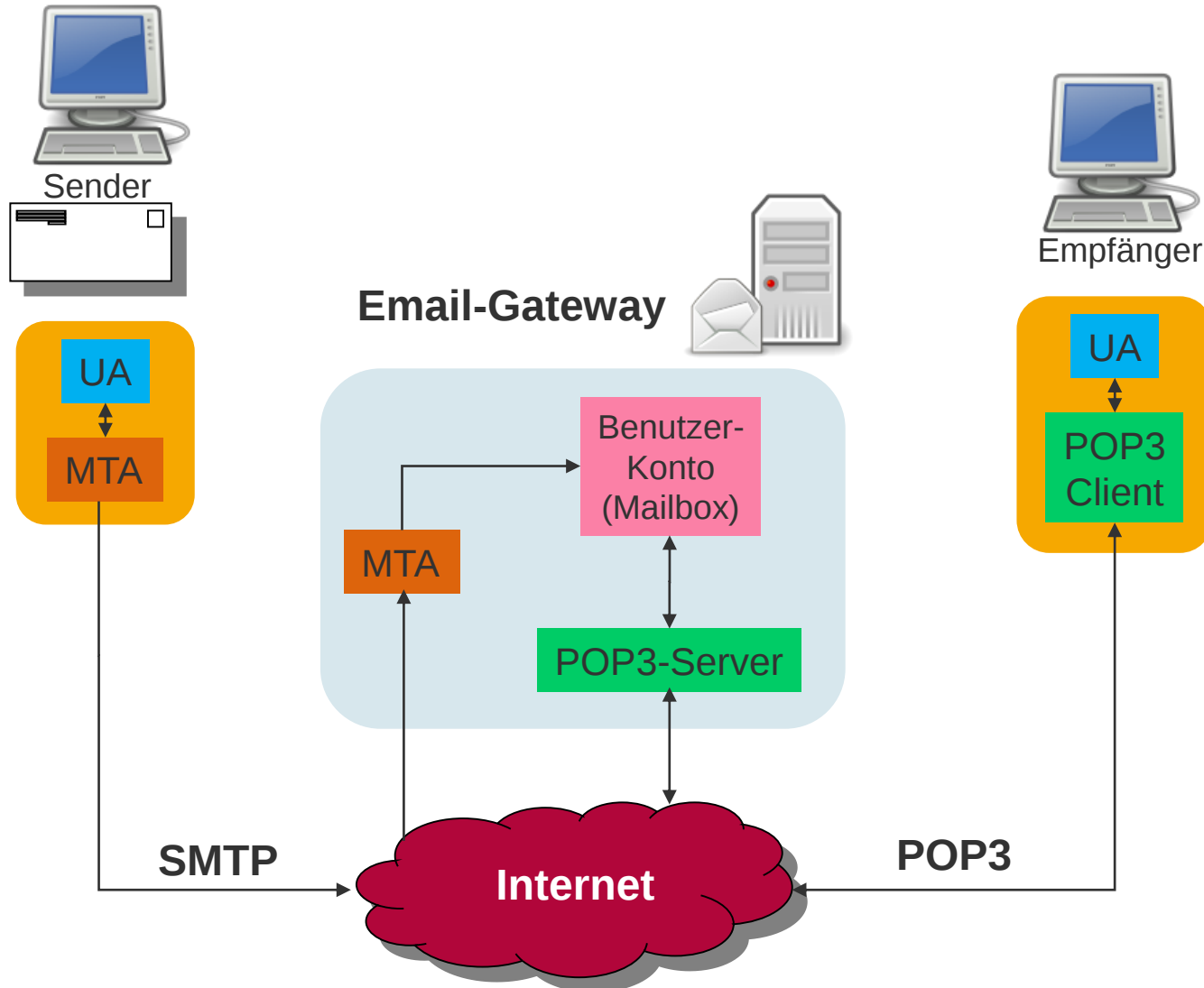
Email-Gateway sorgt für den Versand und Empfang von Email-Nachrichten via SMTP

- In LANs oder Einzelnetzwerken gibt es meist nur ein einziges dediziertes Email-Gateway
- Um ihr Email-Nachrichten zu senden und zu empfangen kommunizieren alle anderen Rechner im Netzwerk mit dem Email-Gateway

Für die Kommunikation mit dem Email-Gateway stehen zwei Protokolle bereit: POP3 und IMAP

- **POP3 - Post Office Protocol 3**
 - POP3 erledigt Kommunikation UA $\leftrightarrow$ MTA
 - Kommunikation erfolgt via TCP Port 110

POP3 - Email-Zustellung (2/3)



POP3 - Email-Zustellung (3/3)

Hauptaufgaben von POP3

- Anmeldung beim Email-Gateway
- Authentifikation des Nutzers durch Passwortabfrage
- Abruf von Email-Nachrichten aus dem Email-Account des Benutzers
- Löschen der abgefragten Email-Nachrichten aus dem Speicher des Email-Gateways
- Kopie der Email-Nachrichten des Nutzers auf dessen lokalen Rechner

Vorselektion oder paralleler Zugriff auf mehrere Email-Konten ist mit POP3 nicht möglich

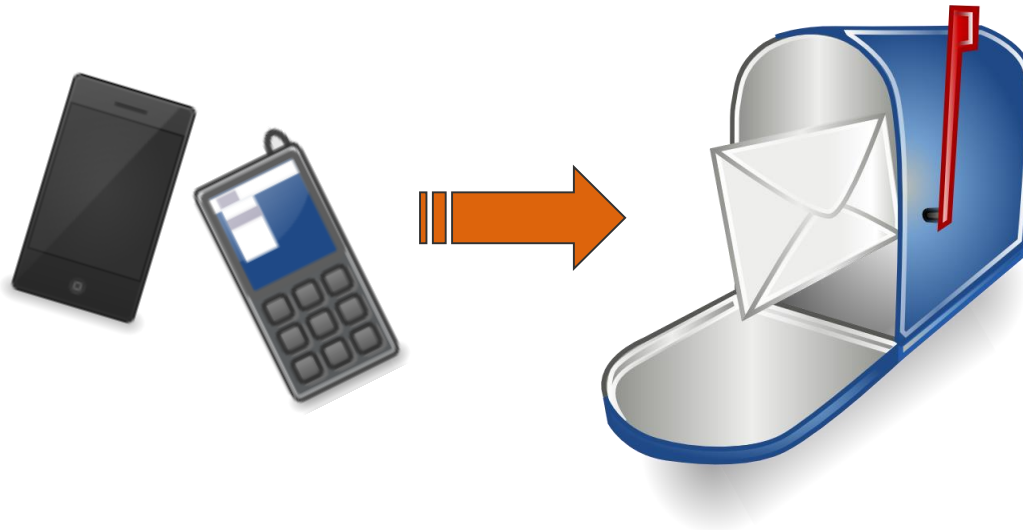
IMAP - Email-Zustellung (1/2)

IMAP - Interactive Mail Access Protocol (RFC 3501)

- Speziell für Nutzer die mit verschiedenen Devices ihre Mail lesen wollen, z.B. im Büro und zu Hause
- Im Unterschied zu POP3 ruft IMAP lediglich Email-Header aus der Mailbox des Nutzers vom Mailserver ab
- Nutzer hat zwar den Eindruck, als befänden sich die vom Mailserver abgerufenen E-Mails auf seinem lokalen Rechner, tatsächlich nutzt IMAP aber den Email-Gateway als zentrales Repository
- IMAP kommuniziert über TCP Port 143

IMAP - Email-Zustellung (2/2)

- Nutzer sieht Email-Header und entscheidet, ob die gesamte Nachricht heruntergeladen/bearbeitet werden soll (Preselection)
- Selektion und Sortieren finden auf dem Email-Gateway statt, sodass auch Zugriff von CPU-schwachen Endgeräten, z.B. Mobiltelefon, auf große Mailboxen möglich ist





openHPI-Kurs: Wie funktioniert das Internet?

WWW – World Wide Web

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

WWW – World Wide Web

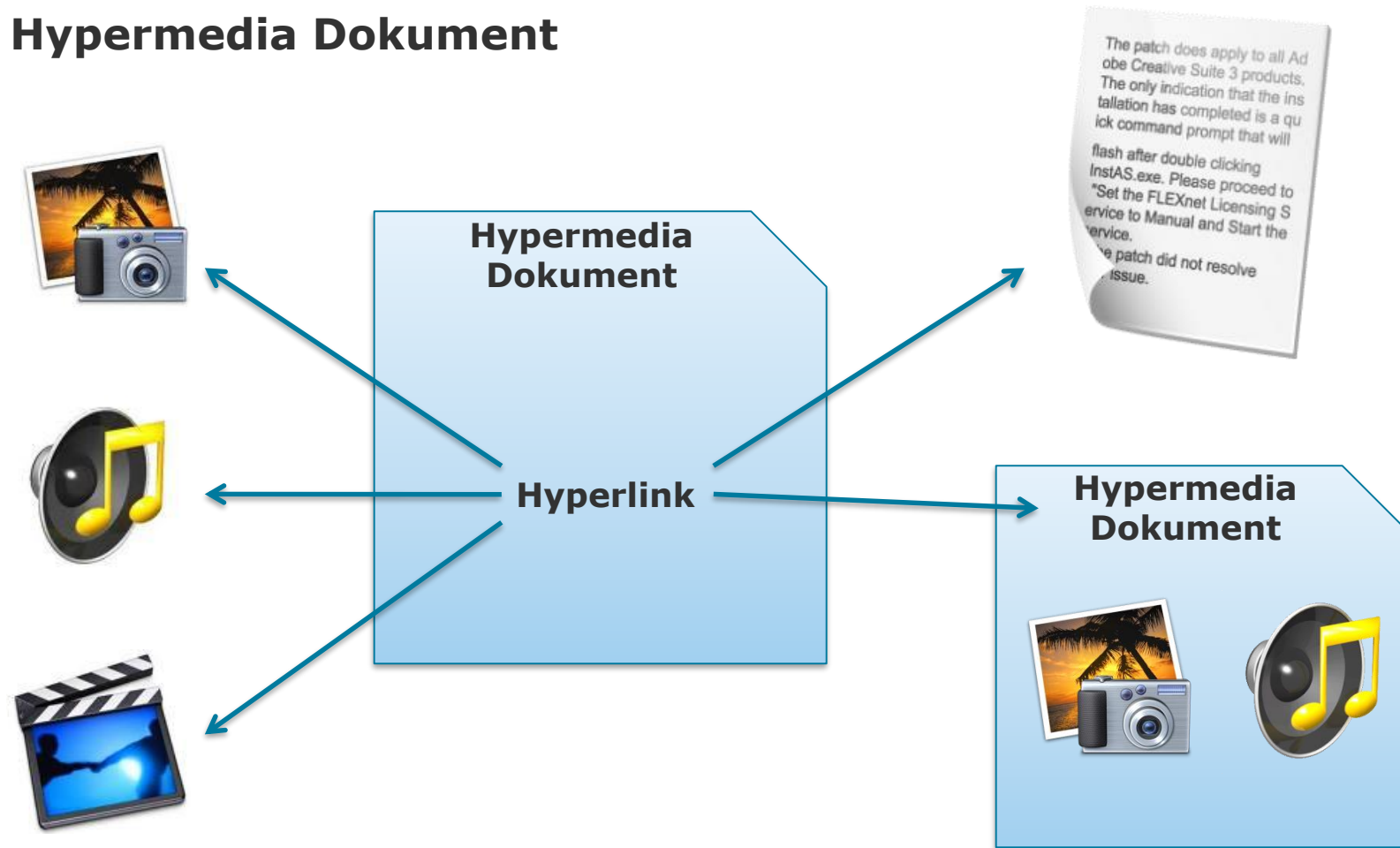
WWW – World Wide Web, Web – bietet eine riesiges, weltweit verteiltes Arsenal von Informationen und Zugängen zu den verschiedensten Services, auf die über das Internet mit Hilfe des **HTTP – Hypertext Transfer Protocol** – auf der Anwendungsschicht zugegriffen werden kann

- Anfang 1989/90 von Sir Tim Berners-Lee und Robert Cailliau im europäischen Kernforschungszentrum CERN entwickelt
- Mitte der 90er Jahre entwickelten hat sich das WWW dank der auch von Laien spielend intuitiv nutzbaren, grafischen Interfaces von **WWW-Browsern** zum bedeutendsten Medium der Gegenwart entwickelt
- Mit Hilfe von **Zwischensystemen** wie z.B.
 - Proxy-Server, Gateways und Cache-Speicherwird die Effizienz der HTTP-Kommunikation deutlich gesteigert

Hypermedia Dokumente und Hyperlinks (1/2)

- Die im WWW angebotenen Informationen und Service-Zugänge liegen in Form von **Hypermedia**-Dokumenten, synonym **Hypertext**-Dokumenten vor
- Hypermedia-Dokumente sind untereinander durch sogenannte **Hyperlinks**, kurz **Links** verbunden und bilden auf diese Weise ein riesiges Netzwerk von Informationen
- Entlang der Links können Nutzer sehr einfach in diesem Informationsnetzwerk navigieren

Hypermedia Dokument



HTML – Hypertext Markup Language

WWW-Dokumente sind in einer speziellen Beschreibungssprache, der **Hypertext Markup Language - HTML**, verfasst

Mit HTML können Autoren

- **Struktur** eines WWW-Dokuments (Gliederung in Überschriften, Absätze, Tabellen,...) beschreiben
- Hyperlinks einbinden
- multimediale Komponenten einbinden
- (bis zu gewissem Grad) die grafische Darstellung beeinflussen

URI – Uniform Resource Identifier

- Mit Hilfe spezieller **Tags** können in einem HTML-Dokument Referenzen „**Links**“ als passive Zeiger auf ein anderes Dokument gespeichert werden
- Die weltweit eindeutigen Identifikation der Dokumenten wird vermitteltels sogenannter **URL – Uniform Resource Locator** sichergestellt
- URLs können in Zeichenketten kodiert werden und setzen sich zusammen aus:
 - Name des Zugriffsprotokolls
 - Computername und
 - Dokumentenname

Browser und WWW-Server

Zugriff auf WWW-Dokumente ist nach **Client/Server-Paradigma** organisiert:

- Nutzer fordert über seinen WWW-Client, den Browser, ein WWW-Dokument an (Maus-Klick oder Eingabe URL)
- Browser kontaktiert den durch die URL spezifizierten WWW-Server und fordert gewünschtes Dokument an
- Server prüft Zugriffsberechtigung des Clients/Nutzers
- Besitzt Client/Nutzer Berechtigung für die Ressource, dann greift der WWW-Server auf sein lokales Filesystem zu und sendet die in der URL spezifizierte Datei zum anfragenden Browser
- Browser empfängt Dokument, interpretiert es und stellt es für den Nutzer dar

HTTP – Hypertext Transfer Protocol

Interaktion zwischen Browser und Server erfolgt auf der Basis des
Hypertext-Transfer-Protokoll - HTTP

- HTTP ist ein sehr **einfaches** und **zustandsloses**, darum schnelles Protokoll – Interaktion findet lediglich in Form eines einfachen Frage/Antwort-Verfahrens statt
- HTTP unterliegt Entwicklungsprozess
 - erste HTTP-Version (HTTP/0.9) entstand 1989/90 am CERN
 - 1992 HTTP/1.0
 - 1997 HTTP/1.1
 - **2015 HTTP/2**
 - heute sind HTTP/1.1 und HTTP/2 im Einsatz

Absicherung im WWW

HTTP-Kommunikation lässt sich auf zwei verschiedene Weisen absichern:

- **SSL – Secure Sockets Layer** - und TLS – Transport Layer Security – bieten sichere Kommunikationsinfrastruktur für HTTP-Kommunikation
- **S-HTTP** bietet eigenständiges auf HTTP basierendes sicheres Kommunikationsprotokoll (hat sich aber nicht durchgesetzt)

Konzentrieren uns in diesem Kurs auf das **Internetworking**

- **WWW** ist ein so umfänglicher und interessanter Internetdienst – **Killer Applikation im Internet** –, dass wir einen eigenen Online-Kurs → **Webtechnologien** anbieten



openHPI-Kurs: Wie funktioniert das Internet?
Streaming von Multimediadaten

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Streaming (1/2)

- Bislang vorgestellte Internetdienste arbeiteten nach dem Schema
 - erst Daten übertragen
 - dann Daten nutzen, z.B. Wiedergabe
- Problematisch bei Audio/Video-Daten
 - Hier müssten so erst riesige Datenmengen übertragen werden, bevor eine Wiedergabe stattfinden kann
 - Nicht tolerierbarer Zeitverzug!
- Wünschenswert ist eine kontinuierliche Medienwiedergabe bereits während der Datenübertragung



Media-Streaming

Streaming (2/2)

Streaming

- Medieninhalt wird
 - kontinuierlich und
 - bereits während der Datenübertragung
 - annähernd in Echtzeit – „**Realtime**“ – wiedergegeben
- Implementierung:
 - übertragene Daten werden zwischengespeichert – gepuffert – und
 - mit sehr kleiner Zeitverzögerung wenig später wiedergegeben
 - Zeitverzögerung nur wenige Sekunden, auch bei sehr großem Datengesamtvolumen

Echtzeitkommunikation (1/4)

Verschiedene Klassen von Echtzeit-Multimediaanwendungen:

Streaming von **gespeicherten** Audio- und Video-Datenströmen

- Daten liegen auf dem Server bereits vor der Übertragung vollständig und in komprimierter Form vor
- Client kann während der Übertragung innerhalb der übertragenen Datei navigieren
- Zeitverzögerung meist unter 10 Sekunden
- Übertragungsschwankungen sind nicht so kritisch
- Anwendung z.B.
 - ☐ Video on Demand
 - ☐ Internet-Radio
 - ☐ ...

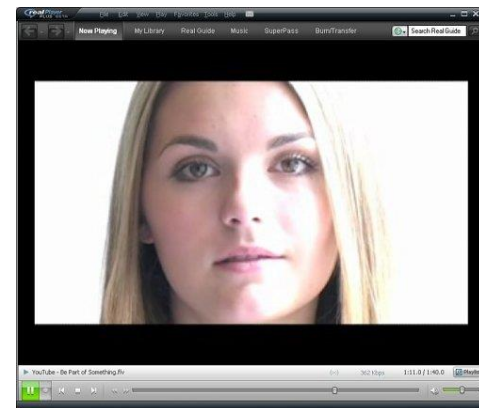


Echtzeitkommunikation (2/4)

Verschiedene Klassen von Echtzeit-Multimediaanwendungen:

Live-Streaming von Audio- und Video-Datenströmen

- Daten werden bereits **während ihrer Aufzeichnung** über das Internet übertragen
- Vergleichbar mit Live-Fernsehsendung
- Übertragene Daten werden nach Pufferung wiedergegeben (Ausgleich von Übertragungsverzögerungen/-Schwankungen)
- Geringe Zeitverzögerung
- Wird Datenstrom lokal abgespeichert, ist Navigation möglich
 - Multicasting-Anwendung



Echtzeitkommunikation (3/4)

Probleme bei Echtzeitkommunikation im Internet

Bezug von Multimediadaten über TCP Protokoll und TCP-Verbindungen viel zu aufwändig. Deshalb müssen folgende Probleme in Eigenregie gelöst werden

- Duplizierte Datenpakete → Verwendung von Sequenznummern
- Vertauschte Reihenfolge → Verwendung von Sequenznummern
- Verloren gegangene Datenpakete
→ Erkennung über Sequenznummern, Neuübertragung nicht sinnvoll
- Schwankungen in der Übertragungsverzögerung („**Jitter**“) → Zeitstempelinformationen und Wiedergabepuffer



Wünschenswert: **Ressourcenreservierung**

Wiedergabepuffer



Je größer der Puffer,

- desto **robuster** gegenüber Jitter und
- desto höher die **Wiedergabequalität**, aber
- aber desto größer auch die **Zeitverzögerung**

Beispiel: RTP – Real Time Transport Protocol

Real Time-Transport Protocol (RFC 3550)

- Wichtiges Protokoll zur Echtzeit-Übertagung von Medien-Daten
- Typischerweise in Kombination mit **RTSP – Real Time Streaming Protocol** – zur Steuerung genutzt
 - Setzt auf UDP als Transportprotokoll auf
 - Nutzbar, falls Datenverluste beim Medium verkraftbar sind
 - Medienstrom wird in Pakete aufgeteilt
 - Header der Pakete enthalten Informationen über Kodierung, Synchronisation sowie Paketreihenfolge (Sequenznummer)
- Empfänger nutzt Metadaten aus Header, um Datenstrom direkt zu rekonstruieren

Nachteil: Zuschauer benötigen passendes Browser-Plugin oder eine native App zum Abspielen

Pseudo-Streaming über HTTP (1/2)

Nutzerwunsch: Streaming über HTTP

Progressive Download – HTTP-Pseudo-Streaming

- Media-Daten werden vom HTTP-Server empfangen
- Medium wird stückchenweise in temporäre Datei kopiert
- Player spielt Medium von der Festplatte ab, sobald dies möglich ist
- Download läuft parallel und unabhängig dazu weiter
- Daten werden üblicherweise nur gepuffert, nicht abgespeichert
- Traffic läuft über HTTP, also Port 80 (Streaming-Ports werden oft in Unternehmensnetzwerken blockiert)

Merkmale:

- Download startet direkt nach Aufruf der URL über HTTP
- Abspielen wird initiiert, sobald genug Daten vorhanden sind, da am Anfang zuerst alle Metadaten übertragen werden
- Download wird fortgesetzt, bis die gesamte Datei empfangen wurde
→ **progressiv**
 - Nur auf dem Webserver komplett vorhandene Dateien können übermittelt werden → kein „echtes“ Live-Streaming
 - Qualitätsstufe ist festgelegt, kann nur durch erneuten Start des Download angepasst werden
 - Es wird potentiell mehr heruntergeladen als angeschaut wird
- Falls Nutzer an eine noch nicht geladene Stelle „spult“, so wird adaptiv ab dieser Stelle heruntergeladen → **Pseudo-Streaming**

Beispiel: Apple's HLS – HTTP Live Streaming

HTTP Live Streaming (HLS)

- Von Apple entwickeltes Streaming-Protokoll über **HTTP**
- War ursprünglich lediglich Teil von Apples OS X und iOS
- Anfangs wird eine Playlist mit Metadaten zu verschiedenen Sub-Streams bezogen
- Unterteilung des ganzen Streams in kleine HTTP-File-Downloads
- Jeder Download beinhaltet einen Teil des beliebig langen Streams
- Zur Anpassung an die verfügbare Datenrate kann **dynamisch** zwischen Sub-Streams mit verschiedener Qualität gewählt werden

m3u8

- Dateiformat für HLS-Playlisten
- Spezifiziert die Quellen für einen oder mehrere Sub-Streams
- Das Video selbst ist nicht enthalten, muss vom Player bezogen werden



openHPI-Kurs: Wie funktioniert das Internet?

Soziale Medien

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Medientypen

Klassische Medien

- Printmedien: Zeitung, Buch, Plakat, Flugblatt
- Radio, Fernsehen

Neue Medien

- CD, DVD, Blu-ray
- Email
- WWW

→ Social Media / Soziale Medien

Internetanwendungen: Social Media

Digitale Medien, Internet-Plattformen und -technologien, die es Nutzern erlauben

- sich in einfachster Weise untereinander über das Internet auszutauschen
- einzeln oder gemeinsam multimediale Inhalte zu erstellen

In Sozialen Medien verschwimmt traditionelle Rollenverteilung in Inhalte-Ersteller, Inhalte-Konsument. Nutzer ist Ersteller und Konsument in Einem

Es gibt viele verschiedene Kategorien sozialer Medien

- Blogs, Foren, Wikis, ...
- Media Sharing, z.B. Videoportale, Fotodienste, ...
- Soziale Netzwerke, z.B. Facebook, LinkedIn, ...
- Social Network Games, z.B. Quizduell, ...
- ...

Internetanwendungen: Social Media

Soziale Medien sind

- global verfügbar
- multimedial
- leicht zugänglich
- benutzerfreundlich und
- sehr aktuell

Was macht Social Media „sozial“ (=gesellschaftlich)?

- Vernetzung
- Interaktion
- Kollaboration
- Nutzer besitzen gleichzeitig die Rolle als Sender und Empfänger
- ...

„Das“ Soziale Netzwerk

- Gegründet im Februar 2004 von den Harvard-Studenten Marc Zuckerberg, Dustin Moskovitz, Chris Hughes und Eduardo Saverin
- Hauptkomponenten von Facebook:
 - Persönliches Profil
 - Fotoalben
 - Timeline
 - Messaging
 - Seiten
 - Apps und Spiele, Suche, Orte, Events, „Like“, ...



Marc Zuckerberg

(CC BY 2.0) Foto editiert. Original unter <https://www.flickr.com/photos/presidenciamx/15128137116/>



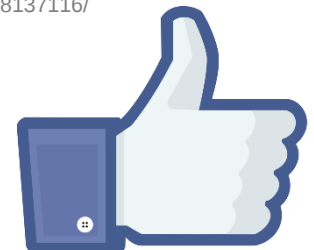
„Das“ Soziale Netzwerk

- Mittlerweile nicht mehr nur im Browser, sondern auch verfügbar in Form von **nativen Apps** auf nahezu allen aktuellen Mobilgeräten
- Kritisiert u.a. für mangelnden Datenschutz, Machtmissbrauch, Zensur, „Facebook-Partys“



Marc Zuckerberg

(CC BY 2.0) Foto editiert. Original unter
<https://www.flickr.com/photos/presidenciamx/15128137116/>



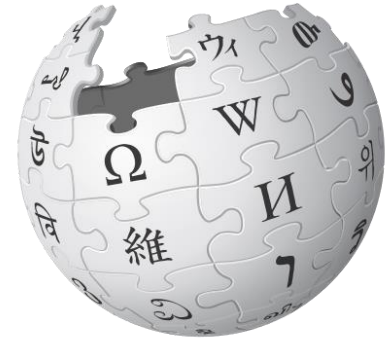
Geschäftliches Soziale Netzwerk

- 2003 in den USA gegründet
- Heute über 450 Millionen Nutzer
- Ausrichtung auf geschäftliche und berufliche Kontakte
- Besitzt einige typische Funktionen sozialer Netzwerke
 - Persönliches Profil mit beruflichem Lebenslauf, Unternehmensprofile
 - Kontakte
 - Thematische Gruppen
- Besonderheiten
 - Eigenes Profil in mehreren Sprachen möglich
 - Weiterempfehlen von Kontakten: Bestätigung einzelner Stärken oder Verfassen von Empfehlungsschreiben/Bericht



Gemeinschaftlich bearbeitetes **Online-Lexikon**

- Gegründet im Januar 2001
- Betrieben von Wikimedia Foundation (Non-Profit)
- Ca. 39,5 Millionen Artikel (Mitte 2016), davon knapp 2 Millionen deutschsprachige Artikel
- Durchschnittlich 52 Bearbeitungen pro Artikel
- Prinzip des **kollaborativen Schreibens**:
 - jeder kann schreiben, ändern, bearbeiten
 - Änderungen gehen nach Sichtung durch erfahrene Nutzer online
- Gewählte „Administratoren“ achten auf Einhaltung der Regeln und haben auch das Recht, Änderungen, Artikel und Nutzer zu sperren



WIKIPEDIA
Die freie Enzyklopädie

By Wikimedia Foundation CC BY-SA 3.0
<http://creativecommons.org/licenses/by-sa/3.0>
via Wikimedia Commons



openHPI-Kurs: Wie funktioniert das Internet?

Social Media – Weitere Beispiele

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Internetanwendung: Social Media

Soziale Medien sind heute fester Bestandteil des Lebens

- Anfangs Nutzung nur im **Browser** am PC
- Heute oft **native Apps auf Mobilgeräten**, die Nutzer noch mehr Komfort bieten und spezielle Fähigkeiten der Geräte ausnutzen
 - Login bleibt auch über Sessions hinweg erhalten
 - optimiert für kleines Display, Bedienung mit Fingern
 - Hintergrundaktivität, Benachrichtigungen
 - Suspend-Modus (App kann „schlafengelegt“ werden, ohne Ressourcen zu verbrauchen) und schnelles Reaktivieren
 - Integration von GPS-Sensor, Handykamera, Telefonbuch, ...
 - Offlinefähigkeit: Ansichten und Daten werden zwischengespeichert und sind auch ohne Internetverbindung verfügbar



WhatsApp / XMPP (1/3)

WhatsApp

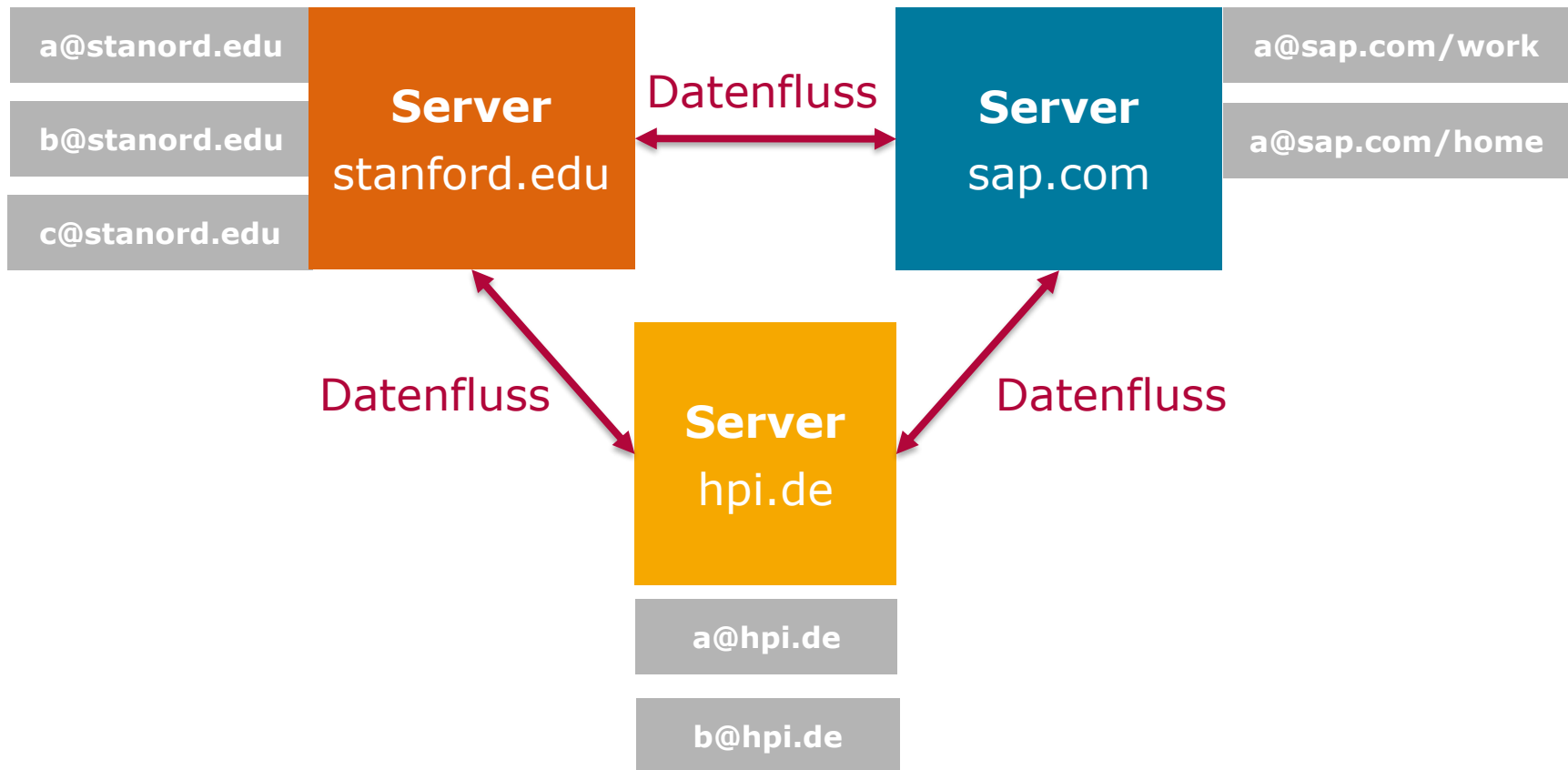
- Sehr beliebter Dienst, der die Kommunikation über Text-, Bild-, Audio- und Video-Nachrichten zwischen zwei Teilnehmern oder in Gruppenchats ermöglicht → **Instant-Messaging-Dienst**
- Ist verfügbar auf alle gängigen Smartphone-Plattformen aber kein Dienst eines Mobilfunkbetreibers, wie z.B. SMS
- Nutzt Internet zur Datenübertragung über das **XMPP** (RFC 6120) Protokoll auf der Anwendungsschicht
 - Setzt auf TCP auf
 - Ermöglicht Echtzeit-Austausch von strukturierten Daten zwischen zwei oder mehreren Netzwerk-Teilnehmern
 - Besitzt dezentralisierte Natur ähnlich zu Email

WhatsApp / XMPP (2/3)

XMPP – Extensible Messaging and Presence Protocol

- Basiert auf Client-Server-Architektur
- Einzelne Server werden typischerweise über Domains kontaktiert
- Dezentrale Natur ermöglicht es Jedem, Server auf eigenen Domains zu betreiben
- Jeder Nutzer benötigt eine JID (Historisch: Jabber-ID)
 - JID hat Gestalt: Nutzer-Name @ Domain-Name, z.B.
 - [nutzer@domain.de](#)
 - Spezifizierung einer „Ressource“, ermöglicht mehrfache Logins, z.B. [nutzer@domain.de/home](#)
- Über ihre JID können Nutzer eindeutig kontaktiert werden
- Nutzer sind so einem Server zugeordnet und empfangen über diesen alle relevanten Informationen

XMPP-Dezentralisierung



Instagram / HTTP REST (1/3)

Instagram

- Sehr beliebte Plattform zum Teilen von Bildern und Videos
- Verfügbar sowohl über Webbrowser, als auch über dedizierte Apps für Android, iOS und Windows 10 mobile
- Bietet Funktionalität eines Sozialen Netzwerkes, das auch mit anderen sozialen Netzwerken, wie z.B. Facebook, Twitter etc. interagieren kann
- Funktioniert als Webanwendung
 - Läuft somit über HTTP(S)
 - Realisiert einige Prinzipien sogenannter **REST**-Architekturen
 - Bietet Clients **APIs** an, um mit den Funktionen über HTTP mit der Instagram-Anwendung zu interagieren
 - Für die Modellierung der Interaktionen werden dabei gewisse **RESTful**-Praktiken beachtet

Instagram / HTTP REST (2/3)

HTTP-REST (HTTP-Representational State Transfer)

- Satz von Regeln/Prinzipien für die Benutzung von HTTP bei Webanwendungen
- Hilft bei der konzeptionellen Modellierung der Anwendung
- Es werden sog. **Ressourcen** in den Fokus genommen (vereinfacht)
 - Ressourcen sind Webseiten, Skripte, Bilder usw.
 - REST-Server besitzt Ressourcen und stellt diese bereit
 - REST-Client greift auf Ressourcen zu und nutzt diese
 - Jede Ressource besitzt eine **URL** (Uniform Resource Locator)
 - Als **Repräsentationen** der Ressourcen kommen verschiedene Formate, wie z.B. XML, Text, JSON etc. zum Einsatz
 - Weiterhin werden **Methoden** zum Zugriff auf die Ressourcen bereitgestellt (GET, PUT, DELETE, POST)

Instagram / HTTP REST (3/3)

Einige REST-Regeln

- REST-Server weiß nur über seinen eigenen Zustand bescheid, Clients verwalten ihren Status selbst
 - Authentifizierung und Session-Verwaltung für Ressourcen sollte über HTTPS-Mechanismen verfügen
- **Transfer** von einem Zustand in einen Anderen wird vom Client über die **Repräsentation** einer Ressource ermöglicht, z.B.
 - Client werden über Links auf einer Webseite zu einen Transfer in einen anderen Zustand veranlasst
- Manipulation von Ressourcen ist nur über deren URLs erlaubt unter Einsatz der dafür vorgesehenen Methoden
- ..



openHPI-Kurs: Wie funktioniert das Internet?

Absicherung auf der Anwendungsschicht

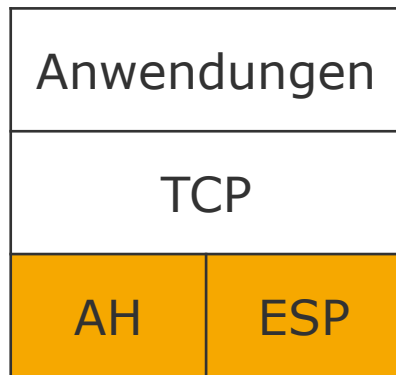
Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

TCP/IP Sicherheitsarchitekturen (1/5)

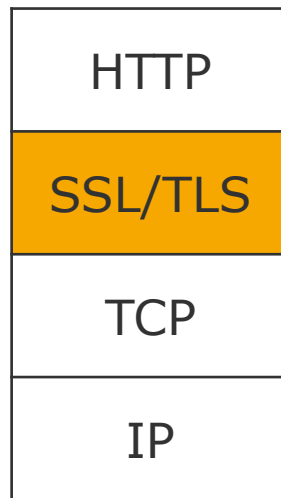
- Um bestimmte Sicherheitsziele und –Anforderungen im Internet zu erreichen, müssen einzelne Internetprotokolle modifiziert und um bestimmte Funktionen ergänzt werden
- Im Ergebnis entsteht eine sogenannte **Sicherheitsarchitektur**
- Hatten schon gesehen, dass **Absicherung der Protokolle** im TCP/IP-Protokollstapel auf unterschiedlichen Schichten erfolgen kann, z.B. auf der
 - Netzwerkschicht
 - Transportschicht
 - Anwendungsschicht

TCP/IP Sicherheitsarchitekturen (2/5)

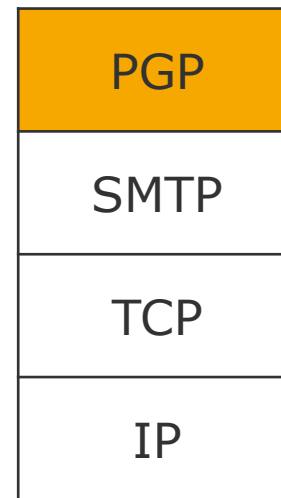
Unterschiedliche Sicherheitsarchitekturen:



IPsec



SSL / TLS



PGP

TCP/IP Sicherheitsarchitekturen (3/5)

Absicherung auf der Netzwerkschicht

- Alle Protokolle und Anwendungen höherer Schichten können unverändert weiterbetrieben werden
- Absicherung recht aufwendig, da Zwischensysteme einbezogen werden müssen und mit aktiven Schutzmechanismen auszustatten sind
- Spezifischen Anforderungen des Anwenders kann nur sehr allgemein entsprochen werden
- **Beispiel:**
 - **IP Security** kurz **IPsec** – Sammlung verschiedener Standards

TCP/IP Sicherheitsarchitekturen (4/5)

Absicherung auf der Transportschicht

- Die eigentlichen Anwendungen bleiben unberührt und können in ursprünglicher Ausgestaltung ohne Anpassungsaufwand betrieben werden
- Da es um Verbindungen auf der Transportschicht geht, sind Sicherheitsmechanismen nur auf den beteiligten Endsystemen erforderlich, Zwischensysteme entlang der Verbindung sind nicht betroffen
- **Beispiele:**
 - **SSL** – Secure Socket Layer (bis SSL 3.0)
 - **TLS** – Transport Layer Security (SSL 3.1 heißt TLS 1.0)

TCP/IP Sicherheitsarchitekturen (5/5)

Absicherung auf der Anwendungsschicht

- Sicherheitsmaßnahmen können gezielt auf spezielle Bedürfnisse einzelner Nutzer oder Anwendungen zugeschnitten werden
- Sehr kostspielig, da jede Implementierung von Neuem mit gewünschten Sicherheitsmechanismen ausgestattet werden muss
- **Beispiel** für Sicherheitsarchitektur auf Anwendungsebene:
 - **PGP** (Pretty Good Privacy) – System zum sicheren E-Mail-Versand

Absicherung auf der Anwendungsschicht: Beispiel: PGP – Pretty Good Privacy (1/2)

- Populäres Beispiel einer Sicherheitsarchitektur auf Anwendungsebene zum sicheren Versenden und Empfangen verschlüsselter Emails
- Bietet vollständige, funktionsfähige Arbeitsumgebung für bekannte und ineinandergreifende Methoden zum abgesicherten Email-Versand ohne eigene proprietäre Entwicklungen
- 1991 entwickelt von Phil Zimmermann (RFC 1991, später ersetzt durch RFC 4880, OpenPGP)
- GnuPG (GNU Privacy Guard) ist als kostenlose Alternative zum kommerziellen PGP-Produkt für viele Plattformen frei erhältlich

Absicherung auf der Anwendungsschicht: Beispiel: PGP – Pretty Good Privacy (2/2)

- PGP bietet
 - Verschlüsselung von Emails – Geheimhaltung
 - Authentifikation der Kommunikationspartner
 - Sicherung der Nachrichten-Integrität über digitale Signaturen
 - Komprimierungsverfahren für Emails