Vorlesung Einführung in Rechnernetze
10. Anwendungssysteme

Prof. Dr. Martina Zitterbart

Dipl.-Inform. Martin Florian, Markus Jung (M.Sc.), Matthias Flittner (M.Sc.)

[zitterbart | florian | m.jung | flittner]@kit.edu

Institut für Telematik, Prof. Zitterbart

Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Basisdienst DNS
3. Anwendungsbeispiel E-Mail
4. Anwendungsbeispiel World Wide Web

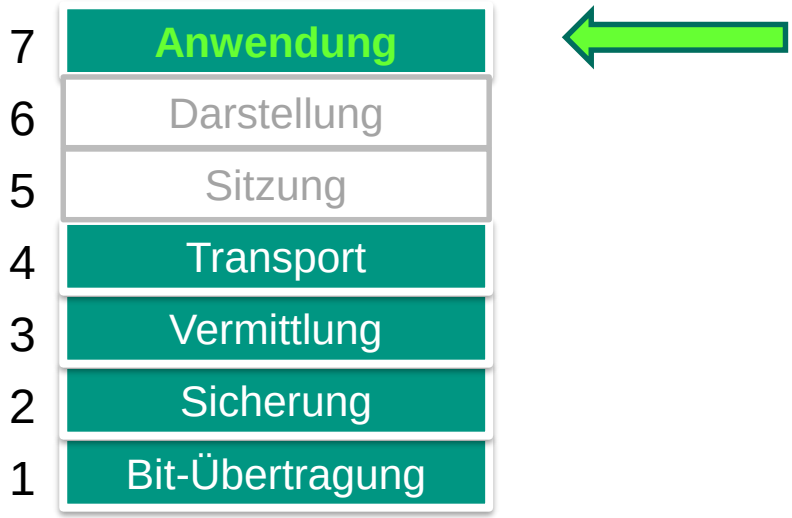
Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Basisdienst DNS
3. Anwendungsbeispiel E-Mail
4. Anwendungsbeispiel World Wide Web

Einordnung

- Wir befinden uns auf Schicht 7 des OSI-Referenzmodells



10.1 Einführung

- Anwendungen
  - Stellen Nutzern des Systems Dienste zur Verfügung
  - Nutzen Kommunikationsdienste der darunter liegenden Schichten
    - Zuverlässige Übertragungsdienste, etc.

- Beispiele für Anwendungen
  - World Wide Web
  - E-Mail
  - Datenaustausch
  - Fernzugriff
  - Voice-over-IP
  - …

- Dafür erforderlich: Basisdienste
  - Domain Name System (DNS)
  - Lightweight Directory Access Protocol (LDAP)
  - …

Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Basisdienst DNS
3. Anwendungsbeispiel E-Mail
4. Anwendungsbeispiel World Wide Web

10.2 Basisdienst Domain Name System (DNS)

- Ziel: Verwendung logischer Namen statt IP-Adressen zur Adressierung von Rechnern (insbesondere Servern)
  - Vergleichbar einem Telefonbuch
  - Namen sind für Menschen leichter zu merken als IP-Adressen
- Beispiel: telematics.tm.kit.edu → 129.13.182.160
- Aufgabe
  - Zuordnung zwischen Name und IP-Adresse
- Funktionalitäten
  - Registrierung von Namen und zugehörigen IP-Adressen
    - Vergabe für „.de“-Domain durch DENIC eG, …
  - Auflösung von Namen in IP-Adressen
    - Verteilte Datenbank mit einer Hierarchie von sog. Name-Servern (DNS-Servern)

[RFC1034/1035]

How the DNS works

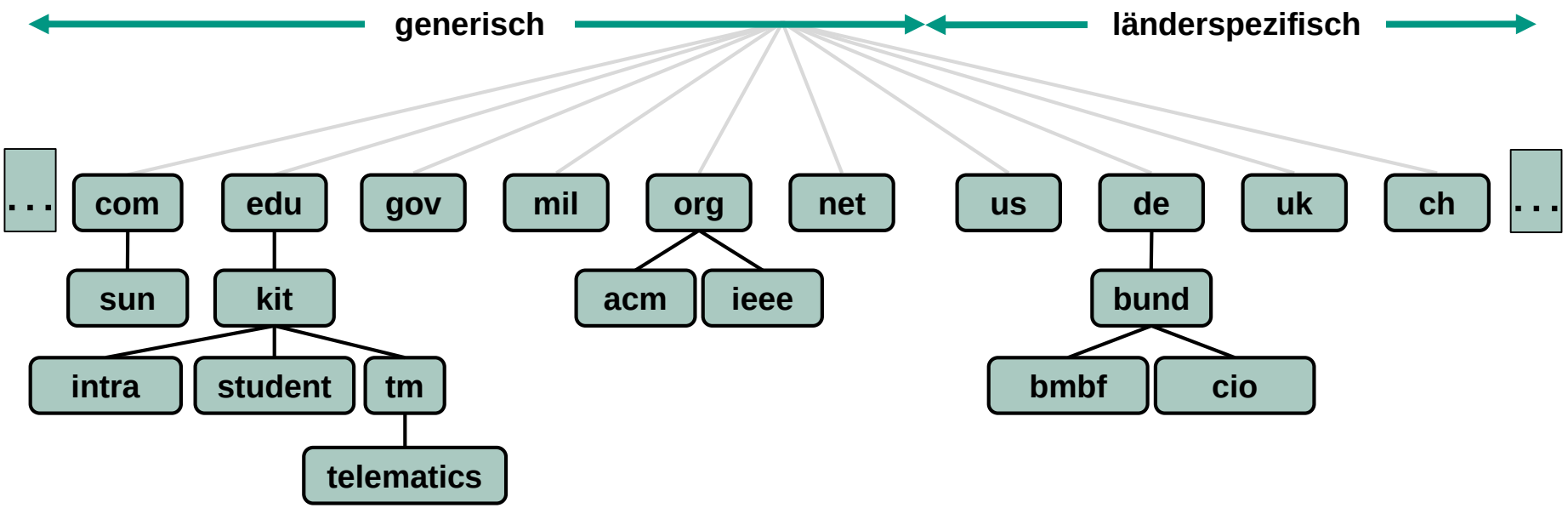
https://www.youtube.com/watch?v=2ZUxoi7YNgs

Exkurs: Bedeutung des DNS heutzutage

„Ausfall der .de-Domain“ am 12. Mai 2010

[Heis10b]

Struktur des Namensraums



- Struktur
  - Hierarchische Struktur: *Subdomäne* **.** *Domäne* **.** *Top-Level-Domain (TLD)*
  - Getrennt durch Punkte
- Top-Level-Domains
  - Länder-Domains (.de, .us, .uk, …) nach ISO 3166-1
  - Generische TLDs (.gov, .edu, …, .museum, …, .xxx) festgelegt durch die ICANN (Internet Corporation for Assigned Names and Numbers)

Einführung neuer Top-level Domains (New gTLD)

- ICANN startet Vergabe für neue Top-Level-Domains (New gTLDs)
  - Bewerbungsphase vom 12. Januar 2012 bis 20. April 2012
  - Für Organisationen und Unternehmen
  - Nicht-Lateinische Schriftzeichen
  - Bsp.: .music, .eco, .green, .africa, .berlin, .游戏

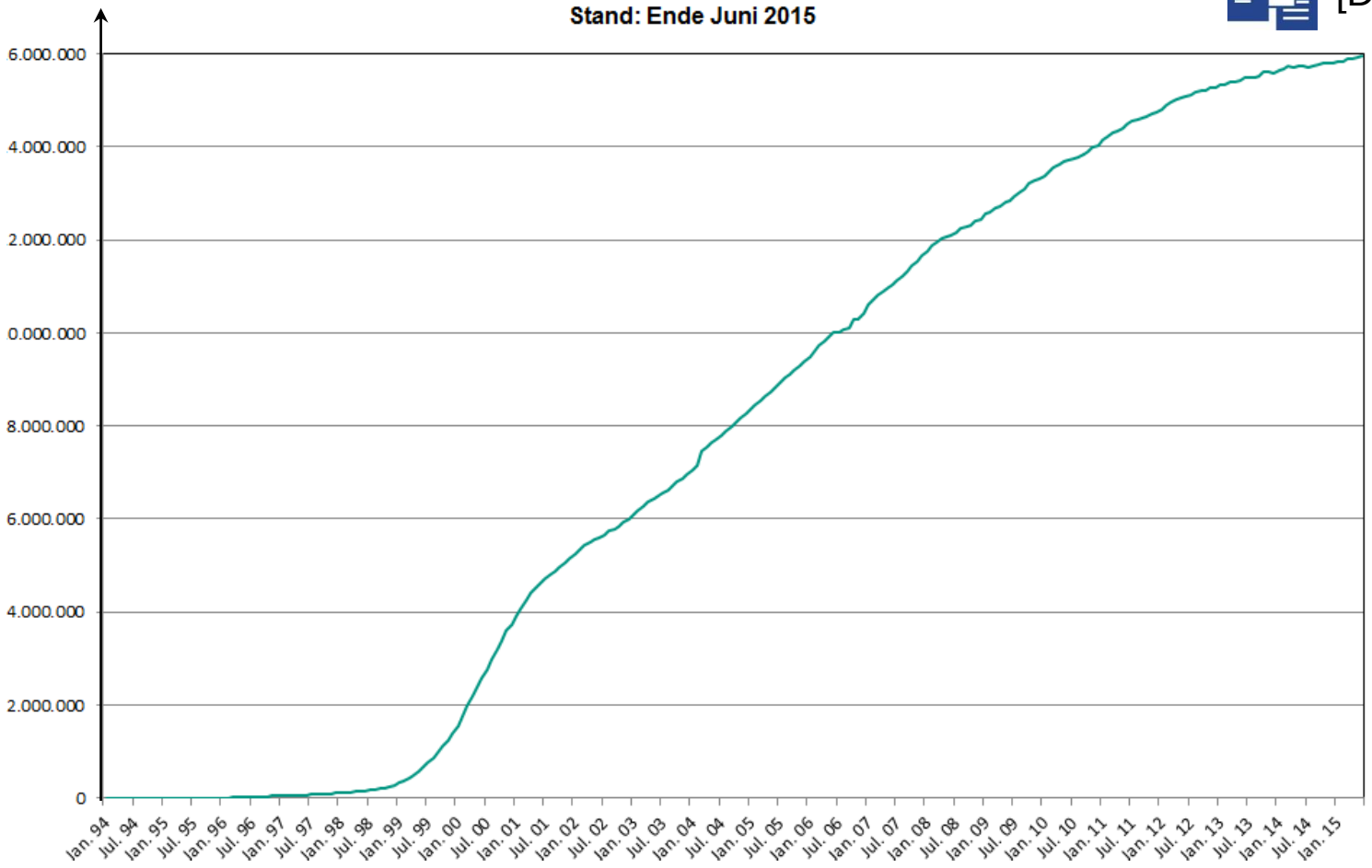
- Allerdings hohe Auflagen
  - Bewerbungsgebühr beträgt 185.000 Dollar
    - Spezielle Unterstützung für Bewerber aus Entwicklungsländern
  - Genaue Prüfung der Bewerber
    - Versuch, sich gegen alle Arten von Missbrauch abzusichern
  - Strikte Einhaltung von Markenschutzrechten

[ICANN]

- Seit Oktober 2013 schrittweise Einführung neuer Domians
  - Nach interner Prioritätenliste

Namensverwaltung der .de-Domäne: DENIC

Anzahl .de Domains



[DENI15]

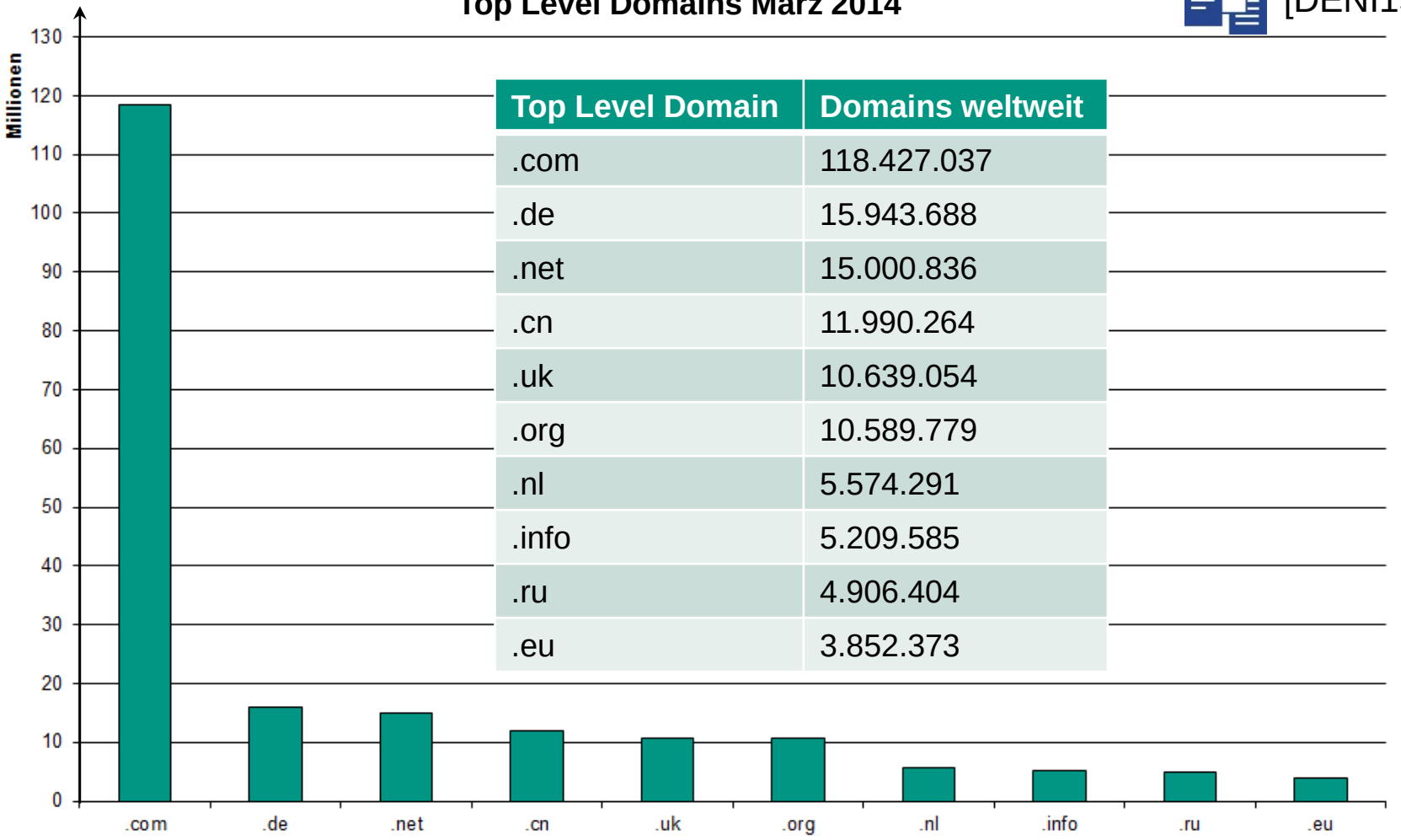
Zahl der .de-Domains am 1. Juli 2015: 15.943.688

Namensverwaltung: Die größten TLDs

Anzahl Domains

Top Level Domains März 2014

[DENI15]



Top Level Domain	Domains weltweit
.com	118.427.037
.de	15.943.688
.net	15.000.836
.cn	11.990.264
.uk	10.639.054
.org	10.589.779
.nl	5.574.291
.info	5.209.585
.ru	4.906.404
.eu	3.852.373

10.2.1 Basisdienst DNS: Aufbau und Eigenschaften

- Verteilte Datenbank mit einer Hierarchie von sog. Name-Servern (DNS-Servern)

[RFC1034/1035]

  - Vor DNS wurden die Namen in der „hosts.txt“-Datei verwaltet, die regelmäßig verteilt wurde
- Arbeitet nach dem Client/Server-Modell
  - Server kann Anfrage an andere Server weiterleiten
- Protokoll der Anwendungsschicht, das Kommunikation mit Name-Servern regelt
  - Über UDP realisiert – Port 53
  - Wieso nicht TCP?

- Basisdienst im Internet – keine eigentliche Anwendung
  - Komplexität ist am Rande des Netzes lokalisiert
  - → Internet Design Philosophie! (Ende-zu-Ende Paradigma)
- Verwendung
  - Wird von anderen Protokollen der Anwendungsschicht eingesetzt
    - HTTP, SMTP, FTP, …

Weitere Dienste von DNS

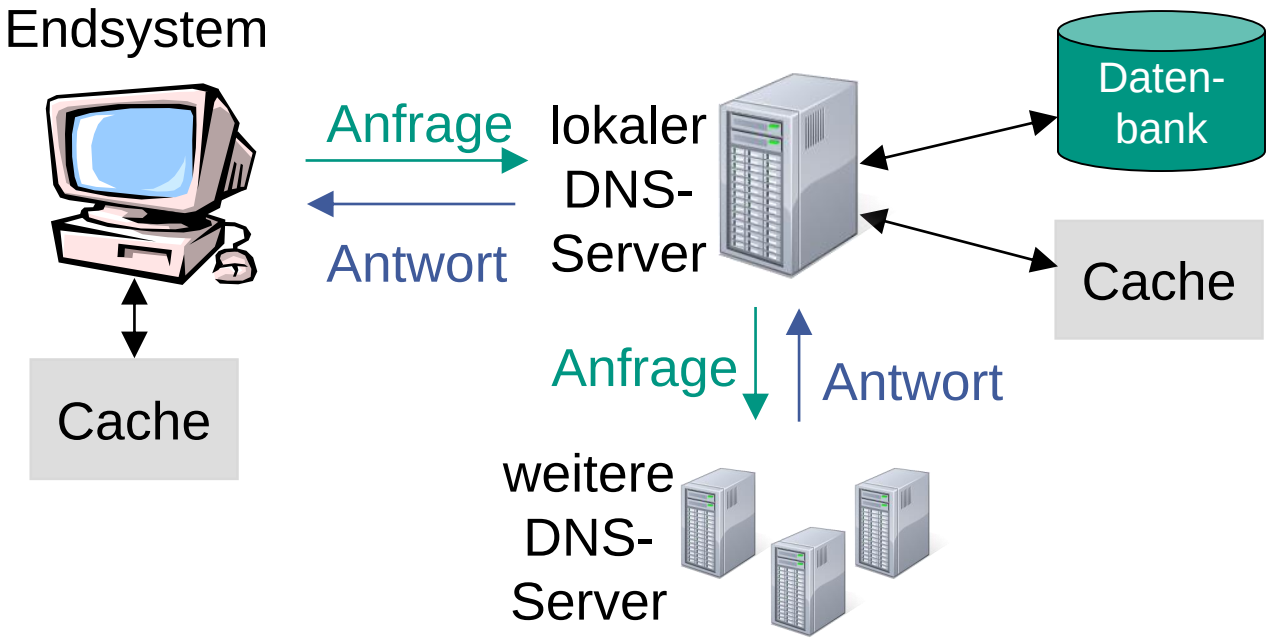
- Host Alias
  - Endsysteme mit komplizierten Namen können über einen oder mehrere Alias-Namen verfügen
    - Alias-Namen sind typischerweise einfacher zu merken
    - DNS kann für einen Alias-Namen den „eigentlichen“ Namen liefern (kanonischer Name)

- Mail Server Alias
  - E-Mail-Adressen sollten so gestaltet sein, dass sie leicht zu merken sind, z.B. praesident@kit.edu
  - Der Host-Name des Mail-Servers ist in vielen Fällen komplexer gestaltet
    - z.B. für praesident@kit.edu der scc-mailin-01.scc.kit.edu

- Lastverteilung
  - Bei redundant ausgelegten Servern (z.B. Server-Cluster) kann eine Menge von IP-Adressen mit einem Namen assoziiert sein
    - DNS-Server antwortet mit gesamter Menge der zugehörigen IP-Adressen, verändert aber die Reihenfolge, in der diese genannt werden

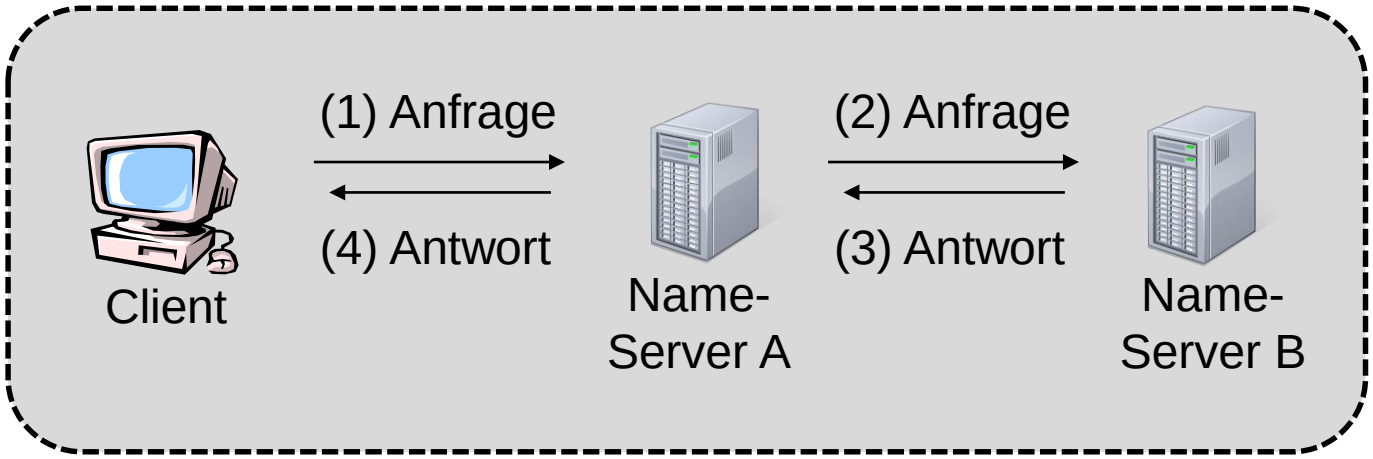
10.2.2 DNS-Anfrage

- Wenn ein Endsystem den gefragten Namen nicht kennt (nicht im eigenen Cache hat), wird lokaler DNS-Server befragt
- Lokaler DNS-Server antwortet entweder
  - aus seiner eigenen Datenbank mit Einträgen (falls autoritativ)
  - aus seinem Cache gespeicherter Antworten auf vorangegangene Anfragen
  - nach Befragung anderer DNS-Server, falls er die Antwort nicht liefern kann



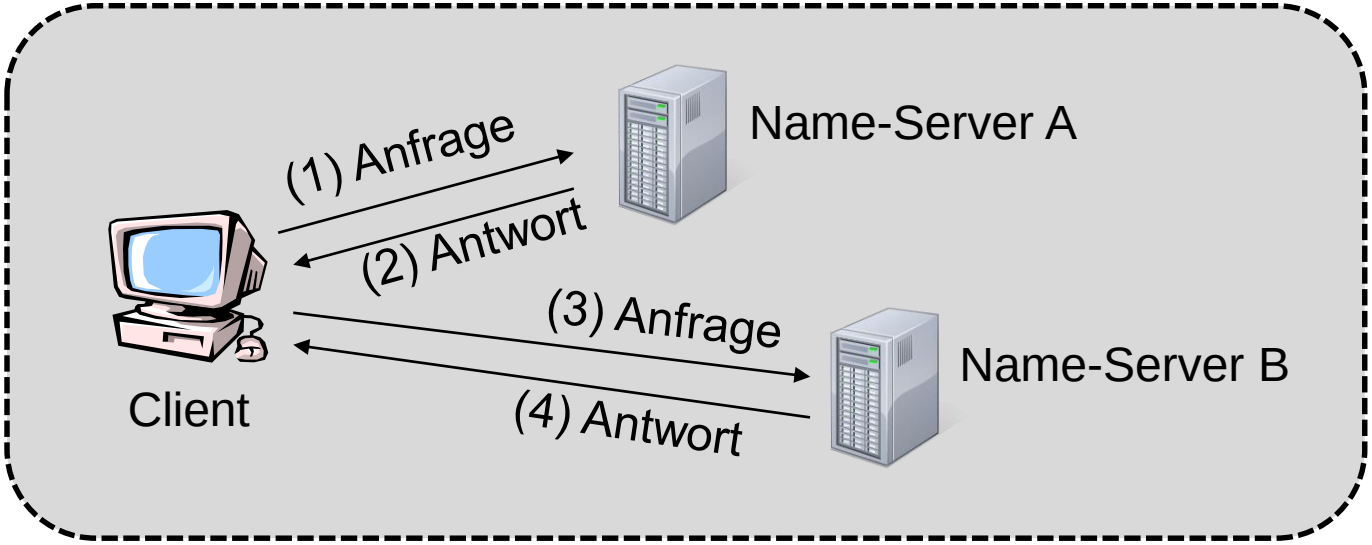
DNS-Anfragen: Rekursive Anfrage

- Rekursive Anfrage
  - Kennt angefragter Server die Antwort nicht, befragt er weitere Server, bis er die Antwort zurückliefern kann
  - Vorteil: Geringer Aufwand für Client, da Arbeit zur vollständigen Namensauflösung dem Server überlassen wird



DNS-Anfragen: Iterative Anfrage

- Iterative Anfrage
  - Kennt angefragter Server die Antwort nicht, kann er mit Verweis auf anderen Server antworten
  - Client befragt dann weitere Server
  - Vorteil: Weniger Aufwand für angefragten Name-Server



10.2.3 DNS-Datenbankeinträge: Resource Records (RR)

- Datenbankeinträge werden als Resource Records (RR) bezeichnet
  - In der Antwort werden ein oder mehrere solche RR mitgeführt

- Tupel mit (Name, Wert, Typ, TTL)
  - Typ=A bzw. Typ=AAAA
    - Name=Hostname, Value=IPv4-, bzw. IPv6-Adresse für Hostname
  - Typ=NS
    - Name=Domain, Value=IP-Adresse des autoritativen Name-Servers für diese Domäne
      - Dieser Server weiß, wie IP-Adressen für Systeme dieser Domäne bestimmt werden können
  - Typ=CNAME
    - Name=Alias hostname, Value=Canonical hostname
      - Hiermit kann kanonischer Name bestimmt werden
  - Typ=MX
    - Name=Alias hostname, Value=Hostname eines E-Mail-Servers
  - TTL entspricht der Gültigkeit eines Eintrags (Time-To-Live)

Ressource Records (RR)

- Varianten von Ressource Records

Typ	Beschreibung
A bzw. AAAA (Address)	Abbildung Name auf IPv4/IPv6-Adresse
MX (Mail Exchange)	E-Mail-Server einer Domäne
NS (Nameserver)	Nameserver einer Domäne
CNAME (Canonical Name)	„Alias“-Namen für Rechner/Domänen
PTR (Pointer)	Abbildung IP-Adresse auf Name
HINFO (Host Info)	Zusätzliche Informationen (CPU, …)

- Weitere Ressource Records-Typen definiert in RFC 1035
  - http://www.iana.org/assignments/dns-parameters

[RFC1035]

10.2.4 DNS: ein verteiltes System aus Name-Servern

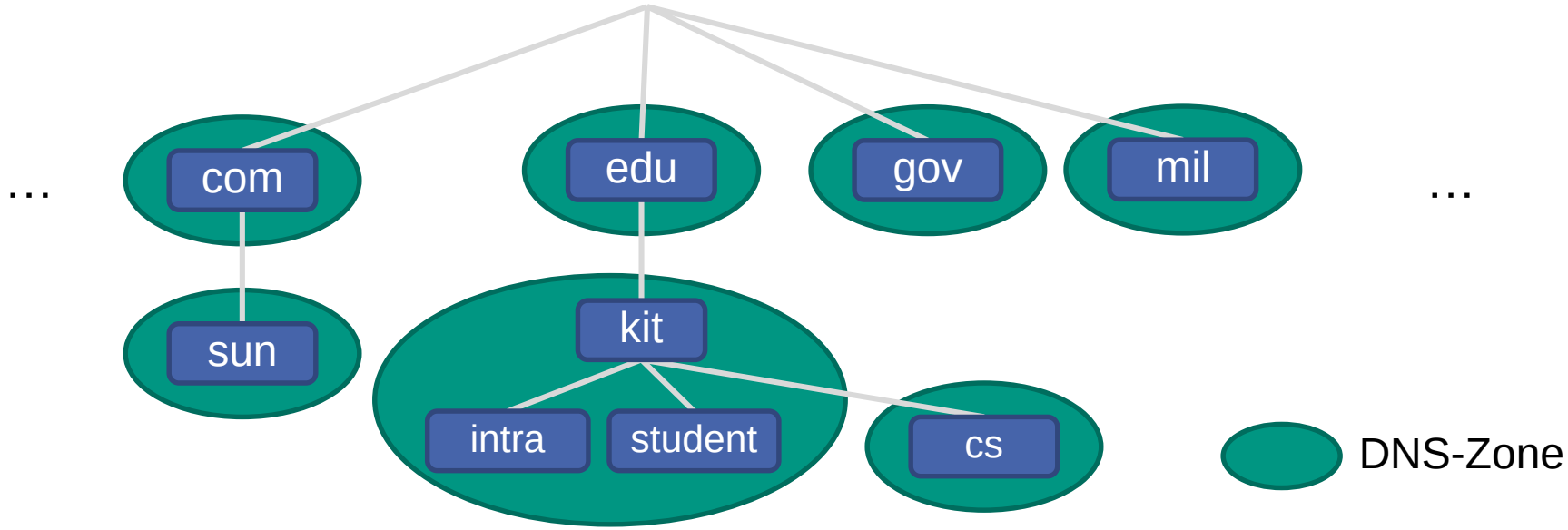
- Wieso kein zentraler Name-Server?
  - Singuläre Fehlerquelle
  - Verkehrsaufkommen an zentralem Server (weltweit!)
  - Entfernung der zentralen Datenbank
  - Verwaltungsaufwand
  - → Ein zentraler Ansatz skaliert hier nicht!

- Verteilung von DNS-Servern
  - Kein Server kennt alle Abbildungen von Namen auf IP-Adressen
  - Lokale DNS-Server
    - Im Allg. haben Netz-Provider, große Firmen etc. einen lokalen DNS-Server
      - Nameserver des KIT: dns-int-01.kit.edu, dns-int-02.kit.edu, …
    - Erste Nachfrage geht immer zu diesem lokalen Server (Default-Server)
  - Autoritativer DNS-Server
    - Enthält autoritative Abbildungen, d.h. liefert maßgebliche Antworten

Hierarchie der Name-Server

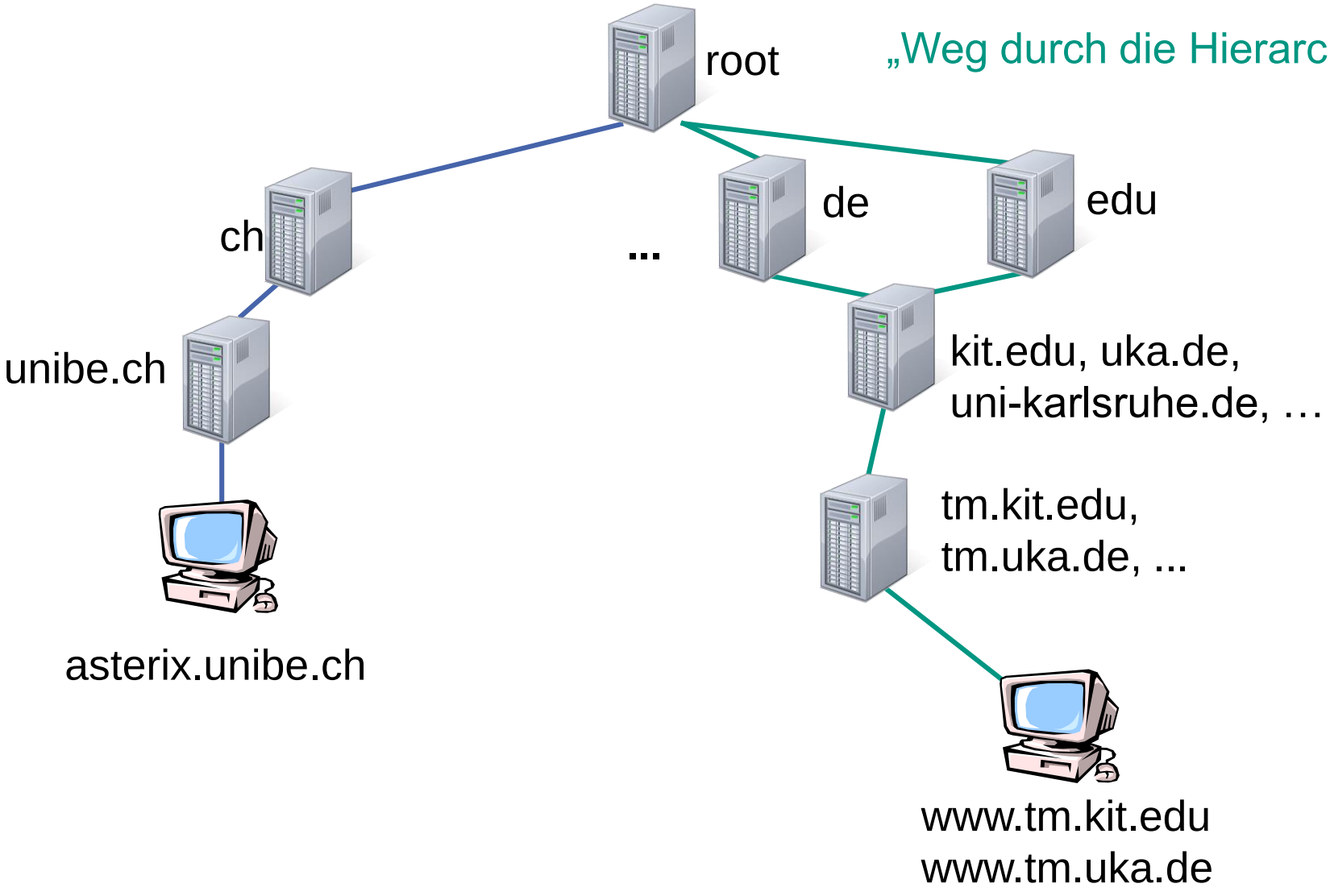
- Name-Server hierarchisch gegliedert
  - Root-Server
    - Weltweit einige wenige solcher Server (bzw. Server-Cluster)
  - Top-Level Domain (TLD) Server
    - Verantwortlicher Server für die Top-Level Domains
  - Autoritative Name-Server
    - Jeder Host ist bei einem solchen Server registriert
    - Häufig gleichzeitig lokaler Name-Server in „seinem“ Netz
  - Lokale Name-Server
    - z.B. Provider, Universität, Firma etc. betreiben lokalen Name-Server
    - Adresse des lokalen DNS-Server manuell konfiguriert oder per DHCP
- Für jede Hierarchie-Ebene (= Domäne) gilt
  - Die IP-Adresse mindestens eines Root-Servers ist bekannt
  - Kennt Name-Server, die für die nächst tiefere Ebene zuständig sind
  - Jeder Name-Server enthält die Einträge, für die er autoritativ ist
    - Besitzt Autorität zur Namensvergabe innerhalb dieser Domäne
  - Ergebnisse von vorhergehenden Anfragen werden zwischengespeichert
    - Timer legt Dauer der Zwischenspeicherung fest

DNS-Zonen



- Zonenkonzept
  - Repräsentiert den Teil des Domänenbaums für den ein Name-Server zuständig ist
  - Kann mehrere Subdomänen enthalten
- Zonendatei beschreibt eine Zone
  - Enthält die Ressource Records der Zone
  - Genau ein SOA RR (*Start of Authority*): Konfiguration der Zone (Gültigkeit, Version)
- Zonentransfer: Spiegelung der Zonendatei auf weitere Nameserver (Redundanz, Performance)

Nameserver-Hierarchie: Beispiel



DNS Root-Server

- IP-Adressen und Lokation der DNS Root-Server unter http://www.root-servers.org
  - IPv4 und bei der Mehrheit auch IPv6-Adressen
  - IP-Adressen sind fix
- Root-Server enthalten nur Einträge für TLDs (Generische und Länder-TLDs)
- Weltweit gibt es 13 Root-Server(-Betreiber)
  - Bezeichnung „A“, …, „M“
  - Bspw. wird der K-Root-Server von RIPE NCC betrieben
  - Root-Server besteht i.d.R. aus mehreren Server-Instanzen
    - insgesamt 480 Server (Stand Juli 2015)
    - Anycast-Dienst

Weltweit verteilte Instanzen des K-Root-Servers

[RIPE14]

- Von RIPE NCC angebotene Instanzen des K-Root-Servers
  - Verteilte Struktur erlaubt besseren Schutz gegen Ausfallsicherheit und kürzere Antwortzeiten durch näher gelegene Server
  - Globale Knoten sind uneingeschränkt global erreichbar
  - Lokale Knoten sind nur für die Kunden der angebundenen Internet Service Provider vorgesehen
    - Z.B. Kunden, die mit DE-CIX in Frankfurt ein sog. Peering haben

Pingo

- Pingo-Link für diese Vorlesung:

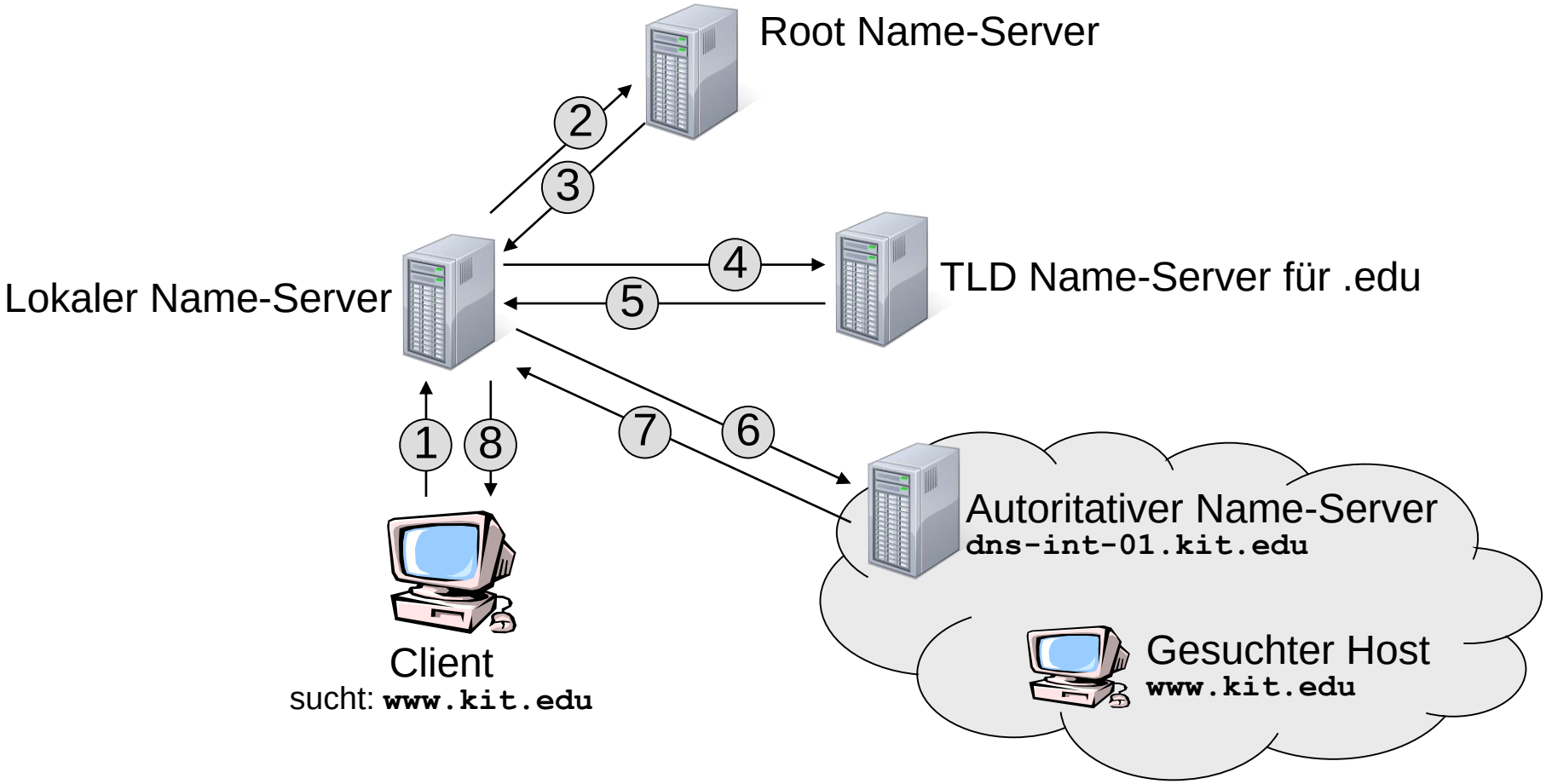
→ http://pingo.upb.de/6466

Pingo

http://pingo.upb.de/

Typischer Ablauf im Internet

- Typischerweise fragt Client seinen lokalen Name-Server rekursiv
- Lokaler Name-Server arbeitet dann normalerweise iterativ



DNS-Beispiel: Akamai

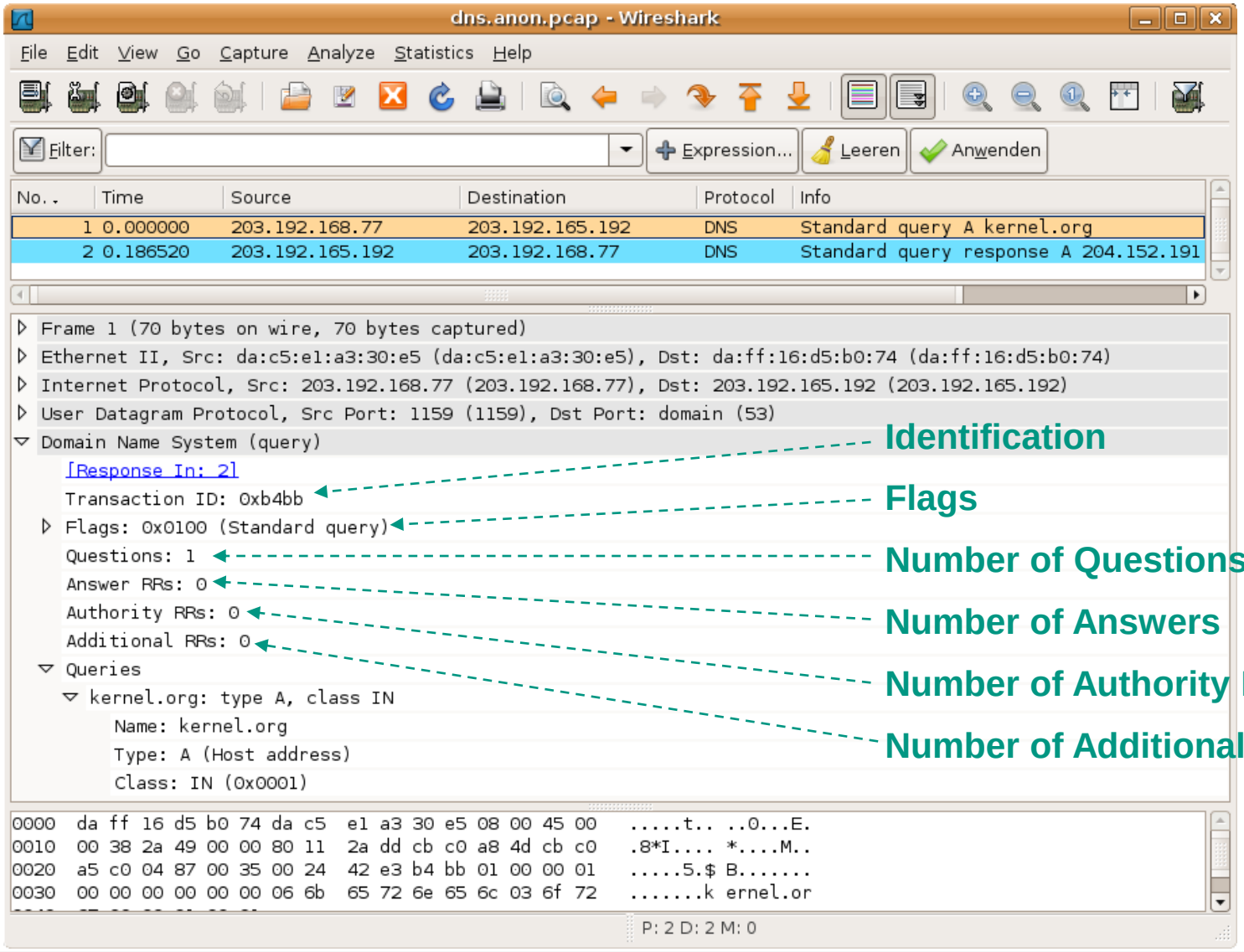
- Ziel
  - Schnellere Auslieferung von Inhalten
- Vorgehensweise
  - Auslieferung durch Proxys → Akamai-Server „nahe“ beim Nutzer (ISP)
- Anpassung der Webseiten notwendig (*Preprocessing*)
  - Eingebettete URLs werden zu ARLs (URLs die auf Akamai zeigen)
  - Text wird i.d.R. von Originalseite geladen, Objekte von Akamai-Servern
- Proxy-Auswahl
  - Entsprechend des Aufenthaltsorts des Nutzers
  - Proxy, der Objekte vorhält (nicht alle Objekte liegen auf allen Proxys)
- Zweistufiger Abbildungsprozess der ARLs
  - HLDNS (High Level DNS, TTL 30 min)
  - LLDNS (Low Level DNS, TTL 30 sec)
- Probleme
  - Aufwendigere Namensauflösung → Erhöhte DNS-Verzögerung/ -Verkehr
  - Nachbarschaftsbestimmung ungenau, da Routing-technisch nicht klar ist, welcher Proxy am „nächsten“ zur IP-Adresse des Users liegt

[Akam09]

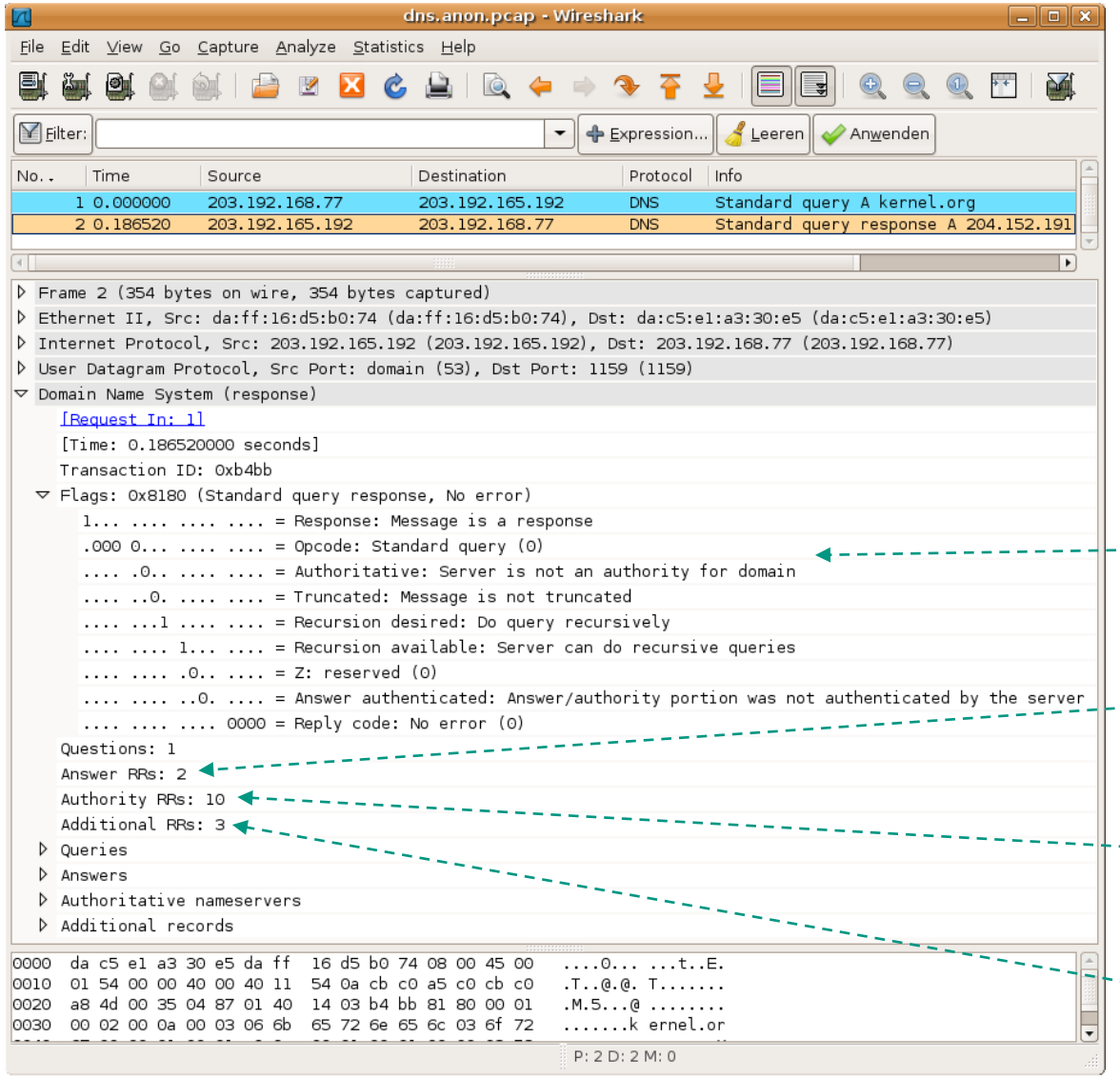
Beispiel: DNS-Anfrage mit dig


```
bash-4.0$ dig  kit.edu MX

; <<>> DiG 9.3.4-P1.2 <<>> kit.edu MX
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51346
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;kit.edu.                       IN      MX

;; ANSWER SECTION:
kit.edu.                3600    IN      MX      10 scc-mailin-01.scc.kit.edu.
kit.edu.                3600    IN      MX      10 scc-mailin-02.scc.kit.edu.
kit.edu.                3600    IN      MX      10 scc-mailin-03.scc.kit.edu.

...

bash-4.0$ dig scc-mailin-01.scc.kit.edu A

; <<>> DiG 9.3.4-P1.2 <<>> scc-mailin-01.scc.kit.edu A
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37589
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;scc-mailin-01.scc.kit.edu.     IN      A

;; ANSWER SECTION:
scc-mailin-01.scc.kit.edu. 3600 IN      A       129.13.185.193
```

Beispiel: DNS-Query – Wireshark



Beispiel: DNS-Reply – Wireshark



Flags im Reply aufgeschlüsselt

2 RRs zu Namen (Answers)

10 RRs anderer autoritativer Server

3 zusätzliche Angaben

Basisdienst DNS – Aktuelle Themen

- DNS Security Extension (DNSSEC)
- Ziel: Gewährleistung von Authentizität und Integrität von DNS-Transaktionen
  - Signierte DNS-Antworten mittels asymmetrischer Kryptografie
    - Besitzer einer Zone signiert jeden einzelnen Record mit geheimem Schlüssel
    - DNS-Clients prüfen mittels öffentlichem Schlüssel
- Erweiterung von DNS nötig
  - Vier neue Resource Record Typen eingeführt
    - Resource Record Signature (RRSIG), DNS Public Key (DNSKEY), Delegation Signer (DS), Next Secure (NSEC)
  - Zwingende Unterstützung großer Nachrichten
- Seit Mitte 2010 auf allen Root-Servern aktiviert
  - .com, .edu, .net, .org, .us, .se, .br, .de … signieren
- Verschlüsselung von DNS-Daten nicht vorgesehen
- Beispiel für RRSIG Record der Domain ripe.net


[Heis10a]


[Heis10c]

```
ripe.net.    9538 IN A  193.0.6.139
ripe.net.    9538 IN RRSIG  A 5 2 21600 20110802100224
                  20110703090224 52192 ripe.net.
                  dS+1JHc1B9d73/6JB2lnfM5ePbSWcf06SXajmIeCK4+wd6tsaPKEgjM/
                  07bR4zGKl+Bj2sdjRRzMAOvDMEM6ZuSDOhY8ADmS7wEt3/4ppV4+e80v
                  6mX7FSWaaezmOSYdo61dDQxE9Uw+93Mv13aHeG+b/jbgZPMJrgjLTBc7
                  LZs=
```

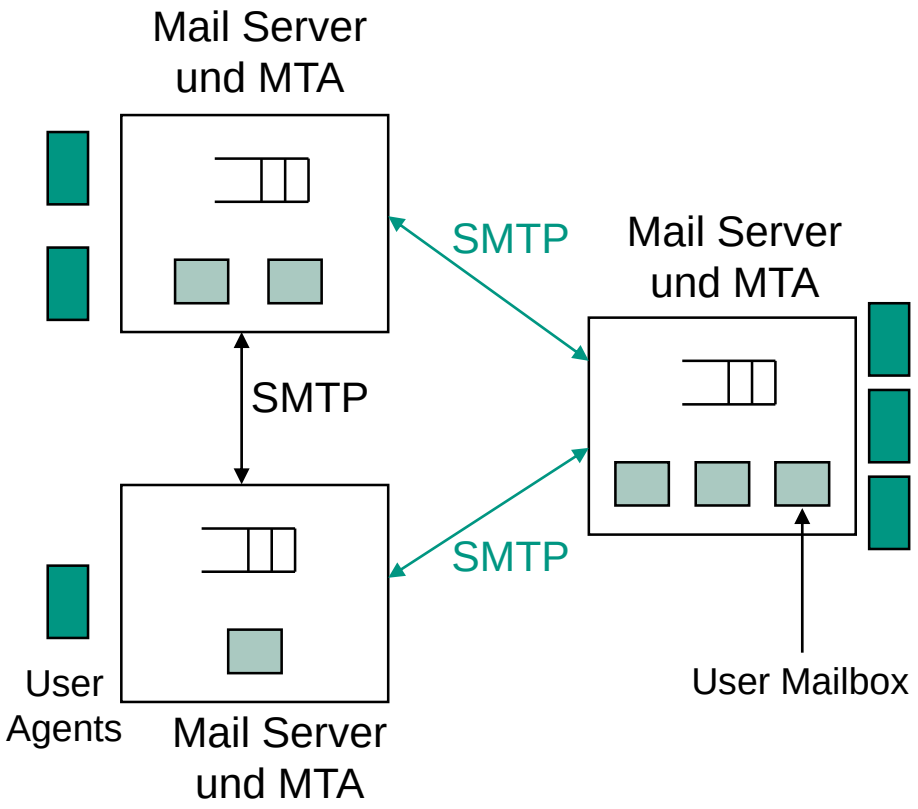
Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Basisdienst DNS
3. Anwendungsbeispiel E-Mail
4. Anwendungsbeispiel World Wide Web

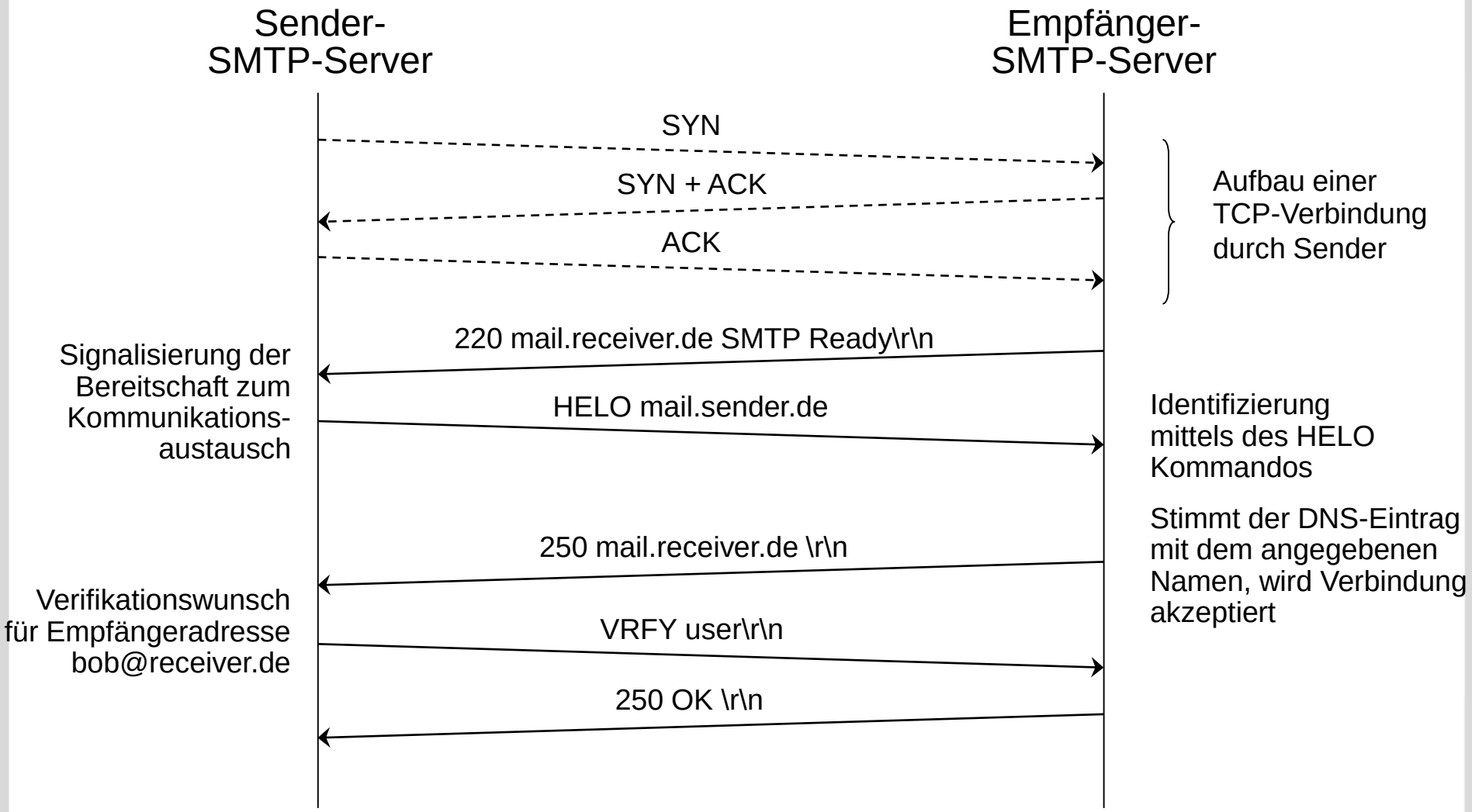
10.3 E-Mail im Internet: SMTP

- E-Mail
  - Vergleichbar mit traditioneller Post
  - Asynchroner Datenaustausch zwischen Benutzern
  - Inhalte
    - Text, Bilder, Audio und Video
- Grundlegende Komponenten
  - User Agent (UA)
  - Mail Server
    - Mail Transport Agents (MTA)
  - Simple Mail Transfer Protocol (SMTP)

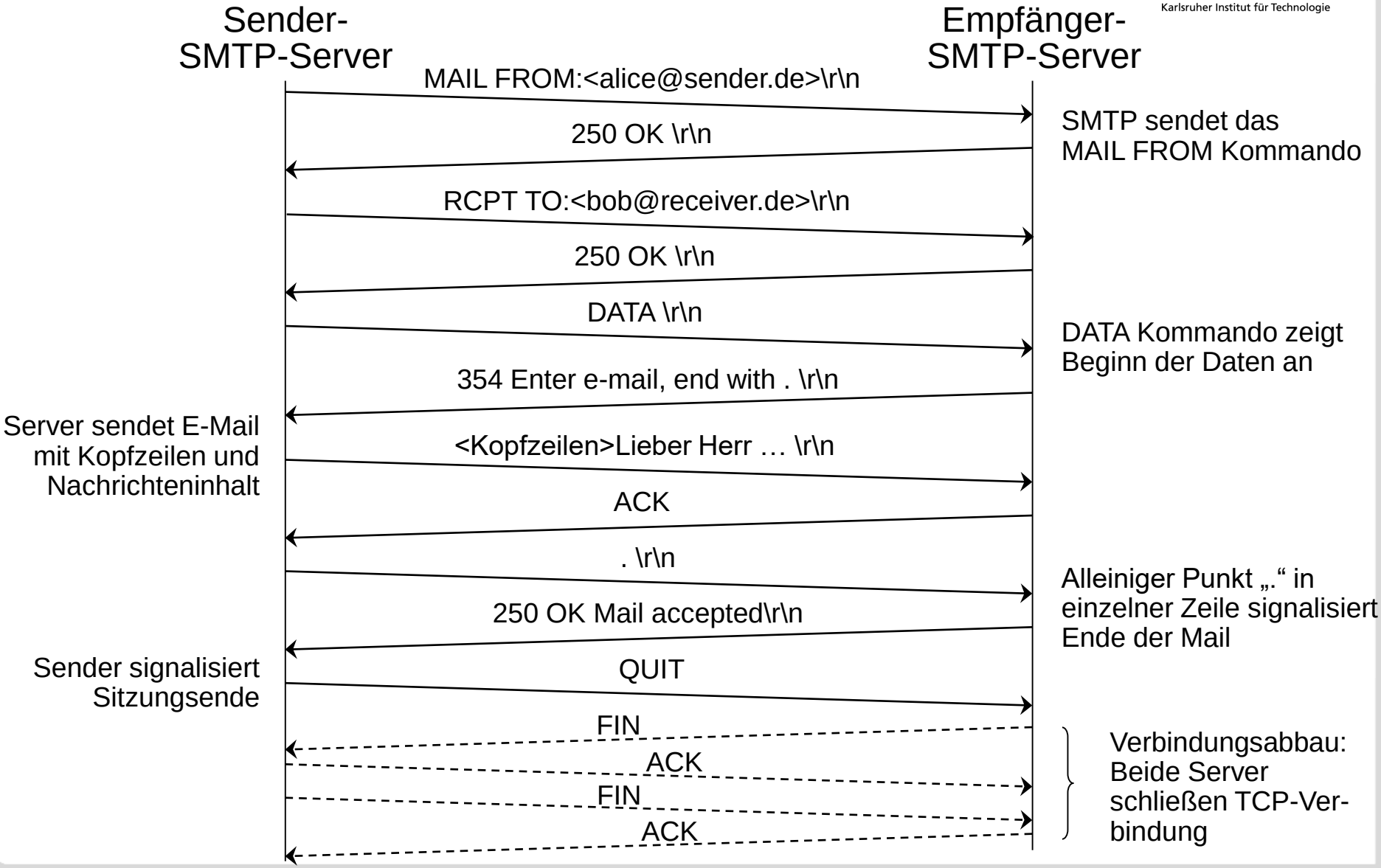


[RFC2821]

Nachrichtenübertragung mit SMTP (1)



Nachrichtenübertragung mit SMTP (2)



E-Mail-Format nach RFC 822

- Beispiel

```
Envelope-to: mroehr@tm.uka.de
Received: from i72pc108.tm.uni-karlsruhe.de ([141.3.71.162] helo=tm.uka.de)
          by irams1.ira.uka.de with smtp (Exim 3.30 #7 (Debian))
          id 161oxI-0005j9-00
          for <mroehr@tm.uka.de>; Thu, 08 Nov 2001 14:11:07 +0100
Message-Id: <E161oxI-0005j9-00@irams1.ira.uka.de>
From: zit@tm.uka.de
Date: Thu, 08 Nov 2001 14:11:07 +0100

Hallo!
...
```

Header

Body

- Grundlegender Aufbau: Header + Body
- Zusätzliche Header-Felder:
  - `BCC`: Blind Carbon Copy
  - `Reply-To`: Antwort-Adresse falls nicht an Sender-Adresse zurück geschickt werden soll
  - `In-Reply-To`: Message-ID auf die Bezug genommen wird
- Probleme des RFC-Standards
  - Keine Übertragung von Binärdateien oder ausführbaren Programmen möglich
  - Keine länderspezifischen Zeichen (7-bit ASCII)

[RFC822]

MIME – Multipurpose Internet Mail Extensions

- Problem: SMTP sieht nur einfache ASCII-Texte (7-bit ASCII) als Nachrichten vor
- MIME (*Multipurpose Internet Mail Extensions*)

[RFC2045/2046]

  - Erweitert den Kopfteil einer Nachricht um Formatinformation
  - Content-Type: Definiert den Typ des E-Mail-Inhalts
    - Bsp.: `content-Type: text/plain; charset=ISO-8859-1`
  - Content-Transfer-Encoding: Definiert die Transfer-Syntax, in der die Daten des Hauptteils übertragen werden
    - Bsp.: `content-transfer-encoding: base64`
- Transfersyntax nach RFC2045
  - 7bit
  - 8bit
  - Binary
  - Base64
  - Quoted-printable

} Im Internet gebräuchlich

MIME: Transfer-Syntax

- Transfersyntax Quoted-printable erlaubt nationale Sonderzeichen
  - Kodierungsvorschrift für Bytes außerhalb des 7-Bit-ASCII-Zeichensatzes
    - Ermöglicht so Abwärtskompatibilität zur 7-Bit-ASCII-Kodierung
  - Kodierung der Zeichen außerhalb der dezimalen Bereiche 33-60 und 62-126
    - Vorangestelltes „=“-Zeichen plus Hexadezimal-Code des Bytes
    - z.B. Darstellung von André (im ISO-8859-1 Zeichensatz): Andr=E9
- Transfersyntax Base64 bildet Binärdaten auf lesbare ASCII-Zeichen ab
  - Drei aufeinanderfolgende Textbytes (24 Bit) werden mit vier 6-Bit-Werten kodiert
    - 64 Zeichen (a-z, A-Z, 0-9, +, /)
  - ca. 30% Overhead durch Base64-Kodierung
- Darstellung kodierter Wörter
  - `=? charset ? encoding ? encoded-text ?=`
- Beispiel aus RFC 1522

```
From: =?US-ASCII?Q?Keith_Moore?= <moore@cs.utk.edu>
To: =?ISO-8859-1?Q?Keld_J=F8rn_Simonsen?= <keld@dkuug.dk>
CC: =?ISO-8859-1?Q?Andr=E9_?= Pirard <PIRARD@vm1.ulg.ac.be>
Subject: =?ISO-8859-1?B?SWYgeW91IGNhbiByZWFkIHRoaXMgeW8=?=
=?ISO-8859-2?B?dSB1bmRlcnN0YW5kIHRoZSBleGFtcGxlLg==?=
```

```
From: Keith Moore <moore@cs.utk.edu>
To: Keld Jørn Simonsen <keld@dkuug.dk>
CC: André Pirard <PIRARD@vm1.ulg.ac.be>
Subject: If you can read this you understand the example.
```

`?Q?` steht für Quoted-printable Kodierung

`?B?` steht für Base64 Kodierung

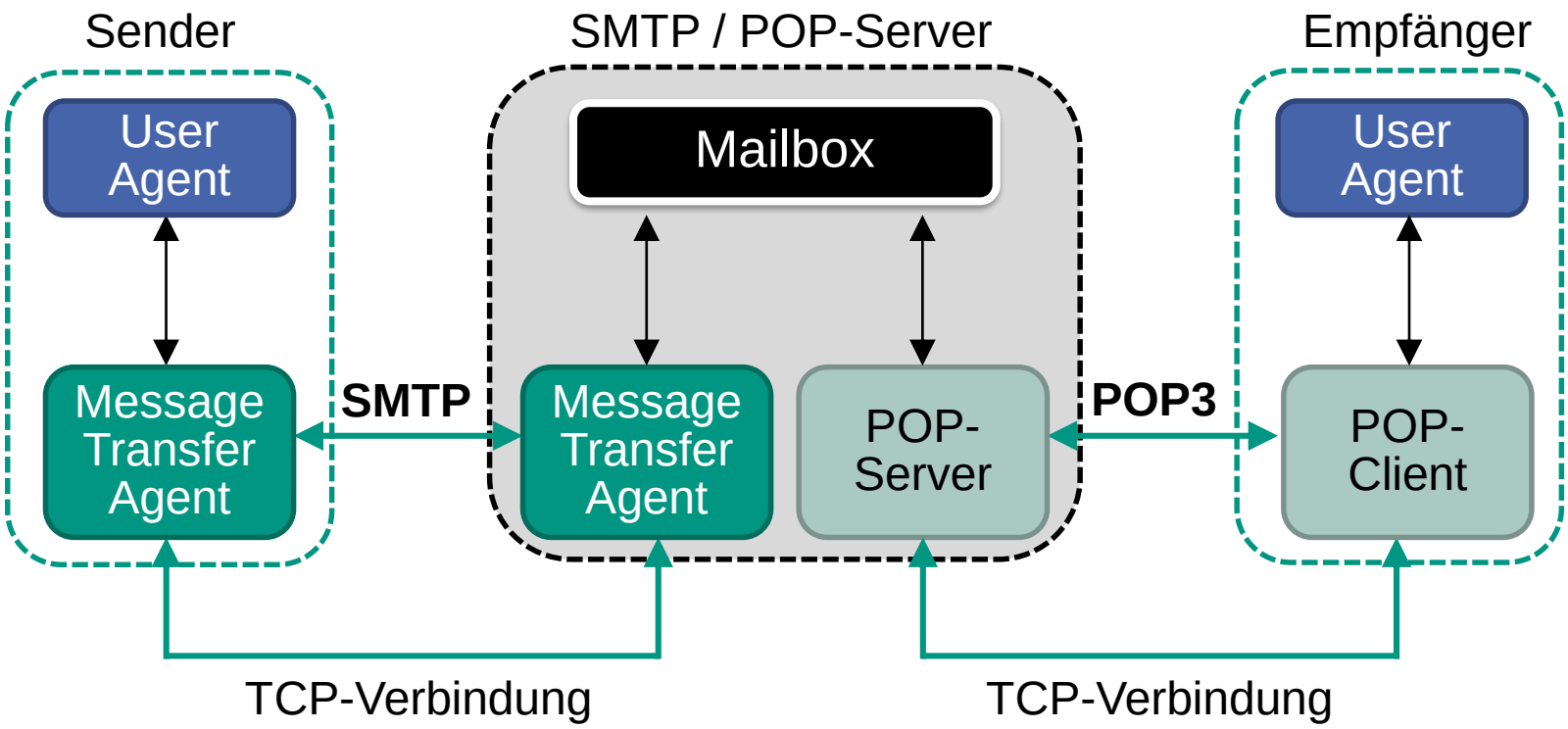
Abfrage des Postfachs

- E-Mail meist zentral über einen Mailserver abgewickelt (Relay-MTA)
  - Abfrage des Postfachs durch weitere Protokolle wie POP3 oder IMAP
  - Auch eine Web-basierte Abfrage des Postfachs ist möglich

- Mittels POP3 (*Post Office Protocol 3*) holt der Client die vom Mailserver empfangenen und in der Mailbox gespeicherten Nachrichten ab
  - Einfache Funktionalität

- IMAP (*Interactive Mail Access Protocol*) dient dazu, die Nachrichten zentral auf einem Mailserver zu verwalten
  - IMAP erlaubt erweiterte Kommandos (z.B. Ordnerverwaltung, Filterung)

Zusammenspiel von SMTP und POP3 am Beispiel



Pingo

http://pingo.upb.de/

Unsichere Email

- Problem: Fehlende Nutzer-zu-Nutzer-Sicherheit
  - Keine Vertraulichkeit und Integrität von Nachrichten
  - Keine Authentifizierung von Sender und Empfänger

- Angriffe
  - Mitlesen von Nachrichten
  - Manipulation oder Fälschen von Nachrichten

- Geforderte Schutzziele
  - Vertraulichkeit der Nachricht
  - Integrität der Nachricht
  - Authentizität des Senders
  - Authentizität des Empfängers

- Abhilfe: de-facto Standard OpenPGP

[RFC4880]

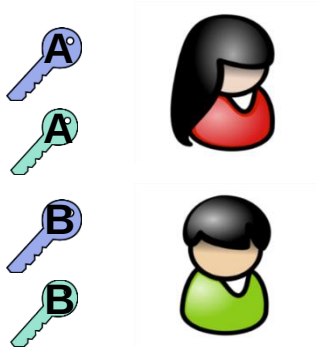
OpenPGP (Open Pretty Good Privacy)

- Datenformat zum Verschlüsseln und / oder Signieren von Nachrichten
  - Zum Beispiel bei EMail
  - Oder zum Speichern von vertraulichen Informationen

- Schutzziele können unabhängig voneinander realisiert werden
  - Vertraulichkeit und Integrität von Nachrichten
    - Mit Hilfe von symmetrischem Verfahren
    - Schlüssel muss dazu ausgetauscht werden
  - Authentizität des Absenders
    - Mit Hilfe von asymmetrischem Verfahren
    - Signatur des Absenders unter der Nachricht
    - Abbildung zwischen Signatur und Identität muss überprüft werden
  - Authentizität des Empfängers
    - Mit Hilfe von asymmetrischem Verfahren
    - Durch Verwendung des öffentlichen Schlüssels des Empfängers
    - Abbildung zwischen Identität und Schlüssel muss überprüft werden

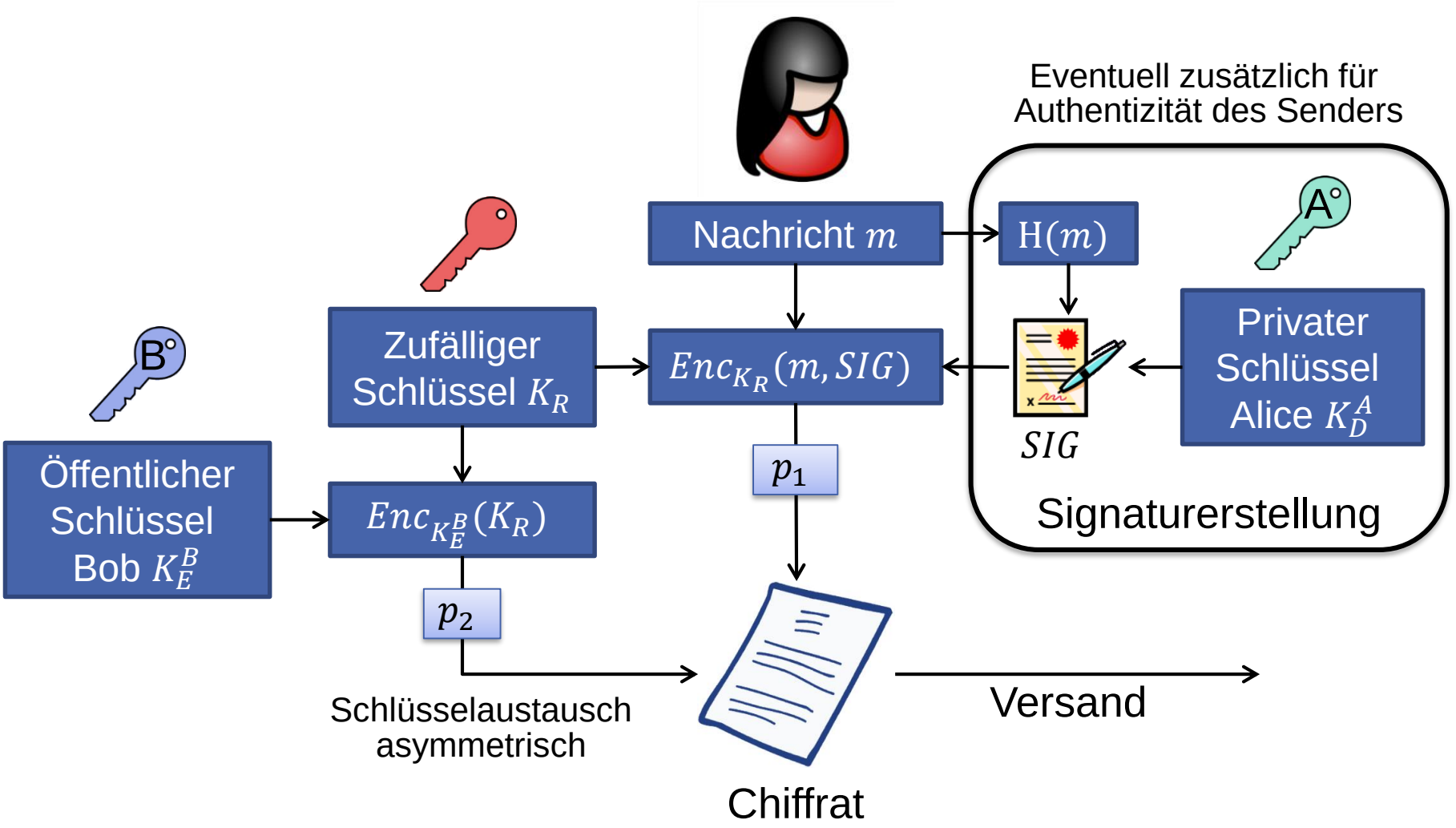
Beispiel

- Nachrichtenversand von Alice an Bob
  - Alice erzeugt
    - Öffentlichen Schlüssel K_E^A
    - Privaten Schlüssel K_D^A
  - Bob erzeugt
    - Öffentlichen Schlüssel K_E^B
    - Privaten Schlüssel K_D^B

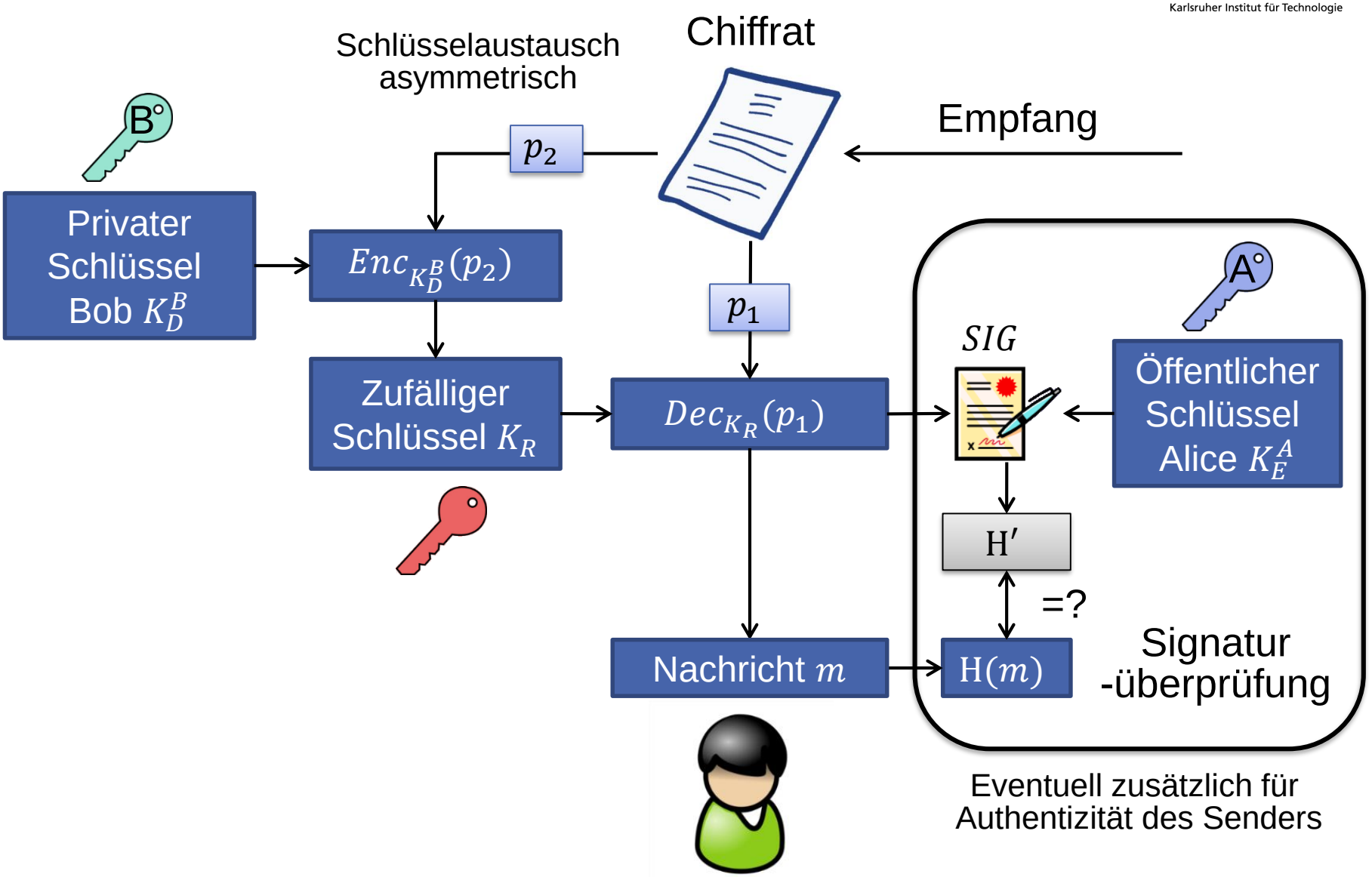


- Schutzziel Vertraulichkeit und Integrität der Nachricht
  - Hybrides Verfahren
    - Schlüsselaustausch asymmetrisch, Verschlüsselung symmetrisch
  - Alice
    - Verschlüsselt Nachricht M symmetrisch mit zufälligem Schlüssel K_R
    - Verschlüsselt Schlüssel K_R asymmetrisch mit Bobs öffentlichem Schlüssel K_E^B
    - Sendet Chiffrat der Nachricht und Chiffrat des Schlüssel an Bob
  - Bob
    - Entschlüsselt den Schlüssel K_R asymmetrisch mit privatem Schlüssel K_D^B
    - Entschlüsselt Nachricht M mit symmetrischen Schlüssel K_R

Versandt von Nachrichten



Empfang von Nachrichten



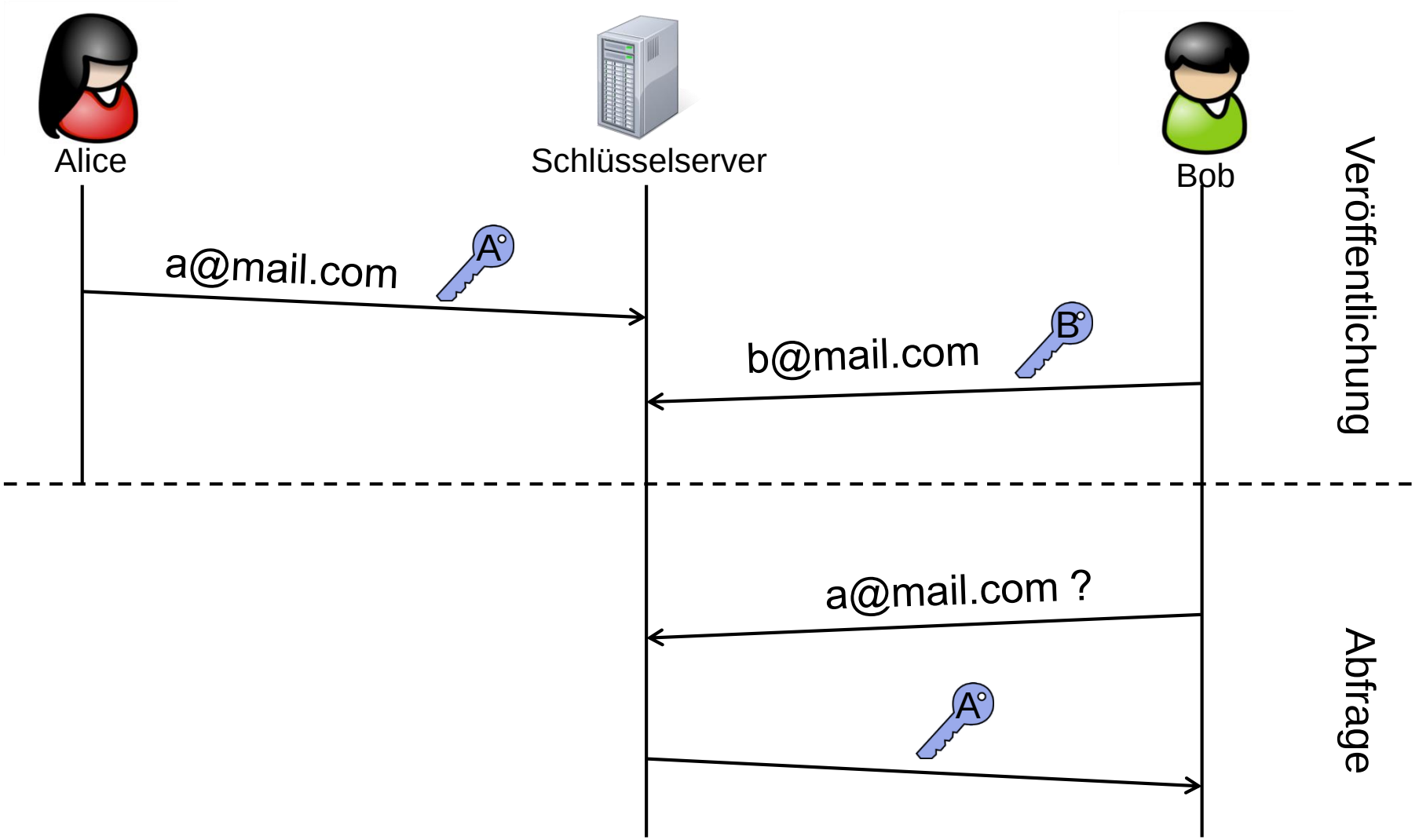
Bekanntgabe öffentlicher Schlüssel

- Verwendung von öffentlichen Schlüsseln
  - Müssen Nutzern vor Versand bekannt sein
  - Zugehörigkeit zu Mailadresse muss überprüfbar sein
    - Für Authentizität des Absenders
    - Für Authentizität des Empfängers

- Bekanntgabe des zur Mailadresse gehörenden öffentlichen Schlüssels
  - Persönlich oder per Mail
  - Über eine bekannte vertrauenswürdige dritte Partei
  - Eintragung auf wohlbekannten Schlüsselservern
    - z.B. pgp.mit.edu, keyserver.pgp.com, zimmermann.mayfirst.org, …

- Überprüfung der Abbildung Mailadresse und öffentlicher Schlüssel
  - Per Telefon, Vergleich des Hashwertes (Fingerprint) des Schlüssels
  - Durch eine vertrauenswürdige Signatur auf den öffentlichen Schlüssel

Öffentliche Schlüsselserver



Wie kann Alice sicher sein, dass

zur Mailadresse von Bob gehört?

Web of Trust (WoT)

- Idee
  - Keine zentrale vertrauenswürdige dritte Partei
  - Völlig dezentraler anarchischer Ansatz
  - Auslagern auf individuelle vertrauenswürdige andere Nutzer
- Strikte Trennung
  - Grad des Vertrauens in andere Nutzer und in deren Signaturen
    - Owner Trust und daraus abgeleitet Signatory Trust
    - Unbekannt, kein, geringes, volles oder absolutes Vertrauen
  - Grad der Authentizität eines öffentlichen Schlüssels
    - Unbekannt, keine, teilweise oder volle Authentizität
- Ablauf
  - Nutzer bauen untereinander graduelle Vertrauensbeziehungen auf
  - Nutzer können öffentliche Schlüssel signieren
  - Nutzer vertrauen Signaturen anderer vertrauenswürdiger Nutzer

WoT – Key Legitimacy

- Nutzer kann einstellen ab wann er einen öffentlichen Schlüssel als authentisch akzeptiert

- Durch Berechnung der sogenannten Key Legitimacy L
  - $L := \frac{Signaturen\ mit\ geringem\ Vertrauen}{X} + \frac{Signaturen\ mit\ vollem\ Vertrauen}{Y}$
  - Vertrauen in Signaturen direkt abgeleitet von Vertrauen in Nutzer
  - Parameter X und Y frei wählbar

- Bewertung der Key Legitimacy (Authentizität)
  - $L < 0$ überprüfter Schlüssel nicht authentisch
  - $0 < L < 1$ überprüfter Schlüssel teilweise authentisch
  - $1 < L$ überprüfter Schlüssel voll authentisch

- Achtung: Vertrauen in eigenen Schlüssel ist absolut
  - Daraus abgeleitet volle Authentizität selbstsignierter Schlüssel

Authentizität eines öffentlichen Schlüssels

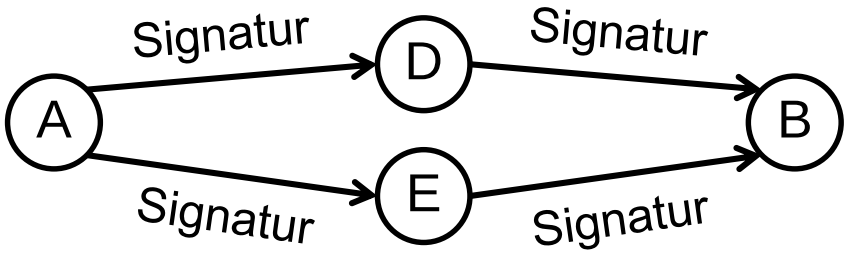
- Problemstellung
  - Alice schickt Nachricht an Bob
    - Verwendet Bob‘s öffentlichen Schlüssel
  - Gehört öffentlicher Schlüssel tatsächlich zu Bob‘s Mailadresse?

- Überprüfung der Authentizität des Schlüssels
  - Alice ruft Bob‘s Schlüssel und alle zugehörigen Signaturen ab
    - Eine Signatur stammt von Carl dem Alice bereits voll vertraut
  - Alice überprüft Carl‘s Signatur mit seinem öffentlichen Schlüssel
    - Alice ist sich bereits sicher dass Carl‘s Schlüssel authentisch ist
  - Alice ist von der Authentizität von Bob‘s Schlüssel überzeugt

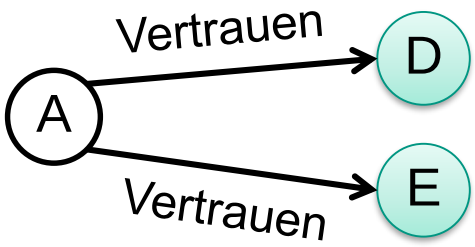
- Transitivität
  - Alice vertraut Carl voll
  - Carl hat Bob‘s Schlüssel signiert
  - Alice akzeptiert daher Bob‘s Schlüssel als voll authentisch

WoT – Beispiel

- Signaturen
  - Nutzer Alice hat den öffentlichen Schlüssel von Dan und Eve signiert
  - Nutzer Dan und Eve haben den öffentlichen Schlüssel von Bob signiert



- Vertrauen
  - Nutzer Alice hat in Dan und Eve geringes Vertrauen



- Von Alice berechnete Key Legitimacy für Bob
  - $L := \frac{2}{2} + \frac{0}{1} = 1$
  - Volle Authentizität von Bob‘s öffentlichem Schlüssel

WoT – Zusammenfassung

- Kein Vertrauen in zentrale Instanz notwendig
  - Anarchischer Ansatz
  - Öffentliche Schlüssel von Nutzern müssen bekanntgegeben werden
    - Über Schlüsselserver oder direkt
- Statt dessen graduelles Vertrauen in Nutzer: Owner Trust
  - Daraus abgeleitet Vertrauen in Signaturen dieser Nutzer: Signatory Trust
- Überprüfung von öffentlichen Schlüsseln über Signaturketten von vertrauenswürdigen Nutzern
  - Authentizität des Absenders
  - Authentizität des Empfängers
- Authentizität nur so stark wie schwächstes Glied
  - Parameter frei einstellbar, je nach Sicherheitsbedürfnis
  - Vertrauen in andere Nutzer muss wohl bedacht gewählt werden
  - Nutzer können Signaturen nach belieben ausstellen

Pingo

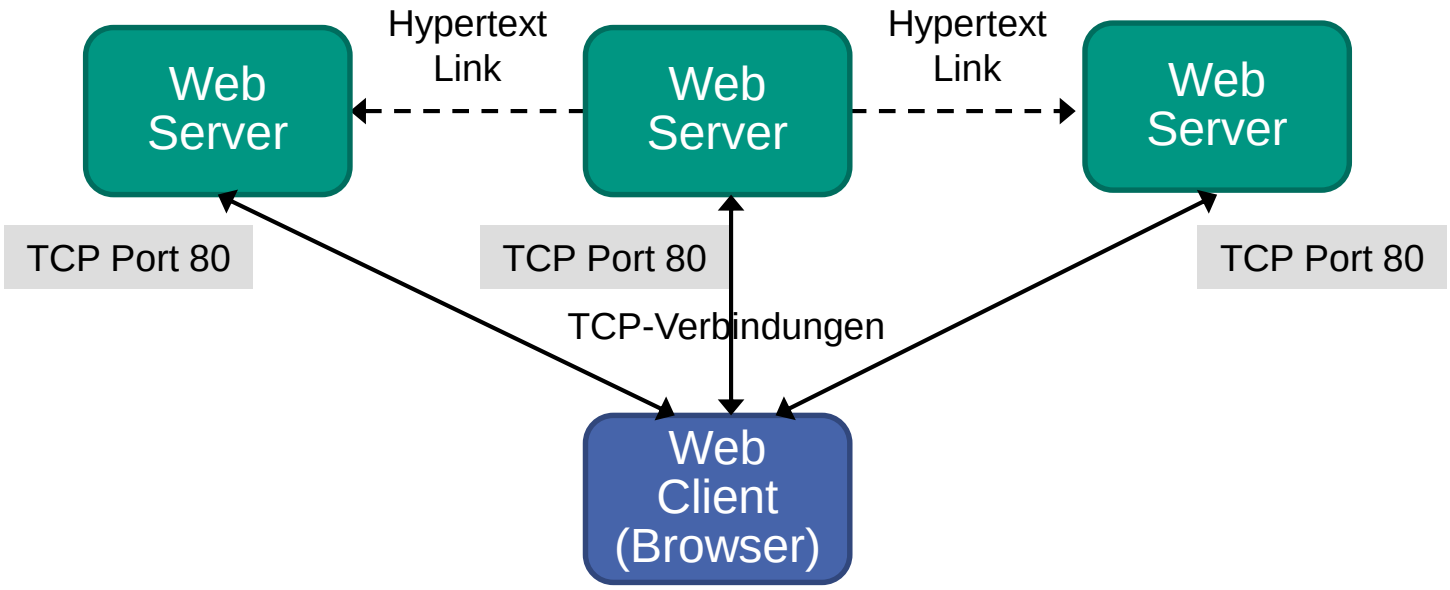
http://pingo.upb.de/

Kapitelübersicht

1. Einführung
2. Netzwerkarchitekturen
3. Physikalische Grundlagen
4. Protokollmechanismen
5. Die Sicherungsschicht: HDLC
6. Die Sicherungsschicht: Lokale Netze
7. Netzkopplung und Vermittlung
8. Die Transportschicht
9. Sicherheit
10. Anwendungssysteme

1. Einführung
2. Basisdienst DNS
3. Anwendungsbeispiel E-Mail
4. Anwendungsbeispiel World Wide Web

10.4 World Wide Web (WWW)



- Client-Server-Anwendung
- Hypertext
  - Über ein Netz verbundene Objekte
  - Die grundlegende Idee „Hypertext“ ist wesentlich älter als WWW
- Grafischer Browser
- WWW – eine „Killer“-Anwendung im Internet
  - Trug wesentlich zur Verbreitung des Internets bei

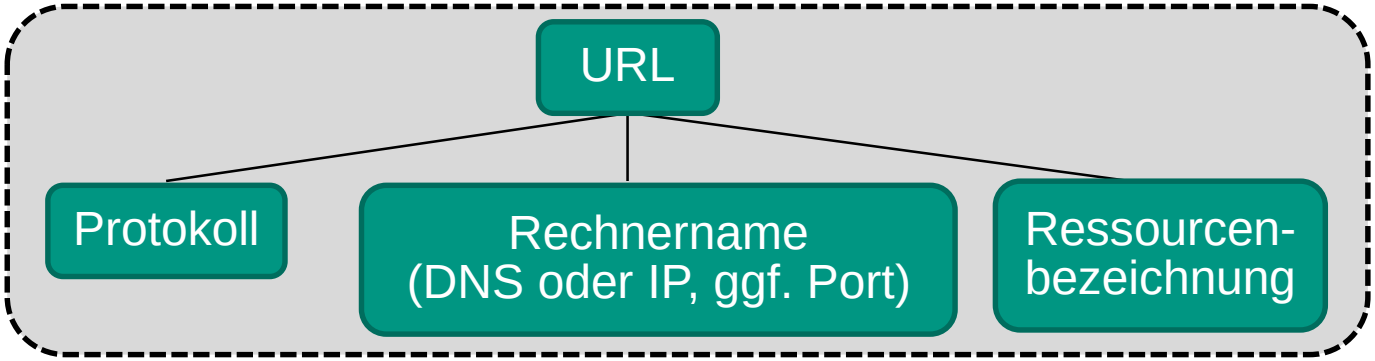
Entwicklung des WWW

- U.a. hervorgegangen aus Arbeiten des britischen Informatikers Tim Berners-Lee am europäischen Forschungszentrum CERN (Genf)
  - Ziel: Einfacher weltweiter Austausch von Dokumenten zwischen Wissenschaftlern
  - Erster Prototyp Ende 1980
  - Grafisch und zeilenorientiert
- Durchbruch des WWW durch den WWW-Client Mosaic
  - Entwickelt von Marc Andreesen und Eric Bina (University of Illinois)
  - Ursprünglich für X-Windows-Systeme
  - Als Quellcode per FTP kostenlos verfügbar → schnelle Verbreitung
  - Marc Andreesen und Jim Clark gründeten im April 1994 die Firma Netscape Communication Corporation
- Gründung des W3-Konsortiums im Juli 1994
  - Vorrangiges Ziel: Weiterentwicklung des WWW, z.B. durch Standardisierung von HTML
  - Vorsitzender: Tim Berners-Lee
  - Infos unter http://www.w3.org

Eindeutige Adressierung eines Web-Dokuments

- WWW nutzt hierzu den Mechanismus des Uniform Resource Identifier (URI) bzw. Uniform Resource Locator (URL)
  - Repräsentiert eine Ressource
  - Auch für Inhalte anderer Server (USENET, FTP, E-Mail, SIP) verwendbar

[RFC3986]



- Ressourcenbezeichnung identifiziert das Objekt, auf das im jeweiligen Server zugegriffen werden soll, identifiziert
  - Bei WWW: aufgerufene Web-Seite
  - Bei Mail: Empfänger der Mail (also die Mail-Adresse)

Komponenten des WWW

- Universal Resource Locator (URL)
  - kompakte Repräsentation der Lokation und Zugriffsmethode auf Internet-Ressourcen (vgl. RFC 1738 und RFC-Updates)
    - http://<host>:<port>/<path>?<searchpart>
    - mailto:<rfc822-addr-spec>
    - nntp://<host>:<port>/<newsgroup-name>/<article-number>

[RFC1738]

- Hypertext
  - Hypertext Markup Language (HTML) zur Beschreibung von WWW-Seiten
    - Strukturierung der Dokumente in Kopf (Header) und Rumpf (Body)
    - Tags zur Markierung von Formatierungsanweisungen
      - `<b>Hallo</b>` → **Hallo** (fett)
      - `<i>Hallo</i>` → *Hallo* (kursiv)
      - `<a href="http://www.kit.edu">KIT</a>` → KIT (Hyperlink)
- Abruf/Austausch von HTML-Seiten
  - Hypertext Transfer Protocol (HTTP)
    - Zustandsloses Protokoll (HTTP-Server hält keine Information über Clients)
    - Oberhalb TCP angeordnet

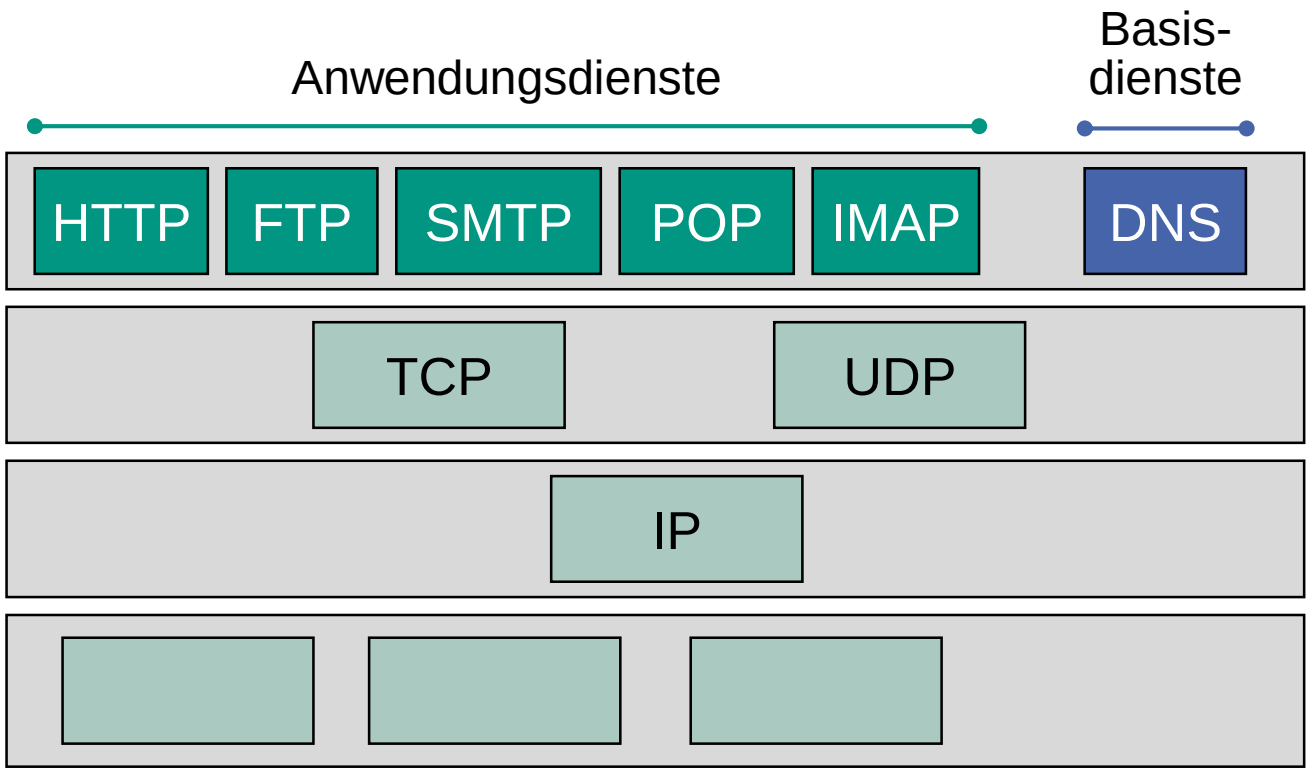
Hypertext Transfer Protocol (HTTP)


[RFC1945]

- Einfaches ASCII-basiertes Transferprotokoll
- Zwei Typen von Nachrichten: Request und Response
  - Zustandslos: Jeder Request wird individuell bearbeitet
  - Setzt auf TCP-Verbindungen auf
  - Default-Port: 80 (World Wide Web HTTP)
- Beispiele von Requests
  - GET: Aufruf eines Objekts (mit und ohne Protokollversion)
  - HEAD: Aufruf des Kopfs einer WWW-Seite
  - PUT: Speichern einer WWW-Seite
  - POST: Übermittlung neuer Informationen; z.B. durch interaktive Benutzer ausgefüllter Inhalt von Formularen
  - DELETE: Löschen einer WWW-Seite
- HTTP wird nicht nur für HTML-Seiten verwendet
  - Datei-Transfer
  - WebDAV: Datei-Verwaltung
  - Web Services
  - …

Zusammenfassung

- Anwendungssysteme bilden oberste Schicht im Internet-Stack
- Nutzung der Protokolle der Transportschicht
- Unterscheidung von Basisdiensten und Anwendungsdiensten



Pingo

http://pingo.upb.de/

Aufgaben

10.1 Was ist ein URL, wozu dient er und wie ist er aufgebaut?

10.2 Wie läuft üblicherweise der Abruf einer WWW-Seite über HTTP ab?

10.3 Erläutern Sie die Aufgaben des Domain Name Systems und dessen Aufbau

10.4 Wie läuft eine Namensauflösung ab und welche unterschiedlichen Anfragetypen können hierbei vorkommen?

10.5 Beschreiben Sie, worin eine DNS-Antwort besteht und geben Sie Beispiele, was sie enthalten kann

10.6 Wozu dient und wie funktioniert eine rekursive DNS-Abfrage?

10.7 Wie werden E-Mails über das Internet transportiert?

10.8 Erläutern Sie, mit welchen Mitteln über E-Mails mehr als nur Text transportiert werden kann

10.9 Welche Protokolle spielen beim Abruf von E-Mails aus Mailboxen die Hauptrolle und worin bestehen deren wichtigste Unterschiede?

Literatur

[Akam09] http://www.akamai.com

[DENI15] DENIC eG; http://www.denic.de/hintergrund/statistiken/

[Hals05] F. Halsall; Computer Networking and the Internet; Addison-Wesley, 5th Edition, 2005

[Heis08] R. Kaps; IPv6 für DNS-Rootserver; heise online News, http://heise.de/-186505, Februar 2008

[Heis10a] U. Wisser; Domain Name System absichern mit DNSsec; *http://heise.de/-903318*, Januar 2010

[Heis10b] M. Parbel; Fehlerhafte Nameserverdaten-Aktualisierung legte Domain .de lahm; heise online News, http://heise.de/-1000581, Mai 2010

[Heis10c] M. Ermert; DNSSEC in der DNS-Rootzone gestartet; heise online News, http://heise.de/-1039401; Juli 2010

[Heis11] M. Ermert; Neue Top Level Domains: ICANN beschließt Öffnung des Internet-Namensraums; heise online News, http://heise.de/-1263102; Juni 2011

[ICANN] http://newgtlds.icann.org/en/

[KuRo10] J. Kurose, K. Ross, Computer Networking – A Top-Down Approach, Pearson, 5th Edition, 2010

Literatur

[RFC822] David H. Crocker (Ed.); Standard for the Format of ARPA Internet Text Messages; RFC 822, Internet Engineering Task Force, August 1982

[RFC1034] P. Mockapetris; Domain Names – Concepts and Facilities; RFC 1034, Internet Engineering Task Force, November 1987

[RFC1035] P. Mockapetris; Domain Names – Implementation and Specification; RFC 1035, Internet Engineering Task Force, November 1987

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk; Hypertext Transfer Protocol – HTTP/1.0; RFC 1945, Internet Engineering Task Force, Mai 1996

[RFC2045] N. Freed, N. Borenstein; Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies; RFC 2045, Internet Engineering Task Force, November 1996

[RFC2046] N. Freed, N. Borenstein; Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types; RFC 2046, Internet Engineering Task Force, November 1996

[RFC2821] J. Klensin (Ed.); Simple Mail Transfer Protocol; RFC 2821, Internet Engineering Task Force, April 2001

[RFC3986] T. Berners-Lee, R. Fielding, L. Masinter; Uniform Resource Identifier (URI): Generic Syntax; RFC 3986, Internet Engineering Task Force, Januar 2005

[RIPE14] RIPE NCC; k-root-server; http://k.root-servers.org